

*Synthèse de l'activité scientifique en
vue d'obtention de l'Habilitation à
Diriger des Recherches*

Contribution à la Surveillance des Systèmes de Production et Utilisant les Techniques de l'Intelligence Artificielle

Daniel RACOCEANU

*Maître de Conférences à l'Université de Franche-Comté
Chercheur au Laboratoire d'Automatique de Besançon
UMR-CNRS 6596*



Avants Propos

Les travaux présentés dans ce mémoire ont été réalisés au Laboratoire d'Automatique de Besançon, dirigé par Monsieur le Professeur Alain Bourjault. Qu'il soit ici persuadé de ma reconnaissance pour m'avoir accueilli au LAB et m'avoir permis de m'exprimer dans le cadre passionnant du métier d'enseignant-chercheur.

Ma première pensée va tout naturellement vers Monsieur le Professeur Noureddine Zerhouni, responsable de l'équipe « ingénierie de maintenance et e-maintenance » du LAB, auquel je tiens à exprimer ici ma profonde gratitude pour les encouragements, la confiance et la sympathie dont il a toujours fait preuve à mon égard. Son dynamisme, son ingéniosité, sa vision scientifique et sa bienveillance ont constitué pour moi un vrai soutien, tout au long de mes travaux.

Qu'il me soit également permis de remercier chaleureusement :

Monsieur Pierre Borne

Monsieur Denis Hammad

Monsieur Daniel Noyes

pour l'honneur qu'ils m'ont fait d'avoir expertisé ce travail.

Mes remerciements vont également à :

Monsieur Alain Bourjault

Monsieur Abdellah El Moudni

Monsieur Noureddine Zerhouni

Qu'ils reçoivent ici toute ma gratitude pour l'intérêt qu'ils portent à ces travaux en participant au jury.

Enfin, je tiens à exprimer mes remerciements à tous ceux qui ont participé aux travaux de recherche présentés dans ce document, ainsi qu'à tous mes collègues et amis du LAB et de l'UFR Sciences.

Table des Matières

NOTATIONS ET ABREVIATIONS.....	5
PARTIE I - PARCOURS PROFESSIONNEL	7
CHAPITRE 1 - CURRICULUM VITAE.....	9
1.1 RENSEIGNEMENTS GENERAUX	9
1.1.1 <i>Etat civil.....</i>	9
1.1.2 <i>Parcours professionnel.....</i>	9
1.1.3 <i>Etudes, titres, diplômes.....</i>	10
1.1.4 <i>Thèse.....</i>	10
1.2 PARCOURS SCIENTIFIQUE.....	10
1.3 RESULTATS SCIENTIFIQUES MAJEURS	11
1.3.1 <i>Simplification des systèmes à évènements discrets complexes.....</i>	11
1.3.2 <i>Ingénierie de la maintenance et E-maintenance.....</i>	12
1.4 ANIMATION DE L'AXE SURVEILLANCE PAR INTELLIGENCE ARTIFICIELLE.....	13
1.4.1 <i>Co-encadrement de theses.....</i>	14
1.4.2 <i>Encadrement de stages DEA et Master Recherche.....</i>	15
1.5 ANIMATION DE LA RECHERCHE.....	17
1.5.1 <i>Au niveau régional.....</i>	17
1.5.2 <i>Au niveau national.....</i>	18
1.5.3 <i>Au niveau international.....</i>	19
1.6 COLLABORATIONS SCIENTIFIQUES	20
1.7 DIVERS RECHERCHE.....	21
1.7.1 <i>Participation à des jurys de thèse.....</i>	21
1.7.2 <i>Congres internationaux.....</i>	21
1.7.3 <i>Relecture d'articles.....</i>	21
1.8 PUBLICATIONS.....	23
1.8.1 <i>Revue internationale avec comité de lecture.....</i>	23
1.8.2 <i>Conférences sur invitation personnelle.....</i>	24
1.8.3 <i>Communications a des congres internationaux avec comite de lecture et publication des actes.....</i>	24
1.8.4 <i>Communications a des congres nationaux ou a audience francophone avec comite de lecture et publication des actes.....</i>	26
1.8.5 <i>Communications dans le cadre de groupes de travail et colloques nationaux.....</i>	27
1.8.6 <i>Rapports de contrats de recherche.....</i>	27
1.8.7 <i>Rapports de recherche projet europeen.....</i>	28
1.8.8 <i>Communications diverses.....</i>	29
1.9 RESUME DES ACTIVITES D'ENSEIGNEMENT	29
1.10 EXPERIENCE EN ENTREPRISE.....	30
1.11 RESUME DES RESPONSABILITES SCIENTIFIQUES, COLLECTIVES ET ADMINISTRATIVES.....	31
1.12 FORMATIONS SUIVIES	31

CHAPITRE 2 - ACTIVITES D'ENSEIGNEMENT.....	33
2.1 PREAMBULE	33
2.2 RESUME DES ACTIVITES D'ENSEIGNEMENT A BELFORT	33
2.3 BILAN DES ACTIVITES D'ENSEIGNEMENT A BESANÇON	33
2.3.1 <i>Structure de l'enseignement avant la mise en place du LMD</i>	33
2.3.2 <i>Structure de l'enseignement après la mise en place de l'organisation LMD</i>	35
2.3.3 <i>Formation continue</i>	39
2.3.4 <i>Suivi d'apprentis, encadrement de projets et stages</i>	39
2.3.5 <i>Relations industrielles par l'enseignement</i>	41
2.3.6 <i>Autres activités et responsabilités pour l'enseignement</i>	41
PARTIE II - CONTRIBUTION SCIENTIFIQUE	43
INTRODUCTION GENERALE	45
CHAPITRE 3 - PROBLEMATIQUE SCIENTIFIQUE : SURVEILLANCE INTELLIGENTE	49
3.1 INTRODUCTION.....	49
3.2 MAINTENANCE ET SURVEILLANCE INDUSTRIELLE DANS LE CONTEXTE DE LA PRODUCTION MODERNE : TELEMANTENANCE, E-MAINTENANCE, MAINTENANCE ET SURVEILLANCE INTELLIGENTE	50
3.3 SURVEILLANCE INDUSTRIELLE : DEFINITIONS, ETAT DE L'ART	54
3.3.1 <i>Introduction</i>	54
3.3.2 <i>Définitions des termes de base utilisés en surveillance</i>	54
3.3.3 <i>Etat de l'art des méthodes de surveillance. Intérêt de l'utilisation des techniques de l'Intelligence artificielle</i>	57
3.4 CONCLUSION ET SYNTHESE	82
CHAPITRE 4 - SURVEILLANCE DYNAMIQUE PAR RESEAUX DE NEURONES RECURRENTS	85
4.1 INTRODUCTION.....	85
4.2 PROPRIETES FONDAMENTALES DES RESEAUX DE NEURONES ARTIFICIELS	87
4.2.1 <i>Apprentissage et mémoire</i>	87
4.2.2 <i>Sous-apprentissage, généralisation et sur-apprentissage</i>	88
4.3 UTILISATION DES RESEAUX DE NEURONES POUR LA SURVEILLANCE INDUSTRIELLE	89
4.3.1 <i>Approximation de fonctions et prédiction par RN</i>	89
4.3.2 <i>Classification ou discrimination par RN</i>	91
4.3.3 <i>Surveillance dynamique par RN</i>	92
4.4 REPRESENTATION DU TEMPS DANS LES RESEAUX DE NEURONES	94
4.4.1 <i>Architectures Neuronales Temporelles</i>	94
4.4.2 <i>Analyse comparative entre les représentations dynamique et spatiale pour la surveillance dynamique</i>	103
4.5 UNE NOUVELLE STRUCTURE DE RESEAU DYNAMIQUE POUR LA SURVEILLANCE – LE RRFR – RESEAU RECURRENT A BASE DE FONCTIONS RADIALES (RRBF – RECURRENT RADIAL BASIS FUNCTION)	105
4.5.1 <i>Architecture du RRFR</i>	106
4.5.2 <i>Mémoire dynamique du réseau RRFR : Architecture LRGF</i>	107
4.5.3 <i>Analyse comparative des trois principaux types de mémoires dynamiques du réseau RRFR</i>	117
4.6 CONCLUSION ET SYNTHESE	118

CHAPITRE 5 - EVALUATION DES PERFORMANCES, ETUDE COMPARATIVE ET EXPLOITATION INDUSTRIELLE DU RRFR..... 121

5.1	INTRODUCTION	121
5.2	EVALUATION DES PERFORMANCES DU RRFR	122
5.2.1	<i>Reconnaissance de séquences temporelles</i>	122
5.2.2	<i>La prédiction temporelle pour le pronostic</i>	128
5.2.3	<i>La reproduction de séquences</i>	136
5.2.4	<i>Bilan de l'évaluation des performances du RRFR</i>	138
5.3	ETUDE COMPARATIVE DU RRBF PAR RAPPORT AUX STRUCTURES DE RESEAUX DE NEURONES TEMPORELS LES PLUS SIGNIFICATIFS, DANS L'OPTIQUE DE L'UTILISATION EN SURVEILLANCE DYNAMIQUE.....	138
5.3.1	<i>Choix des structures neuronales temporelles étudiées</i>	139
5.3.2	<i>Etude comparative dans le domaine du pronostic</i>	141
5.3.3	<i>Etude comparative dans le domaine de la surveillance dynamique</i>	143
5.3.4	<i>Bilan de l'étude comparative des différentes structures neuronales temporelles</i>	145
5.4	EXPLOITATION INDUSTRIELLE DE LA SOLUTION DE SURVEILLANCE DYNAMIQUE PAR RRFR.....	146
5.4.1	<i>Introduction</i>	146
5.4.2	<i>Choix de la solution de surveillance dynamique en temps réel</i>	147
5.4.3	<i>Description de la solution proposée</i>	148
5.4.4	<i>Evaluation des performances du prototype</i>	152
5.4.5	<i>Bilan de l'exploitation industrielle</i>	153
5.5	CONCLUSION ET PERSPECTIVES	154
5.5.1	<i>Principales contributions</i>	154
5.5.2	<i>Perspectives</i>	155

CHAPITRE 6 - SURVEILLANCE DYNAMIQUE PAR RESEAUX DE PETRI FLOUS 157

6.1	INTRODUCTION	157
6.2	ETAT DE L'ART DES RESEAUX DE PETRI FLOUS. UTILISATION DES RESEAUX DE PETRI FLOUS POUR LA SURVEILLANCE DES SYSTEMES A EVENEMENTS DISCRETS	157
6.3	OUTIL DE SURVEILLANCE PAR APPROCHE FLOUE. APPLICATION A LA SURVEILLANCE DYNAMIQUE.....	159
6.3.1	<i>Introduction</i>	159
6.3.2	<i>Description et définition de l'outil RdPFS</i>	160
6.3.3	<i>Modélisation des règles logiques de l'arbre de défaillance par RdPFS</i>	164
6.3.4	<i>Exemple de modélisation en utilisant le RdPFS</i>	164
6.4	ALGORITHME DYNAMIQUE DE SURVEILLANCE PAR APPROCHE FLOUE.....	165
6.4.1	<i>Approche temporelle floue pour la surveillance des systèmes à événements discrets</i>	166
6.4.2	<i>Dynamique des transitions dans le réseau de Petri flou pour la surveillance</i>	167
6.4.3	<i>Algorithme d'évaluation dynamique des défauts</i>	171
6.5	EXEMPLE DE SURVEILLANCE DYNAMIQUE PAR RDPFS.....	173
6.6	ACTION DE MAINTENANCE EN UTILISANT UNE BASE DE REGLES FLOUES.....	177
6.6.1	<i>Approche floue dans la fonction de maintenance</i>	177
6.7	APPLICATION TEMPS REEL DE SURVEILLANCE DANS UNE SOLUTION LOGICIELLE DE E-MAINTENANCE.....	178
6.7.1	<i>Système étudié</i>	178
6.7.2	<i>Implémentation des fonctions détection et surveillance dynamique</i>	179

6.7.3	<i>Implémentation de la fonction de maintenance (reprise)</i>	181
6.7.4	<i>Intégration de l'approche globale de surveillance floue dans une plateforme logicielle</i>	182
6.8	CONCLUSION ET PERSPECTIVES.....	184
CHAPITRE 7 - VERS UN OUTIL HYBRIDE NEURO-FLOU POUR LA SURVEILLANCE DYNAMIQUE		187
7.1	INTRODUCTION.....	187
7.2	GENERALITES SUR LES SYSTEMES NEURO-FLOUS	187
7.2.1	<i>Avantages et inconvénients de la logique floue et des réseaux de neurones</i>	188
7.2.2	<i>Méthodes et combinaisons neuro-floues</i>	189
7.2.3	<i>systèmes neuro-flous hybrides</i>	191
7.2.4	<i>Méthodologie de modélisation neuro-floue</i>	196
7.2.5	<i>Approche neuro-floue en classification</i>	196
7.3	CONCEPTION D'UN OUTIL NEURO-FLOU POUR L'AIDE AU DIAGNOSTIC	197
7.3.1	<i>Utilisation des systèmes neuro-flous pour le diagnostic</i>	197
7.3.2	<i>Spécifications de l'outil en utilisant la démarche UML</i>	199
7.3.3	<i>Description de l'outil neuro-flou utilisé</i>	205
7.3.4	<i>Exemple d'application de l'outil d'aide au diagnostic / pronostic</i>	207
7.3.5	<i>Prototypage de l'outil neuro-flou d'aide au diagnostic</i>	209
7.4	CONCLUSION ET PERSPECTIVES	212
CONCLUSIONS GENERALES ET PERSPECTIVES DE RECHERCHE		213
BILAN, CONCLUSIONS		213
PERSPECTIVES DE RECHERCHE		215
TABLE DES FIGURES		217
LISTE DES TABLEAUX		221
REFERENCES BIBLIOGRAPHIQUES		223
INDEX		241

Notations et abréviations

<i>ADD</i>	<i>Arbre de défaillances</i>
<i>AMDEC</i>	<i>Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticité</i>
<i>CG</i>	<i>Mécanisme de type "chien de garde"</i>
<i>F</i>	<i>Fonction d'appartenance correspondant au fonctionnement défaillant</i>
<i>FN</i>	<i>Fonction d'appartenance correspondant au fonctionnement normal</i>
<i>HMI</i>	<i>Human Machine Interface</i>
<i>ISO</i>	<i>International Organization for Standardization</i>
<i>GdC</i>	<i>Graphe de Causalité</i>
<i>GMAO</i>	<i>Gestion de la Maintenance Assistée par l'Ordinateur (Central Maintenance Management System - CMMS)</i>
<i>MIMOSA</i>	<i>Machinery Information Management Open Systems Alliance</i>
<i>PMC</i>	<i>Perceptron Multicouches (Multi Layer Perceptron - MLP)</i>
<i>RàPC</i>	<i>Raisonnement à Partir de Cas (Case Based Reasoning – CBR)</i>
<i>RdP</i>	<i>Réseau de Petri</i>
<i>RdPF</i>	<i>Réseau de Petri Flou</i>
<i>RdPFM</i>	<i>Réseau de Petri flou pour la maintenance</i>
<i>RdPFS</i>	<i>Réseau de Petri flou pour la surveillance</i>
<i>RdPO</i>	<i>Réseau de Petri à Objets</i>
<i>RdPTO</i>	<i>Réseau de Petri Temporel à Objets</i>
<i>RFR (RBF)</i>	<i>Réseau à Fonctions de base Radiales (Radial Basis Function - RBF)</i>
<i>RRFR(RRBF)</i>	<i>Réseau Récurrent à Fonctions de base Radiales (Recurrent Radial Basis Function - RRBF)</i>
<i>SCADA</i>	<i>Supervisory Control and Data Acquisition (système de supervision et d'acquisition de données des processus industriels)</i>
<i>SED</i>	<i>Système à événements discrets</i>
<i>SIF</i>	<i>Système d'inférence flou</i>
<i>TG</i>	<i>Transition du chien de garde</i>



Partie I - Parcours Professionnel

Chapitre 1 - Curriculum Vitae

1.1 RENSEIGNEMENTS GENERAUX

1.1.1 ETAT CIVIL

Daniel Racoceanu

Né le 11 juin 1968 à Petrosani, Roumanie

Nationalité française

Coordonnées personnelles : 13 rue les hauts du Vorin
25770 Serre les Sapins
Tél. : 03 81 48 20 80

Coordonnées professionnelles : Laboratoire d'Automatique de Besançon
LAB UMR 6596
UFR Sciences et Techniques
24 rue Alain Savary
25000 Besançon
Tél. : 03 81 40 27 92 / *Fax :* 03 81 40 28 09
Courriel : daniel.racoceanu@ens2m.fr

1.1.2 PARCOURS PROFESSIONNEL

- du 01/09/1993 au 31/08/1994 : Enseignant vacataire dans le département OGP de l'IUT de Belfort - Montbéliard, Université de Franche-Comté,

- du 01/09/1994 au 31/08/1996 : ATER à l'IUT de Belfort - Montbéliard, Université de Franche-Comté, départements organisation et gestion de la production et génie télécommunication réseaux,

- 1997-1998 : responsable logistique - ordonnancement, société Gaussin S.A., Héricourt (70), concepteur et fabricant d'automoteurs industriels, systèmes spéciaux de manutention pour l'industrie, les ports et aéroports. CA : 10 M€, 78 personnes,

- 1998-1999 : responsable d'affaires, Alstom Gaz Turbines / General Electric France, Belfort (90), Concepteur et fabricant de turbines à gaz, CA : 1,2 Md€, 1800 personnes,

- 1999-2005 : maître de conférences à l'UFR des Sciences et Techniques de l'Université de Franche-Comté, chercheur au Laboratoire d'Automatique de Besançon – LAB UMR CNRS 6596.

- depuis le 01/09/2005 : en délégation CNRS au laboratoire Image Processing Application Lab – IPAL FRE CNRS 2339 de Singapour

1.1.3 ETUDES, TITRES, DIPLOMES

Juin 1986 Baccalauréat, série C

(1986 – 1987 Service militaire)

Juin 1992 Diplôme d'Ingénieur de l'Ecole Polytechnique de Timisoara, Roumanie, spécialité Génie Mécanique et Productique, option Mécatronique

Juin 1993 DEA en Automatique et Productique, Ecole Nationale d'Ingénieurs de Belfort, major de promotion, mention Très Bien

Janvier 1997 Thèse de Doctorat en Automatique et Informatique à l'Université de Franche-Comté, soutenue à l'Ecole Nationale d'Ingénieurs de Belfort (ENIBe), mention Très Honorable avec les Félicitations du Jury

1.1.4 THESE

Thèse de Doctorat de l'Université de Franche-Comté, intitulée « Contribution à la modélisation et à l'analyse des Chaînes de Markov à échelles de temps et échelles de pondérations multiples. Application à la gestion d'un système hydro énergétique EDF », soutenue le 10 janvier 1997 au Laboratoire de Mécanique et Productique de l'Ecole Nationale d'Ingénieurs de Belfort, spécialité « Automatique et Informatique », mention Très Honorable avec les Félicitations du Jury.

Directeur scientifique : Prof. A. El Moudni (ENIBe – puis UTBM), actuellement directeur du laboratoire SET (Systèmes et Transport) de l'UTBM.

Jury : Prof. Lj T. Grujic (ENIBe – puis UTBM - président du jury), Prof. I. Zambettakis (Institut Universitaire de Technologie - Université Paul Sabatier - Toulouse III - rapporteur), Prof. G. L. Gissinger (Université de Haute Alsace - ESSAIM Mulhouse - rapporteur), Prof. G. Draghici (Université Polytechnique de Timisoara – rapporteur), Prof. J-M. Henrioud (Université de Franche-Comté), Prof. N. Zerhouni (ENIBe – UTBM - ENSMM), Prof. M. Ferney (ENIBe - UTBM), M. J. Mesnier (responsable EDF).

1.2 PARCOURS SCIENTIFIQUE

Mes activités de recherche ont débuté en octobre 1992 au sein du Laboratoire de Mécanique et Productique de l'Ecole Nationale d'Ingénieurs de Belfort (ENIBe). Dans cette première partie de mon activité de recherche, mes travaux se sont orientés vers la simplification des modèles de systèmes à événements discrets complexes, sous la direction du Professeur A. El Moudni. C'est ainsi que nous avons développé une méthode innovante de simplification des chaînes de Markov. Cette méthode, inspirée des techniques de perturbations singulières, a été adaptée aux modèles probabilistes, en permettant de mettre en évidence l'échelle de pondération multiple de ces modèles dans leur comportement permanent. Nous avons ainsi proposé une réduction des modèles markoviens de grandes dimensions, en permettant de garder uniquement l'ensemble des états caractérisés par une probabilité d'état importante en régime permanent (long terme). Les applications développées concernent l'étude de la fiabilité opérationnelle des systèmes de production et l'optimisation de la gestion d'un système hydro énergétique menée en collaboration avec EDF. Ces travaux de recherche ont permis une poursuite des travaux de l'axe dans le domaine de la simplification des réseaux de Petri Stochastiques basée sur les mêmes principes (Amodeo, 1999).

A partir de septembre 1999, date de mon arrivée au Laboratoire d'Automatique de Besançon (LAB UMR 6596), j'ai participé à la création de l'équipe « *ingénierie de maintenance et e-maintenance* », équipe travaillant sous la direction du Professeur N. Zerhouni, dans le cadre du groupe « Conception et Commande des Systèmes de Production ». Profitant des connaissances acquises dans le domaine de la fiabilité lors de mes travaux de thèse et sensibilisé par mon expérience industrielle sur l'importance accrue de la Fonction Maintenance, j'anime depuis septembre 1999 – au sein de cette équipe – les activités de l'axe « Surveillance en utilisant des techniques de l'Intelligence Artificielle ».

Mon arrivée au LAB s'est donc accompagnée d'une reconversion thématique non pas totale, compte tenu de la connotation productique des thèmes de l'équipe « *ingénierie de maintenance et e-maintenance* », mais néanmoins sensible, due à la nécessité de réaliser un état de l'art des techniques de l'intelligence artificielle susceptibles d'apporter une plus-value intéressante au domaine de la surveillance.

L'objectif de l'axe surveillance par IA consiste à créer un pôle de compétence autour de la surveillance industrielle, en proposant des outils de plus en plus souples, adaptatifs et facilement reconfigurables. En effet, les sujets abordés dans nos différents partenariats (partenariats industriels, projets européens, nationaux, régionaux, contrats de recherche,...) dans le domaine de la maintenance, ont mis en évidence la nécessité de mettre au point des outils de surveillance flexibles, faciles à initialiser, configurer et reconfigurer, capables de prendre en compte l'expérience humaine, le raisonnement imprécis et de mettre à jour leurs informations en temps réel pour une efficacité optimale. Dans ce contexte, nos travaux de recherche se sont orientés essentiellement vers :

- ✚ la détection dynamique des défauts,
- ✚ le diagnostic,
- ✚ le pronostic,

problématiques qui constituent les verrous technologiques du métier. Du point de vue scientifique, l'apprentissage dynamique, l'autoapprentissage ainsi que la prise en compte du raisonnement humain nous ont mené naturellement vers des techniques innovantes de l'intelligence artificielle comme :

- ✚ les réseaux de neurones dynamiques,
- ✚ les réseaux de Petri flous et
- ✚ les systèmes neuro-flous.

1.3 RESULTATS SCIENTIFIQUES MAJEURS

1.3.1 SIMPLIFICATION DES SYSTEMES A EVENEMENTS DISCRETS COMPLEXES

(d'octobre 1992 à janvier 1997)

Les travaux que j'ai menés à Belfort jusqu'en janvier 1997 ont notamment conduit aux résultats suivants, correspondant aux publications en revues internationales [MMS, 1995], [SAMS, 1994] et en conférences internationales [IFAC, 1997], [IMACS, 1995], [IEEE ICRA, 1994], [MATHMOD, 1994], indiquées au paragraphe 1.8 :

- ✚ méthode innovante de simplification des modèles markoviens de grande dimension, en introduisant la notion d'échelle de pondération multiple de ce type de modèles probabilistes situés « à la limite de la stabilité ». Cette nouvelle

méthode est inspirée de la propriété d'échelle de temps multiple associée aux techniques des perturbations singulières des systèmes stables.

- ✚ application de la méthode de simplification proposée, à l'étude de la fiabilité opérationnelle des systèmes de production, en mettant en évidence les nouveaux points de vue engendrés par l'étude préliminaire (dans le cadre du système réduit) des états à fortes probabilités en régime permanent (comportement à long terme du système).
- ✚ étude et optimisation de la commande d'un système hydro énergétique EDF situé sur le cours supérieur du Doubs, en utilisant la double échelle de pondération du modèle probabiliste caractérisant le système. L'étude prend en compte des critères d'optimisation globale de la productivité d'énergie électrique et des critères de protection de l'environnement (débit réservé, minimisation des variations de niveau d'eau, etc.). Cette étude, menée en collaboration avec EDF (rapports [EDF, 1995, 1996] – paragraphe 1.8.6) a nécessité une étude statistique de 3 ans et a donné suite à une gestion globale automatisée des centrales, gérées jusqu'à lors individuellement.

1.3.2 INGENIERIE DE LA MAINTENANCE ET E-MAINTENANCE

(à partir de septembre 1999)

Les travaux que j'ai menés dans l'équipe « *ingénierie de maintenance et e-maintenance* » au LAB à Besançon, ont notamment conduit aux résultats les plus marquants suivants :

- ✚ établissement d'un modèle neuronal dynamique – le RRFR - réseau récurrent à base de fonctions radiales (en anglais RRBF – recurrent radial basis function) capable d'effectuer de la détection dynamique de défauts. Cet outil, basé sur le principe du réseau de neurones de type RFR (RBF), présente les avantages de celui-ci (approche locale, apprentissage dynamique, capacité de quantifier le degré de pertinence de sa classification) en y rajoutant une mémoire dynamique, matérialisée par un neurone bouclé. Ce nouveau réseau de neurones dynamiques possède entre autres la capacité d'analyser l'évolution d'un signal, de reconnaître les fausses alarmes et de détecter de manière précoce les paliers de dégradation, qualités très recherchées dans les systèmes modernes de surveillance.
- ✚ réalisation d'un prototype de système de surveillance temps réel basé sur les réseaux de neurones dynamiques RRFR, accessible à distance par un serveur web. Le prototype a été réalisé en collaboration avec la société Avensy Ingénierie de Besançon, spécialiste dans le suivi de production, automatisme et informatique industrielle. Il a donné lieu à un dépôt de brevet d'invention ENSMM-AVENSY intitulé : « *Système de surveillance temps-réel basé sur les réseaux de neurones dynamiques, accessible à distance par un serveur web* »¹.
- ✚ étude et mise au point d'un modèle de surveillance dynamique basé sur les réseaux de Petri flous et baptisé réseau de Petri flou pour la surveillance (RdPFS). Le principe de cet outil consiste à transformer un arbre de défaillance

¹ Auteurs : D. Racocanu, R. Zemouri, N. Zerhouni, R. Labourier, dépositaires du brevet : ENSMM (50%) et Avensy (50%), date de dépôt du brevet : 30 septembre 2002, numéro d'enregistrement : 021 2205. Ce brevet est actuellement en cours de réactualisation avec la société Avensy, dans une nouvelle optique marketing.

en une structure de type réseau de Petri, en y ajoutant un marquage flou et en y introduisant des règles spécifiques de propagation des défauts. Par conséquent, l'arbre de défaillance devient un outil dynamique de surveillance, capable – par l'intermédiaire de son marquage flou évolutif – de prédire une défaillance avant qu'elle n'apparaisse effectivement.

- ✚ prototypage Labview basée sur l'approche homogène réseaux de Petri flous avec essais de surveillance entre le site de Besançon (système flexible avec système d'acquisition) et le site de l'Université Valahia de Targoviste (système distant de surveillance).
- ✚ étude et spécification d'un système neuro-flou d'aide au diagnostic. L'utilisation conjointe du principe neuronal et de la logique floue permettent de concentrer les avantages des deux formalismes : la capacité d'apprentissage des réseaux de neurones et la prise en compte des incertitudes de la logique floue. Les spécifications de ce système d'aide au diagnostic – effectuées dans le langage UML – ont fait l'objet de livrables dans le cadre du projet européen ITEA – PROTEUS² [PROTEUS-D2.1, 2003], [PROTEUS-D2.2, 2003], [PROTEUS-D2.3, 2004].
- ✚ prototypage d'un système d'aide au diagnostic, basé sur les approches neuro-floues développées, en utilisant le langage Labview (National Instrument) et des systèmes embarqués de type carte FPGA (système PXI), dans le cadre d'un projet ANVAR [ANVAR, 2004].

1.4 ANIMATION DE L'AXE SURVEILLANCE PAR INTELLIGENCE ARTIFICIELLE

Dans le cadre de mes travaux de recherche au LAB et la mise en place de l'axe « Surveillance par Intelligence Artificielle » de l'équipe « *ingénierie de maintenance et e-maintenance* », je co-encadre actuellement trois doctorants, après avoir co-encadré deux thèses déjà soutenues. Les travaux de ces chercheurs seront évoqués dans les chapitres relatifs à la recherche.

Il est à noter que j'ai élaboré et rédigé tous les sujets de ces thèses, sujets validés par le Professeur N. Zerhouni. Par ailleurs, les deux doctorants co-encadrés ayant soutenu leur thèse, sont actuellement les deux, maîtres de conférences, l'un au CNAM et l'autre à l'Université Valahia de Targoviste, en Roumanie.

Enfin, depuis 1999, j'ai encadré et co-encadré un nombre de 15 stages DEA–MASTER Recherche (moyenne entre 2 et 3 stages d'initiation à la recherche encadrés par an). Ces travaux ont eu comme objectif d'initier des nouvelles directions de recherche dans l'axe et d'approfondir certains concepts abordés. Mis à part l'impact sur la dynamique globale de recherche de l'équipe et du laboratoire, ces travaux ont mené directement à un nombre de 3 communications à des congrès internationaux prestigieux [IFAC-MCPL, 2004], [IEEE-RAM, 2004], [IEEE-ICRA, 2002] et de 2 communications à des congrès internationaux à audience francophone [IAI, 2003a, b]. Par ailleurs, la grande majorité des stagiaires encadrés sont actuellement ingénieurs, doctorants ou titulaires d'un doctorat.

² PROTEUS a generic platform for e-maintenance (plate-forme générique de e-maintenance) - <http://www.proteus-iteaproject.com>

1.4.1 CO-ENCADREMENT DE THESES

Il est à noter que j'ai élaboré et rédigé tous les sujets de ces thèses, sujets validés par le Professeur N. Zerhouni.

a) THESES SOUTENUES

Les 3 titulaires des doctorats co-encadrés soutenus, sont actuellement maîtres de conférences pour E. Minca et R. Zemouri et respectivement post-doctorant pour N. Palluat.

R. Zemouri

« Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance »

thèse commencée en octobre 2000, **soutenue le 28 novembre 2003**

*Jury : A. Bourjault (président), D. Hammad (rapporteur), D. Noyes (rapporteur), J-M. Faure, **D. Racocceanu**, N. Zerhouni, R. Labourier (PDG Avenisy Ingénierie - industriel invité)*

mention très honorable (au LAB, la mention THFJ n'est plus en vigueur depuis octobre 2003)

financement : allocation ministère de la Recherche

taux d'encadrement : 50% (co-encadrement avec N. Zerhouni 50%)

situation actuelle du diplômé : docteur de l'Université de Franche-Comté et maître de conférences au CNAM Paris - chercheur au LURPA Cachan

E. Minca

« Contribution à la surveillance des systèmes de production à l'aide des réseaux de Petri flous : Application à la e-maintenance »

thèse commencée en octobre 2001 et **soutenue le 24 septembre 2004**

*Jury : I. Cucui (président), C. Sibertin-Blanc (rap.), V. Manzu (rapp.), A. Dorin, **D. Racocceanu**, N. Zerhouni, J. Cardoso (invité).*

mention très honorable,

thèse de doctorat en co-tutelle, Université de Franche-Comté - Université Valahia de Targoviste, Roumanie,

financement – Ambassade de France en Roumanie

taux d'encadrement : 50% (co-encadrement avec N. Zerhouni 30% et D. Alexandru – Université Valahia de Targoviste, Roumanie 20%)

situation actuelle de la diplômée : docteur de l'Université de Franche-Comté et maître de conférences à l'Université Valahia de Targoviste (Roumanie)

N. Palluat

« Méthodologie de surveillance dynamique à l'aide des réseaux neuro-flous temporels »

thèse commencée en octobre 2002 et **soutenue le 12 janvier 2006**

*Jury : C. Chu (président), P. Brézillon (rapp.), B. Ould-Bouamama (rapp.), A. Bourjault, J-M. Olive, **D. Racocceanu**, N. Zerhouni, J-B Leger (PDG Predict - industriel invité).*

mention très honorable,

financement : allocation ministère de la Recherche

taux d'encadrement : 60% (co-encadrement avec N. Zerhouni 40%)

situation actuelle de la diplômée : docteur de l'Université de Franche-Comté, post-doc à l'Université Fédérale Santa Catarina, Florianópolis (Brésil)

b) THESES EN COURS DE PREPARATION

A. Ghernaut - « *Aide au diagnostic des systèmes mécatroniques en utilisant une approche duale maintenance-qualité. Application chez Cegelec France* »

thèse de doctorat d'état commencée en 2003, soutenance prévue octobre 2006

financement : bourse de coopération algéro-française, département de mécanique, faculté des sciences de l'ingénieur, université de Tlemcen-Algérie, en collaboration avec le LAB.(Besançon)

taux d'encadrement : 60% (co-encadrement avec N. Zerhouni 40%)

O. Vasile (épouse Dragomir) - « *Optimisation des systèmes de surveillance, contrôle et sécurité dans les bâtiments industriels à l'aide des algorithmes de l'intelligence artificielle* »

thèse de doctorat en co-tutelle avec l'Université Valahia de Targoviste (habilitée par l'Université Polytechnique de Bucarest), Roumanie, commencée en octobre 2005

financement : Ambassade de France en Roumanie.

taux d'encadrement : 40% (co-encadrement avec N. Zerhouni 30% et S. Iliescu – Université Polytechnique de Bucarest, Roumanie 30%)

1.4.2 ENCADREMENT DE STAGES DEA ET MASTER RECHERCHE

Il est à remarquer que parmi les stagiaires DEA / M2R encadrés, 3 sont actuellement docteurs (1 maître de conférences, 1 post-doc, 1 ATER), 7 doctorants et 3 ingénieurs :

R. Zemouri, *Modélisation et simplification des processus stochastiques : application à l'optimisation de la maintenance chez PSA Peugeot Citroën*, DEA IAP (Informatique, Automatique et Productique) de l'Université de Franche-Comté, septembre 2000

taux d'encadrement : 50% (co-encadrement avec N. Zerhouni 50%)

situation actuelle du diplômé : docteur de l'Université de Franche-Comté et maître de conférences au CNAM Paris - chercheur au LURPA Cachan

N. Addouche, *Modélisation et Evaluation des Performances d'un Système de Production intégrant la Télémaintenance Corrective et Préventive*, DEA IAP de l'Université de Franche-Comté, septembre 2001

taux d'encadrement : 50% (co-encadrement avec N. Zerhouni 50%)

situation actuelle de la diplômée : doctorant à l'INPG (Grenoble)

N. S. Ouldyahia, *Diagnostic des systèmes de production en utilisant les réseaux de neurones. Introduction au neuro-flou*, DEA IAP de l'Université de Franche-Comté, septembre 2001

taux d'encadrement : 100%

situation actuelle de la diplômée : doctorant à l'UTT (Troyes)

J. Kaabi, *Maintenance prédictive systématique basée sur la fiabilité. Application pour la gestion des pinces à souder robotisées chez PSA Peugeot Citroën Sochaux*, DEA IAP, septembre 2001

taux d'encadrement : 50% (co-encadrement avec N. Zerhouni 50%)

situation actuelle de la diplômée : docteur de l'UFC (LAB). ATER à l'Ecole des Mines de Nancy

S. Pross, *Propagation des incertitudes dans un arbre de défaillances. Application au système flexible du LAB dans le cadre de la plate-forme de télémaintenance associée*, DEA IAP, septembre 2001

taux d'encadrement : 100%

situation actuelle de la diplômée : ingénieur d'affaires – Electricité Industrielle de l'Est - Mulhouse

N. Palluat, *Méthodologies neuronales de diagnostic des défaillances. Intégration des procédures neuronales dans la plate-forme de télémaintenance du LAB*, DEA IAP de l'Université de Franche-Comté, septembre 2002

taux d'encadrement : 70% (co-encadrement avec N. Zerhouni 30%)

Situation actuelle du diplômé : doctorant au LAB

A.D. Djouak, *Etude d'un multicateur distribué à base de réseaux de neurones et de traitement de l'image*, DEA IAP de l'Université de Franche-Comté, septembre 2002

taux d'encadrement : 100%

situation actuelle du diplômé : doctorant au CEMIF, Evry Val d'Essonne

O.A. Djaffar, *Utilisation des réseaux neuro-flous pour le pronostic industriel*, DEA IAP de l'Université de Franche-Comté, septembre 2002

taux d'encadrement : 100%

situation actuelle du diplômé : doctorant au Laboratoire MIPS, ESSEAIM, Université de Haute Alsace

S. Ouaret, *Conception et programmation Java d'une interface de surveillance en ligne via Internet à l'aide d'un automate Schneider avec coupleur serveur web. Application à la reconnaissance des séquences d'évènements discrets*, DEA IAP de l'Université de Franche-Comté, septembre 2003

taux d'encadrement : 100%

situation actuelle du diplômé : Etudes doctorales à l'Ecole Polytechnique de Montréal, Québec, Canada

H. Sefiane, *Mise au point d'un algorithme et développement d'un prototype de superviseur pour la surveillance et la commande des systèmes flexibles de production en utilisant les réseaux de Petri flous et le langage Java*, DEA IAP de l'Université de Franche-Comté, septembre 2003

taux d'encadrement : 100%

situation actuelle du diplômé : doctorant au laboratoire LAMIH-UMR 8530, Université de Valenciennes

T. Montroig, *E-maintenance. Etude générique des plates-formes de e-maintenance distribuées. Application : prototypage du démonstrateur de e-maintenance du LAB dans le cadre du projet européen Proteus*, DEA IAP de l'Université de Franche-Comté, septembre 2003

taux d'encadrement : 50% (co-encadrement avec N. Zerhouni 50%)

situation actuelle du diplômé : ingénieur Cegelec S.A. France

M. Monnin, *Diagnostic et pronostic industriel en utilisant les techniques de l'intelligence artificielle. Application à la plate-forme de e-maintenance du LAB*, DEA IAP de l'Université de Franche-Comté, juin 2004

taux d'encadrement : 80% (co-encadrement avec N. Zerhouni 20%)

situation actuelle du diplômé : doctorant bourse DGA au laboratoire LAMIH-UMR 8530 – Université de Valenciennes et du Hainaut Cambrésis

J. Xu, *Utilisation des réseaux de neurones dynamiques pour la e-maintenance. Instrumentation d'une maquette de surveillance temps réel*, DEA IAP de l'Université de Franche-Comté, septembre 2004

taux d'encadrement : 100%

H. Wei, *Arbre de défaillance flou*, Master Recherche Mécatronique, date prévisionnelle de fin : septembre 2005

taux d'encadrement : 50% (co-encadrement avec N. Palluat 50%)

S. Brust, *Détection de défauts en utilisant les techniques neuronales dynamiques. Prototypage en Labview*, Master Recherche Mécatronique, mai 2005.

taux d'encadrement : 100%

situation actuelle : ingénieur R&D, Daimler Chrysler AG, Allemagne.

1.5 ANIMATION DE LA RECHERCHE

1.5.1 AU NIVEAU REGIONAL

Au niveau régional, l'équipe « *ingénierie de maintenance et e-maintenance* » du LAB a inscrit une de ses actions majeures dans le cadre de l'Institut des Sciences et Technologies de l'Information (ISTI) de Franche-Comté. Dans le cadre du soutien aux actions de recherche, l'ISTI regroupe des projets de la région de Franche-Comté concernés par les Sciences et Technologies de l'Information et de la Communication (STIC). L'Institut fait ainsi valoir essentiellement l'implication de plusieurs laboratoires de recherche afin de renforcer le caractère pluridisciplinaire des actions qu'il soutient.

Dans ce contexte, en partenariat avec le Laboratoire d'Informatique de Franche-Comté (LIFC – FRE CNRS 2661), l'INRIA et les entreprises Cegelec, Schneider, Nipson, notre équipe est à l'origine d'un projet de télémaintenance industrielle coopérative (acronyme SYMPATIC³). Cette collaboration nous a permis de constituer tout au long de notre parcours un capital savoir-faire très utile, exploité dans différents autres projets ou propositions de projets (projet européen Proteus, projet NEMOSYS⁴ avec la Division des Constructions Navales, proposition de projets européens ARCOIRIS⁵, PROMISE⁶, etc.).

L'objectif principal du projet SYMPATIC est de concevoir et de développer une solution d'expertise et d'aide à la décision pour les responsables de la maintenance des installations industrielles. Par le biais du réseau de communication, ces outils permettent aux ingénieurs d'accéder à distance aux informations pertinentes et de disposer

³ SYMPATIC - SYstème d'aide à la décision pour la gestion intégrée des fonctions de Maintenance et Production à l'Aide des Technologies de l'Information et de la Communication

⁴ Plus de renseignements dans le paragraphe suivant, dédié aux collaborations nationales.

⁵ ARCOIRIS - Towards the smart and flexible factory: Intelligent production unit networks for manufacturing optimisation and management, FP6 Integrated Project - Integrating Technologies for the Fast and Flexible Manufacturing Enterprise, FP6 - IST-NMP joint call – projet élaboré en 2004.

⁶ PROMISE - PRomote and Optimise manufacturing and Maintenance with Intelligent system for human Safety on Exploitation – Competitive and Sustainable Growth – FP5 - RTD Framework Programme - <http://www.cordis.lu/fp5/>, projet élaboré en 2000.

d'indicateurs et de tableaux de bord leur permettant d'évaluer l'efficacité des services concernés par la maintenance et d'aider à la prise de décision. Grâce à son aspect coopératif, cette aide à la décision prend ainsi une dimension stratégique très importante.

La contribution de notre équipe dans le cadre de ce projet, concerne l'ingénierie de maintenance. La problématique à résoudre relève d'une part de l'élaboration et de l'exploitation des bases de données temps réel pour l'expertise, afin d'analyser la pertinence d'une solution de maintenance. D'autre part, en présence d'une défaillance, il s'agit de proposer une aide au diagnostic rapide et fiable ainsi que de proposer la meilleure solution en termes de stratégie de maintenance. En ce sens, le projet se propose de matérialiser une plate-forme technologique basée sur une architecture distribuée et des modèles de référence ainsi qu'une méthodologie prenant en compte aussi bien des connaissances "métier" que scientifiques. La plate-forme matérielle et logicielle et s'oriente autour de deux axes principaux :

- *Elaboration et mise en place d'interfaces entre les niveaux managérial, décisionnel et opérationnel pour l'expertise,*
- *Développement et intégration de systèmes classiques et de systèmes intelligents et coopératifs pour l'élaboration des indicateurs, le télédiagnostic et la télémaintenance.*

En tant que membre du groupe « Conception et commande des systèmes de production » du LAB, je me suis naturellement impliqué dans les actions de l'Institut de Productique de Besançon. Dans ce contexte, j'ai participé (en 2000) à un groupe de réflexion sur l'impact des TIC⁷ sur la Productique, ainsi qu'à l'organisation de journées techniques impliquant un nombre important d'industriels et universitaires de la région :

- Thématique : « *Sûreté de fonctionnement des systèmes de production* », 4 mai 2000, à l'Institut de Productique de Besançon ;
- Thématique : « *De l'ingénierie de maintenance à l'e-maintenance : réalités industrielles* », 22 mai 2003, au Centre de Développement du Multimédia de Montbéliard ;
- Thématique : « *Les atouts de la e-maintenance pour la performance industrielle* », 19 mai 2005, chez Cegelec Belfort.

Dans le cadre de la journée du 19 mai 2005, j'ai ainsi effectué une présentation sur les techniques de l'intelligence artificielle appliquées au diagnostic et au pronostic [IP, 2005].

Dans le même objectif de diffusion de l'information auprès des entreprises régionales, deux articles parus dans l'hebdomadaire « En Direct Productique » traitent des sujets issus de la recherche et susceptibles de donner lieu à des exploitations industrielles. J'ai ainsi co-édité 2 communications : une première autour de notre collaboration avec la société Avensy dans le domaine de la surveillance par réseaux de neurones dynamiques [EDP, 2004] et une deuxième autour de nos travaux pour le compte du projet NEMOSYS avec la DCN [EDP, 2005].

1.5.2 AU NIVEAU NATIONAL

La communauté française travaillant autour de la maintenance coopérative et distribuée s'est organisée récemment dans un groupe de travail GT MACOD (modélisation et optimisation de la MAintenance COopérative et Distribuée) faisant partie du groupe de recherche GdR MACS (Modélisation, Analyse et Conduite des Systèmes dynamiques)

⁷ TIC – Technologies de l'Information et de la Communication

labellisé par le CNRS. Dans ce cadre, nous avons ainsi présenté et discuté nos premiers résultats de recherche [GRP, 2002], [MACS, 2003] et [MACS, 2005] afin de positionner nos travaux au sein de la communauté et d'ouvrir la voie à de nouvelles collaborations.

Le groupe MACOD – dont le LAB est un des membres les plus actifs – s'est remarqué par une activité importante, marquée notamment par la signature en 2004 d'un contrat de recherche – NEMOSYS⁸ – avec la Division des Constructions Navales (DCN). Dans ce projet d'envergure, impliquant des partenaires universitaires et industriels de MACOD, notre équipe – par l'intermédiaire de son responsable, le Professeur N. Zerhouni, assure – dans le cadre de l'Institut de Productique – la coordination scientifique.

Enfin, le groupe MACOD s'est révélé aussi un cadre très propice à une réflexion de fond dans le domaine de la maintenance moderne, en mettant assez rapidement en évidence les verrous scientifiques et technologiques de ce domaine d'activité. Une conséquence directe de cette réflexion est le projet actuel de rédaction d'un ouvrage commun sur la e-maintenance.

1.5.3 AU NIVEAU INTERNATIONAL

Le co-encadrement de la thèse en co-tutelle de Mme. Minca (2004) avec la Roumanie, s'inscrit dans une coopération plus vaste, comprenant la mise en place des bases d'une formation commune de type master européen en mécatronique et des échanges de type Erasmus, échanges dont je suis le responsable au sein de mon service d'enseignement (GAP⁹) de l'UFR Sciences. Dans ce contexte, des interventions au niveau master ont déjà eu lieu en 2002 et en 2005 par des collègues du GAP. Par ailleurs, de nouvelles collaborations avec des industriels franco-roumains ou européens ont été initiées, notamment avec Dacia-Renault et Schneider Roumanie. Ces collaborations connaissent un soutien actif de la part des partenaires roumains et de l'Ambassade de France à Bucarest, soutien est marqué par le financement d'une nouvelle thèse en co-tutelle, concernant Mlle. Vasile (thèse qui démarrera en octobre 2005).

Concernant la Roumanie, un autre projet de type Leonardo da Vinci¹⁰ d'échange et collaboration entre enseignants-chercheurs a été approuvé par la Communauté Européenne. Ce projet, mis en place en coopération avec l'Université Polytechnique de Timisoara dans le domaine du diagnostic et de la e-maintenance, a comme objectif la cristallisation d'un pôle de compétences autour des problématiques de la maintenance et plus généralement de la productique. Dans ce contexte, au mois de mai 2005, nous avons accueilli 2 enseignants-chercheurs de l'Université Politehnica de Timisoara pour un échange autour d'applications concrètes de diagnostic par imagerie infrarouge et de e-maintenance.

Les contacts avec la Roumanie se sont concrétisés aussi par une conférence plénière sur invitation personnelle [C2I-plen, 2002] à l'occasion de la Conférence Internationale d'Ingénierie Intégrée organisée par l'Université « Politehnica » de Timisoara. Cet événement marque l'aboutissement de plus de 10 années de collaborations, depuis ma sortie des bancs de cette prestigieuse Université et mon passage à l'ENI de Belfort, jusqu'à mon arrivé à l'Université de Besançon.

⁸ NEMOSYS – Naval e-maintenance oriented system – la e-maintenance appliqué à un système naval de défense de la DCN – Division des Constructions Navales. Projet piloté par l'Institut de Productique de Besançon, regroupant autour de lui un consortium comprenant six universités et deux sociétés industrielles spécialisées dans la e-maintenance

⁹ GAP - Groupe Automatique et Productique de l'UFR Sciences, Université de Franche-Comté

¹⁰ LEONARDO DA VINCI - Leonardo da Vinci RO/2004/93019/EX – Trainers in diagnosis and e-maintenance

En élargissant le spectre des relations internationales, le co-encadrement de la thèse de M. Ghernaut s'inscrit dans une dynamique de coopération entre la France et les pays du Maghreb. Cette coopération est marquée par le projet CORUS¹¹ sur la maîtrise à distance des installations industrielles par télémaintenance et télédiagnostic, projet initié et piloté par le Professeur N. Zerhouni, notre responsable d'équipe.

Pour finir, le projet majeur auquel a participé l'équipe *ingénierie de maintenance et e-maintenance* du LAB sur le plan international est sans aucun doute le projet européen PROTEUS¹² du cadre FP5 Integrated Project du programme ITEA¹³ (ITEA s'inscrit dans les projets EUREKA « stratégiques » ayant pour finalité l'innovation). Labellisé en octobre 2002 pour une durée de 30 mois, ce projet a comme défi technologique majeur, le développement d'une plate-forme de e-maintenance permettant l'intégration des TIC dans le processus de maintenance industrielle.

Le projet PROTEUS regroupe des partenaires industriels et universitaires français, allemands et belges. Cette collaboration à l'échelle européenne fait intervenir des partenaires industriels prestigieux comme CEGELEC (France et Allemagne), SCHNEIDER (France et Allemagne), AKN Eisenbahn AG (Allemagne) ainsi que des partenaires universitaires reconnus, comme les laboratoires français LORIA et LIP6, l'Université Technique de Munich et l'Institut de Recherche IFAK en Allemagne.

1.6 COLLABORATIONS SCIENTIFIQUES

Impliquant fortement les TIC, la thématique de la e-maintenance a exigé un travail en coopération avec des informaticiens de l'équipe Distribution et Parallélisme (DP) du Laboratoire d'Informatique de Franche-Comté (LIFC FRE CNRS 2661), l'équipe SYStèmes d'aide à la DEcision et à la Formation (SysDeF) du Laboratoire d'Informatique de Paris 6 (LIP 6 UMR CNRS 7606) et l'équipe Machine Intelligente Autonome (MAIA) du Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA UMR CNRS 7503). Parmi les sujets traités dans le cadre de ces collaborations, les plus significatifs concernent :

- ✚ Spécification d'une plate-forme de e-maintenance.
- ✚ Etude d'un méta outil d'aide au diagnostic dans un contexte de e-maintenance

Vu la finalité de nos recherches ainsi que l'intérêt que les différents partenaires privés ont manifesté par rapport à nos domaines de compétence, les collaborations scientifiques de notre équipe ont naturellement impliqué des partenaires industriels comme :

- ✚ Cegelec S.A. Belfort : projet ANVAR – conception et prototypage d'un système d'aide au diagnostic en utilisant les techniques neuro-floues ;
- ✚ Avensty Ingénierie, Besançon : conception et prototypage d'un système de surveillance temps réel basé sur les réseaux de neurones dynamiques, accessible à distance par un serveur web ;
- ✚ Division des Constructions Navales, Toulon : contrat de recherche – Nemosys – e-maintenance des systèmes de combat naval.

¹¹ *Projet CORUS : France-Algérie/Maroc/Tunisie, projet intitulé : Maîtrise à distance de la sûreté des installations industrielles par télémaintenance et télédiagnostic, projet n° : 02 212 210 211 006*

¹² *PROTEUS a generic platform for e-maintenance (plate-forme générique de e-maintenance) - <http://www.proteus-iteaproject.com>*

¹³ *Information Technology for European Advancement.*

1.7 DIVERS RECHERCHE

1.7.1 PARTICIPATION A DES JURYS DE THESE

Membre en tant qu'examineur des jurys de thèses suivants

Ryad ZEMOURI

« Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance »

Besançon, le 28 novembre 2003

*Jury : A. Bourjault (président), D. Hammad (rapp.), D. Noyes (rapp.), J-M. Faure, **D. Racoceanu**, N. Zerhouni, R. Labourier (PDG Aveny Ingénierie – industriel invité)*

Eugenia MINCA.

« Surveillance des systèmes de production en utilisant les réseaux de Petri flous. Application à la E-maintenance des systèmes flexibles de production »

Targoviste, Roumanie, 24 septembre 2004

*Jury : I. Cucui (président), V. Manzu (rapp.), C. Sibertin-Blanc (rapp.), A. Dorin, **D. Racoceanu**, N. Zerhouni, J. Cardoso (invité)*

Nicolas PALLUAT

« Méthodologie de surveillance dynamique à l'aide des réseaux neuro-flous temporels »

Besançon, le 12 janvier 2006

*Jury : C. Chu (président), P. Brézillon (rapp.), B. Ould-Bouamama (rapp.), A. Bourjault, J-M. Olive, **D. Racoceanu**, N. Zerhouni, J-B Leger (PDG Predict - industriel invité).*

1.7.2 CONGRES INTERNATIONAUX

Membre du comité d'organisation de Mecathronics'01 - 5th Franco-Japanese Congress & 3rd European-Asian Congress, Besançon, 9-11 octobre, 2001

Membre du comité d'organisation du congrès annuel du club des enseignants et des chercheurs en E.E.A, Besançon, 14-16 mai 2003

Membre du comité scientifique du congrès Pentom 2005 - 2ème édition du colloque international francophone « Performance et Nouvelles Technologies en Maintenance », Marrakech (Maroc), 18 - 20 avril 2005

Président de la session « Fault Diagnosis and Maintenance » – IEEE Conference on Robotics, Automation and Mechatronics (IEEE RAM 2004), Singapour, December 1-3, 2004.

1.7.3 RELECTURE D'ARTICLES

L. Seabra Lopes, "Symbolic Learning and Planning for Faillure Recovery in Assembly", *IEEE Transactions on Robotics and Automation*, article examiné en juillet 2002 et en octobre 2005.

Amine Boufaied, Audine Subias, Michel Combacau, Détection distribuée : reconnaissance floue de chroniques distribuées, *JESA – Journal Européen des Systèmes Automatisés*, article examiné en septembre 2003.

Mathieu Glade, Patrick Lyonnet, Gwenaëlle Grosset, Using neural networks to model direct maintenance cost of helicopter sub-assemblies, JESA – *Journal Européen des Systèmes Automatisés*, article examiné en juillet 2004.

Amadou Coulibaly, Xavier Zwingmann, Daoud Ait-Kadi, Bernard Mutel, "Evaluation de la fiabilité des produits mécaniques au stade de la conception par simulation numérique" soumis pour PENTOM 2005 : Performances et Nouvelles Technologies en Maintenance, article examiné en décembre 2004.

Helali Naoufel, "Apport des bio systèmes dans la mise au point de bases de données neuroniques dynamiques", soumis pour PENTOM 2005 : Performances et Nouvelles Technologies en Maintenance, article examiné en décembre 2004.

Référence GI013¹⁴, Détection des défauts en ligne par l'approche statistique : Application du test G.L.R. à fenêtre glissante, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en mars 2005.

Référence GI026, Co-conception de l'implémentation technique et de l'organisation-projet en phase d'avant-projet : le cas d'un projet NTIC pour la production de services, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en février 2005.

Référence GI086, Mise en place d'un System de Management de la Maintenance Cas de Shell-Butagaz Tunisie, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en février 2005.

Référence GI100, Amélioration de la fiabilité des équipements à travers l'optimisation de la maintenance préventive, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en février 2005.

Référence GI109, Le pronostic, processus clé d'une stratégie de maintenance prévisionnelle pour le Maintien en Conditions Opérationnelles d'un système de production, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en février 2005.

Référence GI114, Déploiement de la Maîtrise Statistique des Procédés (MSP) dans une production mécanique en petites séries aéronautique sur une ligne de production automatisée, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en février 2005.

Référence GI137, Aide à la décision pour la conception et le dimensionnement d'un atelier de maintenance, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en février 2005.

Référence GI158, Modèle de dégradation stochastique composé, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en mars 2005.

Référence GI201, Evaluation de la performance en conception, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en février 2005.

Référence GI212, Estimation des coûts dans des industries à haut degré de diversité : modèles interprétables neuro-flous, 6^{ème} Congrès international du Génie Industriel, 7-10 juin 2005, Besançon, France, article examiné en février 2005.

¹⁴ Auteurs sous couvert d'anonymat

1.8 PUBLICATIONS

1.8.1 REVUES INTERNATIONALES AVEC COMITE DE LECTURE

- [CiI, 2006] N. Palluat, **D. Racocceanu**, N. Zerhouni, A neuro-fuzzy monitoring system. Application to flexible production system, *E-maintenance special issue, Computers in Industries*, Ed. Elsevier, vol. 4, avril, 2006.
- [RIA, 2005] N. Palluat, **D. Racocceanu**, N. Zerhouni, Utilisation des réseaux de neurones temporels pour le pronostic et la surveillance dynamique. Etude comparative de trois réseaux de neurones récurrents, *Revue des sciences et technologies de l'information - RSTI, série Revue d'intelligence artificielle – RIA*, n°6/2005.
- [EEAI, 2003] R. Zemouri, **D. Racocceanu**, N. Zerhouni, Recurrent Radial Basis Function network for Time-Series Prediction, *Engineering Applications of Artificial Intelligence, The International Journal of Intelligent Real-Time Automation, journal IFAC - the International Federation of Automatic Control*, ed. Elsevier Science, vol. 16, Issue 5-6, pp.453-463, 2003.
- [JESA, 2003] R. Zemouri, **D. Racocceanu**, N. Zerhouni, Réseaux de neurones récurrents à fonctions de base radiales. Application à la surveillance dynamique, *JESA – Journal Européen des Systèmes Automatisés*, vol. 37, n°1, pp. 49-81, 2003.
- [RIA, 2002] R. Zemouri, **D. Racocceanu**, N. Zerhouni, Réseaux de neurones récurrents à fonctions de base radiales. Application au pronostic, *Revue des sciences et technologies de l'information - RSTI, série Revue d'Intelligence Artificielle - RIA*, vol. 16, n° 13, ISBN : 2-7462-0568-8, Ed. Hermes – Lavoisier, Paris, pp. 307-338, 2002.
- [SIC, 2002] E. Minca, **D. Racocceanu**, N. Zerhouni, Monitoring system modelling and analysis using fuzzy Petri nets, *Studies in Informatics and Control Journal*, vol. 11, n°4, pp. 331-338, décembre 2002.
- [RISPM, 2003] R. Zemouri, **D. Racocceanu**, N. Zerhouni, S. Durand, Utilisation conjointe des techniques de modélisation analytique et de simulation pour l'évaluation des performances et l'optimisation de l'organisation d'un service de maintenance, *International Journal of Mechanical Production System Engineering*, n° 7, pp. II-3 à II-14, décembre 2003.

Publications en revues internationales de la période doctorale :

- [MMS, 1995] **D. Racocceanu**, A. El Moudni, S. Zerhouni, M. Ferney, ON A NEW METHOD OF MARKOV CHAIN REDUCTION, *revue Journal of Mathematical Modelling of Systems*, vol 1, n° 3, octobre 1995.
- [SAMS, 1994] **D. Racocceanu**, A. El Moudni, S. Zerhouni, M. Ferney, A SINGULAR PERTURBATION APPROACH TO MODELLING AND RESOLUTION OF MARKOV CHAINS, *revue SAMS – System Analysis Modelling Simulation*, vol. 15, 1994.

1.8.2 CONFERENCES SUR INVITATION PERSONNELLE

- [Tsingua Univ., 2005]

D. Racocceanu, N. Zerhouni, DYNAMIC MONITORING SYSTEMS USING RECURRENT NEURAL NETWORKS AND NEURO-FUZZY SYSTEMS. NEMOSYS PROJECT, Présentation devant les membres du département Intelligence Artificielle de l'Université Tsingua de Pékin, Chine, 17 octobre 2005.

[NUS, 2005] **D. Racocceanu**, MONITORING APPROACH USING RECURRENT RADIAL BASIS FUNCTION NEURAL NETWORKS AND NEURO-FUZZY SYSTEMS, Présentation des travaux devant les membres du département recherche d'information de la Faculté d'Informatique de l'Université Nationale de Singapour, 7 décembre 2005.

[C2I-plen, 2002] **D. Racocceanu**, RESEAUX DE NEURONES DYNAMIQUES. APPLICATIONS A LA MAINTENANCE PREVENTIVE ET A LA TELEMANTENANCE, *Conférence plénière, Conférence Internationale d'Ingénierie Intégrée - C2I 2002*, Timisoara, Roumanie, 25-26 avril 2002.

1.8.3 COMMUNICATIONS A DES CONGRES INTERNATIONAUX AVEC COMITE DE LECTURE ET PUBLICATION DES ACTES

[CIRAS, 2005] **D. Racocceanu**, N. Palluat, DYNAMIC NEURO-FUZZY SYSTEM FOR DIAGNOSIS AID, The Third International Conference on Computational Intelligence, Robotics and Autonomous Systems - CIRAS 2005, Singapour, 13-16 décembre 2005.

[IEEE ICNN&B, 2005]

D. Racocceanu, N. Zerhouni, Monitoring Approach Using Recurrent Radial Basis Function Neural Networks and Neuro-Fuzzy Systems, 2005 International Conference on Neural Networks and Brain - IEEE ICNN&B'05, Pékin, Chine, 13-15 octobre 2005.

[IFAC, 2005] N. Palluat, **D. Racocceanu**, N. Zerhouni, An UML modeling of a neuro-fuzzy monitoring system, *16th IFAC World Congress* Prague, République Tchèque, 4-8 juillet 2005.

[IFAC-MCPL, 2004] M. Monnin, N. Palluat, **D. Racocceanu**, N. Zerhouni, Diagnosis Methods Using Artificial Intelligence. Application of Fuzzy Petri Nets and Neuro-Fuzzy systems, *IFAC MCPL 2004 – Third Conference on Management and Control of Production and Logistics*, Santiago, Chili, 3-5 novembre 2004.

[IEEE-RAM, 2004] M. Monnin, **D. Racocceanu**, N. Zerhouni, Overview on diagnosis methods using artificial intelligence, *IEEE Conference on Robotics, Automation and Mechatronics - IEEE RAM 2004*, Singapour, 1-3 décembre, 2004.

[IEEE-MechRob, 2004] PROTEUS ITEA European Project WP2 TEAM (L. Déchamp, A. Dutech, T. Montroig, X. Qian, **D. Racocceanu**, I. Rasovska, P. Brézillon, F. Charpillat, J-Y. Jaffray, N. Moine, B. Morello, S. Müller, G. Nguengang, N. Palluat, L. Pelissier), On the Use of Artificial Intelligence for Prognosis and Diagnosis in the PROTEUS E-maintenance platform, *IEEE Mechatronics & Robotics 2004*, Aachen, Germany, E-Maintenance Special Session, September 13-15, 2004.

- [DX, 2004] R. Zemouri, **D. Racocceanu**, N. Zerhouni, Coupling the dynamic memory with the static memory of a neural network for the diagnosis by pattern recognition, *DX-2004, 15th International Workshop on Principles of Diagnosis*, Carcassonne, France, 23-25 juin 2004.
- [IMS, 2004] N. Palluat, **D. Racocceanu**, N. Zerhouni, Diagnosis using neuro-fuzzy systems, *Intelligent Maintenance System - IMS 2004*, 14-15 juillet 2004, Arles, France.
- [IEEE-ICRA, 2003] E. Minca, **D. Racocceanu**, N. Zerhouni, Fuzzy logic Petri nets for monitoring systems modeling, *IEEE International Conference on Robotics & Automation, ICRA'2003*, 12-17 mai 2003, Taipei, Taiwan.
- [IFAC, 2002] R. Zemouri, **D. Racocceanu**, N. Zerhouni, Application of the dynamic RBF network in a monitoring problem of the production systems, *IFAC World Congress 2002*, Barcelone, Espagne, 21-26 juillet 2002.
- [SCI, 2002] E. Minca, **D. Racocceanu**, N. Zerhouni, I. Brezeanu, Fuzzy based Petri nets for the production systems diagnosis, *6th World Multiconference on Systemics, Cybernetics and Informatics SCI 2002*, Orlando, USA, 14-18 juillet 2002.
- [IEEE-WCCI, 2002] R. Zemouri, **D. Racocceanu**, N. Zerhouni, From the spherical to an elliptic form of the dynamic RBF neural network influence field, *2002 IEEE World Congress on Computational Intelligence, International Joint Conference on Neural Networks, IEEE-WCCI/IJCNN2002*, USA, 12-17 mai 2002.
- [IEEE-ICRA, 2002] **D. Racocceanu**, N. Zerhouni, N. Addouche, Modular Modeling and Analysis of a Distributed Production System with Distant Specialised Maintenance, *IEEE ICRA 2002*, Washington, USA, 11-15 mai 2002.
- [IEEE-ETFA, 2001] R. Zemouri, **D. Racocceanu**, N. Zerhouni, The RRBF Dynamic representation of time in Radial Basis Function Network, *8th IEEE International Conference on Emerging Technologies and Factory Automation – IEEE ETFA'2001*, vol. 2, pp. 737-740, Antibes Juan-Les-Pins, France, 15-18 octobre 2001.
- [IEEE-ICRA, 2001] R. Zemouri, **D. Racocceanu**, N. Zerhouni, A Petri nets graphic method of reduction using birth-death processes, *IEEE International Conference on Robotics and Automation, ICRA'2001*, pp 46-51, Séoul, Corée, 21-26 mai 2001 (Best paper Award finalist)
- [IFAC-MCPL 2000] **D. Racocceanu**, N. Zerhouni, Use of Singular Perturbations for the Reduction of Manufacturing System Models, *Congrès IFAC - MCPL-2000, 2nd IFAC Conference on Management and Control of Production and Logistics*, Grenoble, France, 5-8 juillet 2000.

Publications en conférences internationales durant la période doctorale :

- [IFAC, 1997] **D. Racocceanu**, A. El Moudni, S. Zerhouni, M. Ferney, A DISCRETE SINGULAR PERTURBATION APPROACH TO MODELING AND CONTROL OF BILINEAR MARKOV DECISION PROCESSES, congrès IFAC "Singular solutions and perturbations in control system" – SSPCS-97, Pereslavl – Zalessky, Russie, 7-11 juillet 1997.
- [IMACS, 1995] **D. Racocceanu**, A. El Moudni, M. Ferney, S. Zerhouni, A NEAR OPTIMAL CONTROL OF MARKOV CHAINS USING DISCRETE SINGULAR

PERTURBATIONS IN THE CASE OF FIRST ORDER POLYNOMIAL CONTROL. TERMINAL PROBLEMS, congrès IMACS-SAS'95, Berlin, Allemagne, 27-29 juin 1995.

[IEEE ICRA, 1994] **D. Racocceanu**, A. El Moudni, S. Zerhouni, M. Ferney, *USE OF AN HOMOGRAPHIC TRANSFORMATION JOINTLY TO THE SINGULAR PERTURBATION FOR THE STUDY OF MARKOV CHAINS. APPLICATION TO THE OPERATIONAL SAFETY STUDY, IEEE International Conference on Robotics and Automation, San Diego, USA, 8-13 mai 1994.*

[MATHMOD, 1994] **D. Racocceanu**, A. El Moudni, S. Zerhouni, M. Ferney, *A MARKOV CHAIN SINGULAR PERTURBATION RESOLUTION, congrès 1-Mathmod Vienna, Autriche, 2-4 février 1994.*

1.8.4 COMMUNICATIONS A DES CONGRES NATIONAUX OU A AUDIENCE FRANCOPHONE AVEC COMITE DE LECTURE ET PUBLICATION DES ACTES

[GI, 2005] A. Ghernaut, **D. Racocceanu**, N. Zerhouni, Approche abductive utilisant les réseaux de Petri flous pour le diagnostic et la qualité des systèmes mécatroniques, *6^{ème} Congrès international du Génie Industriel*, 7-10 juin 2005, Besançon, France.

[PENTOM, 2005] A. Ghernaut, **D. Racocceanu**, N. Zerhouni, Approche duale maintenance – qualité pour la surveillance des systèmes mécatroniques, *PENTOM 2005 : Performances et Nouvelles Technologies en Maintenance*, 18-21 Avril 2005, Marrakech, Maroc.

[CIFA, 2004] R. Zemouri, **D. Racocceanu**, N. Zerhouni, Réseau RRFR pour la surveillance dynamique : application en e-maintenance, *CIFA-2004, Conférence Internationale Francophone d'Automatique*, Douz, Tunisie, 22-24 novembre 2004

[IAI, 2003a] Djouak Amar, **Racocceanu Daniel**, Zerhouni Nouredine, Etude d'un multicapteur intelligent à base de réseaux de neurones, *4^{ème} Conférence Internationale sur l'Automatisation Industrielle*, 9-11 juin 2003, Montréal, Canada.

[IAI, 2003b] **Racocceanu Daniel**, Djaffar Ould Abdeslam, Réseaux Neuro-Flous pour la Surveillance des Systèmes, *4^{ème} Conférence Internationale sur l'Automatisation Industrielle*, 9-11 juin 2003, Montréal, Canada.

[PENTOM, 2003] E. Minca, **D. Racocceanu**, N. Zerhouni, Approche de supervision par réseaux de Petri flous, *PENTOM – Performance et Nouvelles Technologies en Maintenance*, Valenciennes, 19-21 mars 2003.

[QUALITA, 2003] E. Minca, **D. Racocceanu**, N. Zerhouni, Approche temporelle floue pour la surveillance, *QUALITA 2003*, Nancy, 11-13 mars 2003.

[ENSET, 2002] R. Zemouri, N. Zerhouni, **D. Racocceanu**, Application des réseaux RBF Récurrents (RRBF) à un problème de surveillance, *International Conference on Mechanical Engineering, SIGMA'02*, pp. 355-360, 28-29 avril 2002, ENSET, Oran, Algérie.

[C2I, 2002] J. Khaabi, **D. Racocceanu**, N. Zerhouni, Elaboration d'une politique de maintenance préventive pour la gestion d'un parc de pinces à souder robotisées, *Conférence Internationale d'Ingénierie Intégrée - C2I 2002*, Timisoara, Roumanie, 25-26 avril 2002.

- [GI, 2001] R. Zemouri, **D. Racocceanu**, N. Zerhouni, S. Durand, Simulation et évaluation des performances d'un atelier de maintenance, actes du 4^{ème} *Congrès International du Génie Industriel*, vol II, pp. 649-661, Marseille, France, 12-15 juin 2001.
- [SIMI, 2001] R. Zemouri, **D. Racocceanu** et N. Zerhouni, Evaluation des performances d'un atelier de maintenance de Peugeot Citroën Automobiles, site de Sochaux – France, *Symposium International sur la Maintenance Industrielle*, SIMI 2001, Alger, Algérie, 21-23 janvier 2001.

1.8.5 COMMUNICATIONS DANS LE CADRE DE GROUPES DE TRAVAIL ET COLLOQUES NATIONAUX

- [IP, 2005] **D. Racocceanu**, Diagnostic et pronostic à l'aide des techniques de l'intelligence artificielle, communication dans le cadre de la journée technique organisée par l'*Institut de Productique de Besançon*, intitulée : « Les atouts de la e-maintenance pour la performance industrielle », Cegelec S.A., Belfort, 19 mai 2005.
- [MACS, 2005] A. Ghernaut, **D. Racocceanu**, N. Zerhouni, Surveillance des systèmes mécatroniques par une approche maintenance – qualité, réunion commune aux GTs MACOD (maintenance coopérative distribuée) et AMOEP (évaluation de performances), *journées du GdR MACS*, Clermont-Ferrand, France, 30 mars -1^{er} avril 2005.
- [MACS, 2003] R. Zemouri, **D. Racocceanu**, N. Zerhouni, Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance, *journées du GdR MACS*, Bordeaux, France, 16-17 octobre 2003.
- [NSI, 2002] R. Zemouri, **D. Racocceanu**, N. Zerhouni, Prédiction des séries temporelles par le réseau Recurrent Radial Basis Function, *XI^{èmes} journées neurosciences et sciences de l'ingénieur*, NSI'2002, L'Agelonde, La Londe Les Maures (Var), France, 15-18 septembre 2002.
- [GRP, 2002] R. Zemouri, **D. Racocceanu**, N. Zerhouni, Application des réseaux de neurones récurrents (RRBF) à la détection dynamique de dégradations, *journées du GRP – Groupement de Recherche en Productique*, groupe de travail ASSF – Automatisation et Systèmes Sûrs de Fonctionnement, Grenoble, 21-22 mars 2002.

1.8.6 RAPPORTS DE CONTRATS DE RECHERCHE

Rapports de recherche et rapports de formation développés dans le cadre du projet **NEMOSYS** (*Naval E-Maintenance Oriented SYStem*) pour le compte de la **DCN** – Division des Constructions Navales :

- [DCN, 2004] **D. Racocceanu**, *Techniques d'Intelligence artificielle pour la surveillance industrielle* (Intervention de 3 heures), rapport de formation du personnel de la société DCN (Division des Constructions Navales) dans le cadre du contrat DCN – LAB : séance 3/3 du 15-16 décembre 2004. Session : Les Techniques et Technologies de la e-Maintenance, section : Techniques pour les processus « Métier » de la e-Maintenance.

- [DCN, 2005] **D. Racoceanu**, N. Palluat, *Système de surveillance dynamique embarquée et d'aide distante au diagnostic*, rapport pour le compte de la Division des Constructions Navales - DCN, référence LAB-DCN-DR01, 26 janvier 2005.

Rapports de recherche **ANVAR** :

- [ANVAR, 2004] N. Palluat, **D. Racoceanu**, Conception et prototypage d'un système de surveillance dynamique embarqué et d'aide au diagnostic d'un centre d'usinage à grande vitesse, *Rapport ANVAR* – aide aux jeunes pour l'innovation, Besançon, décembre 2004.

Rapports de recherche élaborés durant la période doctorale dans le cadre d'une collaboration avec EDF – exploitation hydro énergétique Jura Morvan :

- [EDF, 1996] **D. Racoceanu**, J. Mesnier (responsable d'exploitation EDF), *Commande quasi-optimale pour la gestion du système hydro-énergétique Jura Morvan*, rapport pour le compte d'EDF Energie Est, avril 1996.
- [EDF, 1995] **D. Racoceanu**, J. Mesnier (responsable d'exploitation EDF), *Processus de diffusion stochastique appliqué à la gestion des ressources d'eau des barrages du Chatelot, Refrain et Vaufrey du système hydro-énergétique Jura-Morvan*, rapport pour le compte d'EDF Energie Est, septembre 1995.

1.8.7 RAPPORTS DE RECHERCHE PROJET EUROPEEN

Rapports de recherche développés dans le cadre du projet européen ITEA (Information Technology for European Advancement) – **PROTEUS** (plate-forme de e-maintenance) :

- [PROTEUS-D2.1, 2003] Delivable 2.1. – 4 juillet 2003 - PROTEUS-D-211-LORIA-030704 – *State of the art on AI and decision-making* (134 pages)
- Chapitre : [NN1] Use of the Recurrent Radial Basis Function (RRBF) network for dynamic monitoring and preventive maintenance, pp.77-80
- Chapitre : [NN2] Use of the Radial Basis Function (RBF) Neural network for production system monitoring and curative maintenance, pp. 81-84
- Chapitre : [FL2] Fuzzy logic and fuzzy Petri nets for diagnosis, pp.108-111
- [PROTEUS-D2.2, 2003] Delivable 2.2.- 8 décembre 2003 - PROTEUS-D-221- IML - 08122003 – *Specification of Knowledge based Methods for Corrective and Predictive Maintenance* (114 pages)
- Chapitre : [M6] Radial Basis Function Neural Networks and Recurrent Radial Basis Function, pp. 83-91
- Chapitre : [M7] Fuzzy logic and fuzzy Petri nets – pp. 92-96
- [PROTEUS-D2.3, 2004] Delivable 2.3.- 28 juillet 2004 - PROTEUS-D-23-all 28072004(3) – *Specification of the decision tool* (214 pages)
- Neuro-Fuzzy Diagnosis and Prognosis tool (General concepts, UML modelling, Summary of Neuro-Fuzzy methods)

1.8.8 COMMUNICATIONS DIVERSES

a) DIFFUSION DES CONNAISSANCES :

- [EDP, 2005] N. Zerhouni, B. Morello, C. Varnier, **D. Racocceanu**, LES ACTIVITES DE E-MAINTENANCE A L'INSTITUT DE PRODUCTIQUE – « Mieux vaut prévenir que guérir » à l'aide des technologies de l'internet et des techniques d'intelligence artificielle, revue « *En Direct Productique* », Université de Franche-Comté, Institut de Productique, n°192, février 2005
- [EDP, 2004] N. Zerhouni, **D. Racocceanu**, R. Labourier, UN SYSTÈME DE E-MAINTENANCE FONDÉ SUR LES RÉSEAUX DE NEURONES DYNAMIQUES - Premier né de la collaboration entre le laboratoire d'Automatique de Besançon et la société AVENSY INGÉNIERIE, revue « *En Direct Productique* », Université de Franche-Comté, Institut de Productique, n°187, 1^{ère} page, septembre 2004

b) RAPPORTS INTERNES :

- [IUT, 1995] **D. Racocceanu**, Programme de gestion du système flexible industriel d'assemblage de l'IUT de Belfort, rapport pour le compte de l'IUT de Belfort, avril 1995.
- [IUT, 1994] **D. Racocceanu**, Etude des paramètres de fonctionnement du système flexible de l'IUT de Belfort sur le logiciel Siman/Arena, rapport pour le compte de l'IUT de Belfort, octobre 1994.

1.9 RESUME DES ACTIVITES D'ENSEIGNEMENT

NB : Les activités d'enseignement sont détaillées dans le chapitre 2

Depuis mon arrivée à l'UFR des Sciences et Techniques de Besançon en septembre 1999, j'effectue mon enseignement au sein du Groupe Automatique et Productique (GAP) de l'UFR Sciences et Techniques de Besançon. Ce service d'enseignement fédère les enseignements dans les domaines de l'automatique, la productique, la robotique, la vision et l'informatique industrielle.

Ainsi, mon service d'enseignement se répartit sur le deuxième et troisième cycle universitaire, en formation initiale, mais également en formation continue (formation pour la DCN – Division des Constructions Navales, année 2004/2005), essentiellement en productique, informatique industrielle, vision, automatique et intelligence artificielle, aussi bien en cours, qu'en TD et en TP, sans oublier bien évidemment l'encadrement des projets industriels.

Les disciplines enseignées et les filières concernées sont résumées ci-après :

-  Productique (systèmes à événements discrets, simulation de flux, implantation, maintenance, fiabilité, diagnostic) :
 - o Maîtrise EEA, option Automatique et Productique
 - o IUP SAPIAA (Systèmes Automatisés de Production pour les Industries Agro-Alimentaires), 3^{ème} année
 - o Licence professionnelle GPI (Gestion de la Production Industrielle) option logistique industrielle, 3^{ème} année

- o Licence professionnelle GPI (Gestion de la Production Industrielle) option gestion de production intégrée, 3^{ème} année
- o Master Recherche MT (Mécatronique), 2^{ème} année
- + Informatique Industrielle :
 - o IUP GMI (Génie Mathématique et Informatique), 3^{ème} année
 - o Maîtrise EEA, tronc commun
 - o Master Professionnel MT (Mécatronique), 1^{ère} année
 - o Master Professionnel ELO (Electronique et Optique), 1^{ère} année
- + Vision :
 - o DESS PA (Productique Appliquée)
 - o DESS SMTII (Systèmes de Mesure, de Test et d'Imagerie Industriels)
 - o Master Professionnel MT (Mécatronique), 2^{ème} année
 - o Master Professionnel ELO (Electronique et Optique), 2^{ème} année
 - o IUP SAPIAA (Systèmes Automatisés de Production pour les Industries Agro-Alimentaires), 3^{ème} année
- + Automatique :
 - o Maîtrise EEA, option Automatique et Productique
- + Intelligence artificielle :
 - o Ecole doctorale Sciences Physiques pour l'Ingénieur et Microtechniques
- + Encadrement de projets en productique et vision :
 - o DESS PA
 - o Maîtrise EEA
 - o Licence EEA
- + Tutorat pédagogique - apprentis formation par alternance :
 - o IUP SAPIAA

Mise à part mon expérience professionnelle industrielle, mes activités pédagogiques actuelles me permettent d'être en contact avec le secteur industriel par le biais de l'encadrement de projets et de stages (DESS, Maîtrise), et d'étudiants en contrat d'apprentissage (IUP SAPIAA). Vu mon parcours, ceci a été pour moi un souci permanent.

1.10 EXPERIENCE EN ENTREPRISE

Avant mon arrivé en tant que maître de conférences, j'ai été employé dans le secteur privé, dans deux structures différentes (une PME et une grande entreprise) qui m'ont permis – chacune d'entre elles – de découvrir un mode d'organisation différent d'une entreprise et de développer mes compétences essentiellement organisationnelles.

1997-1998 Responsable logistique - ordonnancement, Gaussin S.A., Héricourt, concepteur et fabricant d'automoteurs industriels, systèmes spéciaux de manutention pour l'industrie, les ports et aéroports. CA : 10 M€, 78 personnes

- Responsable transports internes / internationaux
 - Responsable des flux d'entrée, sortie et internes de l'entreprise
 - Responsable planning – ordonnancement
 - Pilote du groupe Qualité Fournisseurs dans le cadre de la démarche de certification ISO 9002
 - Pilote GPAO – Production (mise en place du système et suivi)
 - Contact direct avec la clientèle et les fournisseurs
 - Supervision BM, Achats, Magasin, Expéditions et rôle de liaison entre tous les services de l'entreprise.
 - Encadrement de 5 personnes
- 1998-1999 Responsable d'affaires, Alstom Gaz Turbines / General Electric France, Belfort, Concepteur et fabricant de turbines à gaz, CA : 1,2 Md€, 1800 personnes.
- Responsabilité financière de 5 affaires de turbines gaz
 - Coordination des services concernés par les projets (réunions de lancement, réunions techniques, suivi, bilan, ...)
 - Veiller au respect du planning, du budget et des particularités techniques de chaque affaire
 - Contact avec le client
 - Rapport périodique (réunions d'affaires) auprès de la hiérarchie interne (chef de service)

1.11 RESUME DES RESPONSABILITES SCIENTIFIQUES, COLLECTIVES ET ADMINISTRATIVES

- + Membre élu du conseil du laboratoire depuis le 1^{er} janvier 2000
- + Responsable projets Schneider et Avensy-Siemens (prototype breveté) de l'équipe « ingénierie de maintenance et e-maintenance »
- + Coresponsable de la plate-forme technologique e-maintenance de l'équipe
- + Membre de la commission de spécialistes de 61^{ème} section Université de Franche-Comté entre 2000 et 2004

1.12 FORMATIONS SUIVIES

Depuis mon arrivée au LAB, j'ai suivi les formations suivantes :

- + Formation CNRS anglais au CLA (2000)
- + Formation au langage Java (2003)
- + Stage de programmation Orientée Objet. Modélisation UML (2003)

Chapitre 2 - Activités d'Enseignement

2.1 PREAMBULE

De même que mes activités de recherche, mes activités d'enseignement peuvent se décomposer en deux périodes, une première période belfortaine (septembre 1993 – août 1996) à l'IUT de Belfort – Montbéliard de Université de Franche-Comté et une seconde période bisontine (depuis septembre 1999) à l'UFR Sciences et Techniques de l'Université de Franche-Comté.

2.2 RESUME DES ACTIVITES D'ENSEIGNEMENT A BELFORT

Ces enseignements ont été effectués au sein de l'IUT de Belfort Montbéliard en tant que vacataire de septembre 1993 à août 1994, puis en tant qu'ATER de septembre 1994 à août 1996. Dans ce cadre, il est à noter que j'ai été amené à créer une série de TP de réseaux locaux industriels et informatique industrielle sur un système flexible de production de l'IUT de Belfort, ainsi qu'un module complet cours/TD/TP en modélisation et simulation de flux. Ces enseignements ont été créés, ne constituant pas la reprise d'un enseignement précédent.

Par ailleurs, j'ai assuré des TP de GPAO et un cours/TD d'Implantation d'atelier.

2.3 BILAN DES ACTIVITES D'ENSEIGNEMENT A BESANÇON

Le service d'enseignement effectué dans le cadre de mes fonctions en tant que maître de conférences (au sein du GAP, Groupe Automatique et Productique, service d'enseignement de l'UFR Sciences et Techniques), se décompose en deux parties : architecture classique - jusqu'à août 2003 - et nouvelle architecture de diplômes, induite par le passage au système LMD (Licence, Master, Doctorat) - depuis septembre 2004.

2.3.1 STRUCTURE DE L'ENSEIGNEMENT AVANT LA MISE EN PLACE DU LMD

En résumé, les différentes disciplines enseignées entre septembre 1999 et août 2004, avec leurs poids respectifs globaux sont les suivantes (indiquées en heures étudiant) :

-  Productique (Systèmes à Evénements Discrets, Implantation, Simulation de flux, Maintenance, Gestion de production) (28%)
-  Vision industrielle (20%)
-  Informatique industrielle (18%)
-  Automatique non linéaire (6%)
-  Intelligence Artificielle (3%)
-  Encadrement de projets en EEA et productique (25%)

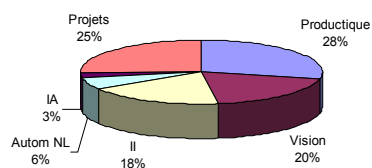


Figure 1. Poids des disciplines enseignées entre 1999 et 2004 en heures étudiant

Par ailleurs, une représentation des répartitions par formation et type d'activité est représentée (en heures réelles) par les schémas suivants :

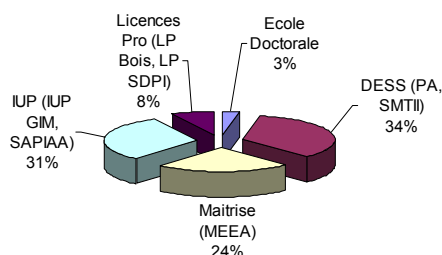


Figure 2. Répartition moyenne des heures réelles par type de formation entre sept 1999 et sept 2004

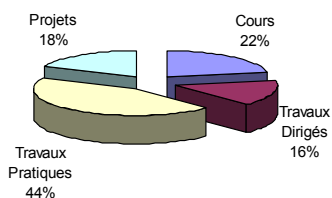


Figure 3. Répartition moyenne des heures réelles par type d'activité entre sept 1999 et août 2004

Les paragraphes suivants donnent, par cycle universitaire, le détail des enseignements dispensés. Notons que tous les TP, TD et Cours que j'ai créés sont mentionnés dans le détail des enseignements.

NB : Lorsqu'un enseignement est dispensé depuis plusieurs années et que son volume horaire a peu évolué, le nombre d'heures et d'étudiants indiqués ici correspondent à des valeurs moyennes.

a) DEUXIEME CYCLE UNIVERSITAIRE

1999-2004 Informatique industrielle et systèmes temps réel, IUP Génie Informatique, 3ème année option Informatique Industrielle, 18h C, 18h TD et 18h TP par an (création d'un TP temps réel, 16 étudiants entre 1999 et 2003 et 32 étudiants la dernière année 2003/2004).

- 1999-2004 Systèmes temps réel, Maîtrise EEA, tronc commun, 7h C, 3h TD, 8h TP par an (création d'un TP temps réel, 32 étudiants).
- 1999-2004 Systèmes à Evènements discrets, Maîtrise EEA, option Automatique et Productique, 16h C, 10h TD par an (création de l'ensemble du cours et des TD, 10 étudiants).
- 1999-2004 Automatique non-linéaire, Maîtrise EEA, option Automatique et Productique, 16h C, 9h TD par an (création de l'ensemble du cours et des TD (10 étudiants).
- 2000-2001 Vision Industrielle, IUP SAPIAA (Systèmes Automatisés de Production pour les Industries Agro-Alimentaires) 3ème année, 12h TP (20 étudiants).
- 2001-2004 Vision Industrielle, IUP SAPIAA 3ème année, 6h C, 12h TP par an (20 étudiants).
- 2000-2004 Simulation de systèmes de production, IUP SAPIAA 3ème année, 6h C, 6h TP par an (création du cours et du TP de simulation de flux de production spécifique à l'industrie agro-alimentaire, 20 étudiants).
- 1999-2001 Gestion de production, IUP SAPIAA 3^{ème} année, 8h TP par an (création d'un TP de gestion d'entreprises concurrentes sur un même marché).
- 2003-2004 Maintenance Industrielle, Licence professionnelle « Bois et ameublement - Techniques et management dans les industries du bois », 8h C, 4h TD, 8h TP (cours et TD créé, 16 étudiants).
- 2003-2004 Implantation et simulation de flux, Licence Professionnelle Bois et ameublement, 8h C, 4h TD, 8h TP (cours et TD créés, 16 étudiants).
- 2000-2002 Simulation de flux, Licence professionnelle SDPI (Science de la Production Industrielle) option logistique industrielle, 10h TD, 16h TP par an (20 étudiants).

b) TROISIEME CYCLE UNIVERSITAIRE

- 1999-2004 Vision industrielle, DESS PA (Productique Appliquée), 24h TP par an (20 étudiants).
- 1999-2004 Vision industrielle, DESS SMTII (Systèmes de Mesure, de Test et d'Imagerie Industriels), 12h TP par an (10 étudiants).
- 2002-2004 Réseaux de neurones, Ecole Doctorale SPIM (École Doctorale Sciences Physiques pour l'Ingénieur et Microtechniques) de l'Université de Franche-Comté, 10h C par an (cours entièrement créé basé sur nos derniers travaux de recherche, moyenne de 8 étudiants).

2.3.2 STRUCTURE DE L'ENSEIGNEMENT APRES LA MISE EN PLACE DE L'ORGANISATION LMD

Le passage à l'organisation LMD et l'évolution de la carte des formations de l'UFR Sciences, a induit – en ce qui me concerne – un recentrage autour de mes préoccupations de recherche et plus généralement autour de la productique par sa composante maintenance industrielle. Des modules d'enseignements de type automatique non-linéaire ont été abandonnés au profit d'interventions en maintenance (fiabilité, diagnostic) ; intelligence artificielle appliquée à la surveillance, productique (simulation de flux, implantation, gestion de la production) et informatique industrielle. Par ailleurs, un volume important de

cours récemment créés, constitue une grande partie de mon intervention au niveau Master Mécatronique 2^{ème} année, orientation recherche. Ces cours (Fiabilité, Diagnostic industriel, Intelligence artificielle) s'inspirent fortement de nos derniers résultats de recherche, ainsi que d'un état de l'art au goût du jour concernant les nouvelles tendances dans ces domaines d'activité.

Dans la nouvelle maquette LMD, j'ai conçu et proposé les programmes pédagogiques suivants :

- ✚ Informatique Industrielle :
 - o Informatique Industrielle et Systèmes temps réel
 - ✚ Master Mécatronique 1^{ère} année – UFR Sciences
 - ✚ Master Electronique et Optique 1^{ère} année – UFR Sciences
- ✚ Intelligence Artificielle :
 - o Cours de réseaux de neurones – module automatique avancée
 - ✚ Master Mécatronique Recherche 2^{ème} année – UFR Sciences – ENSMM
 - ✚ Ecole Doctorale SPIM – UFR Sciences
- ✚ Maintenance, Sûreté de Fonctionnement, Surveillance – Diagnostic
 - o Cours de Sûreté de Fonctionnement et Maintenance
 - ✚ Master Mécatronique 1^{ère} année – UFR Sciences
 - ✚ Licence Professionnelle Bois – UFR Sciences
 - o Cours Diagnostic – Surveillance
 - ✚ Master Mécatronique Recherche 2^{ème} année – UFR Sciences – ENSMM
 - o Cours de Fiabilité – module E-factory
 - ✚ Master Mécatronique Recherche 2^{ème} année – UFR Sciences – ENSMM
- ✚ Ingénierie des systèmes de production
 - o Cours d'Implantation d'atelier
 - ✚ Licence SDPI – IUT Besançon
 - o Cours de Simulation de flux
 - ✚ Licence Professionnelle Bois – UFR Sciences

Les différentes disciplines enseignées depuis septembre 2004, avec leurs poids respectifs globaux, sont les suivantes (comptabilisées en heures étudiant) :

- ✚ Productique (Maintenance, Fiabilité, Diagnostic, Simulation de flux, Gestion de production) (38%) - augmentation de 10 %
- ✚ Vision industrielle (22%) – augmentation de 2%
- ✚ Informatique industrielle (23%) – augmentation de 5%
- ✚ Intelligence Artificielle (5%) – augmentation de 2%
- ✚ Encadrement de projets en productique (12%)

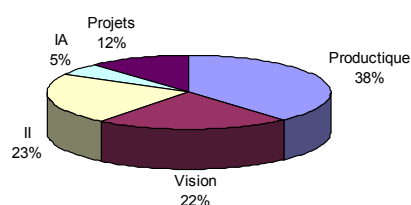


Figure 4. Poids des disciplines enseignées depuis 2004 en heures étudiant

Par ailleurs, une représentation des répartitions par formation et type d'activité est représentée (en heures réelles) par les schémas suivants :

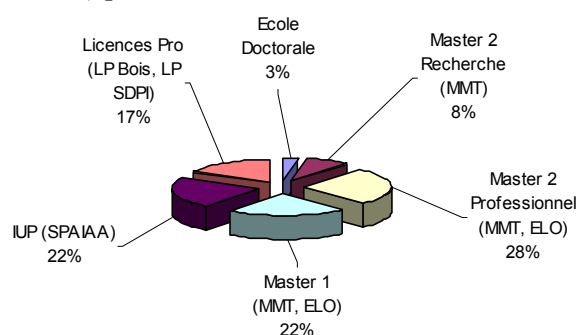


Figure 5. Répartition moyenne des heures réelles par type de formation après 2004 (LMD)

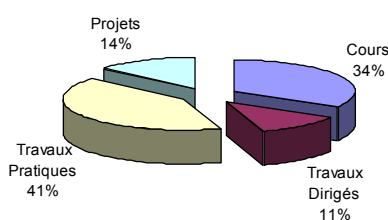


Figure 6. Répartition des heures réelles par type d'activité à partir de sept 2004 (LMD)

Les paragraphes suivants donnent, par niveau de formation (Licence, Master, Doctorat), le détail des enseignements dispensés depuis septembre 2004. Notons que cette fois-ci, quasiment tous les enseignements de type cours et TD ont été créés (mention dans le détail des enseignements).

NB : Lorsqu'un enseignement est dispensé depuis plusieurs années et que son volume horaire a peu évolué, le nombre d'heures et d'étudiants indiqués ici correspondent à des valeurs moyennes. Par ailleurs, les enseignements qui continuent sous le même format (à caractère provisoire – formations en négociations) qu'avant le LMD sont marqués en *italique* par un astérisque (*).

a) NIVEAU L – LICENCE

- 2004-2005 Implantation des systèmes de production, Licence SDPI (Science de la Production Industrielle) option gestion de production intégrée, 8h C, 4h TD, 8h TP (cours et TD créés, 16 étudiants).
- 2004-2005* Maintenance Industrielle, Licence professionnelle « Bois et ameublement - Techniques et management dans les industries du bois », 8h C, 4h TD, 8h TP (cours et TD créés en 2003, 16 étudiants).
- 2004-2005* Implantation et Simulation de flux, Licence Professionnelle Bois et ameublement, 8h C, 4h TD, 8h TP (cours et TD créés en 2003, 16 étudiants).

b) NIVEAU M – MASTER

- 2004-2005* Vision Industrielle, IUP SAPIAA (Systèmes Automatisés de Production pour les Industries Agro-Alimentaires) 3ème année, 12h TP (20 étudiants).
- 2004-2005* Vision Industrielle, IUP SAPIAA 3ème année, 6h C, 12h TP par an (20 étudiants).
- 2004-2005* Simulation de systèmes de production, IUP SAPIAA 3ème année, 6h C, 6h TP par an (cours et TP créés en 2000, 20 étudiants).
- 2004-2005* Gestion de production, IUP SAPIAA 3^{ème} année, 8h TP.
- 2004-2005 Fiabilité, module E-Factory, Master MT (Mécatronique) 2^{ème} année Recherche, parcours CESP (Commande et Exploitation des systèmes de production), 12 h C (cours créé, 10 étudiants)
- 2004-2005 Diagnostic, module Diagnostic-Surveillance, Master MT 2^{ème} année Recherche, parcours CESP, 14 h C (cours créé, 10 étudiants)
- 2004-2005 Réseaux de neurones, module automatique avancée, Master MT 2^{ème} année Recherche, parcours Micromécatronique - Microrobotique, 14 h C (cours créé, 10 étudiants)
- 2004-2005 Vision industrielle, Master MT 2^{ème}année Professionnel, 28h TP (20 étudiants).
- 2004-2005 Modélisation et simulation de flux, Master MT 2^{ème}année Professionnel, 12h TP (TP créés, 20 étudiants).
- 2004-2005 Temps réel, Master MT 1^{ère} année, 12hC, 4 h TD, 24 h TP (cours, TD, TP créé, 10 étudiants).
- 2004-2005 Vision industrielle, Master ELO 2^{ème} année, 12h TP (6 étudiants).
- 2004-2005 Systèmes temps réel, Master ELO (Electronique et Optique) 1^{ère} année, 12h C, 6h TD, 20h TP par an (20 étudiants).

c) NIVEAU D – DOCTORAT

- 2004-2005 Réseaux de neurones et réseaux neuro-flous pour la surveillance, Ecole Doctorale SPIM (École Doctorale Sciences Physiques pour l'Ingénieur et Microtechniques) de l'Université de Franche-Comté, 10h C (cours entièrement crée basé sur nos derniers travaux de recherche, réactualisé notamment avec l'aspect neuro-flou et un état de l'art complet des techniques de l'IA pour le diagnostic, env. 10 étudiants par an).

2.3.3 FORMATION CONTINUE

Décembre 2004

« Techniques de l'Intelligence artificielle pour la surveillance industrielle », chapitre : Techniques et Technologies de la e-Maintenance, section : Techniques pour les processus « Métier » de la e-Maintenance cours à destination du personnel de maintenance (15 personnes) de la société DCN (Division des Constructions Navales) dans le cadre du contrat DCN – MACOD/GdR MACS : séance 3/3 du 15-16 décembre 2004, Besançon.

2.3.4 SUIVI D'APPRENTIS, ENCADREMENT DE PROJETS ET STAGES

Etant donné mon expérience industrielle et ma sensibilité concernant cette activité, depuis mon arrivée à l'UFR Sciences je n'ai pas hésité à m'impliquer – à chaque fois que le besoin s'en est fait sentir – dans les activités d'encadrement de projets de différents niveaux, de suivi de stagiaires et d'apprentis. Dans ce sens, je peux énumérer les activités suivantes :

1999-2004 Encadrement de projets de binômes d'étudiants de DESS PA (200h de projet par étudiant, un ou deux binômes par an) :

Entreprise	Sujet du Projet DESS PA	Année
FCI – Framatome, Besançon (25)	Inspection d'assemblage de petites dimensions par vision - Optimisation d'un système de vision - contrôle dimensionnel de connecteurs	1999-2000
HUMBERT Outillage, Le Luhier (25)	Remise en cause globale de l'organisation actuelle des méthodes de production d'une entreprise de mécanique générale	1999-2000
Fromageries BEL, Lons le Saunier (39)	Adéquation d'un stock de pièces de rechange en fonction des contraintes d'exploitation de l'outil de production	2000-2001
GAUSSIN, Héricourt (70)	Conception de systèmes de direction de remorques sur le principe du suivi monotrAce	2000-2001
SANIJURA, Champagnole (39)	Réorganisation du secteur usinage	2000-2001
HORIS – TGCP, Poligny (39)	Analyse et propositions d'amélioration de nos process de soudure	2001-2002
SANIJURA, Champagnole (39)	Automatisation de petits postes de travail flexibles	2001-2002
SANIJURA, Champagnole (39)	Définition des moyens nécessaires à la mise en place d'un laboratoire interne	2002-2003
SANIJURA, Champagnole (39)	Réorganisation de l'îlot Assemblage "Panneau Miroir"	2002-2003
AVENSY Ingénierie, Besançon (25)	Intégration d'une solution de surveillance temps réel neuronale dans une solution globale de surveillance à distance des systèmes de production, basée sur la plate-forme AQUIVUS et un module MSP.	2003-2004
SANIJURA, Champagnole (39)	Étude d'une ligne de montage finition et emballage de meubles sous vasque	2003-2004

2004-2005 Encadrement de projets de binômes d'étudiants de Master MT 2^{ème} année parcours Professionnel (100h de projet par étudiant, deux binômes) :

Entreprise	Sujet du Projet d'innovation MMT Professionnel	Année
AVENSY Ingénierie, Besançon (25)	Etude d'une solution de maintenance préventive et corrective concernant l'industrie de la machine-outil.	2004-2005
FABRICOM, Besançon (25)	Création de modules spécifiques de simulation sur Witness. Etude d'un cas de simulation de flux	2004-2005
SANIJURA, Champagnole (39)	Equipement d'un laboratoire d'essais. Conception et prototypage d'une machine spéciale de test des charnières de portes et de tiroirs.	2004-2005

1999-2004 Encadrement de projets de maîtrise EEA (un ou deux par an) au LAB (binôme ou trinôme d'étudiants, 80h étudiant par projet réparties d'octobre à avril)

L. Baronchelli, G. Clerget, Optimisation des programmes GRAFCET du système flexible de production du LAB (1999-2000)

N. Palluat, M. Dauge, Supervision du système de production LAB (2000-2001)

B. Aydin, S. Michel, Système de télémaintenance du transfert industriel flexible du LAB (2001-2002)

E. Gresset Bourgeois, P. Ndimurukundo, Surveillance en ligne basée sur les réseaux de neurones : modélisation VHDL du RBF statique (2001-2002)

M. Monnin, M. Paris, Programmation Java d'une interface de surveillance en ligne via Internet à l'aide d'un automate SIEMENS avec coupleur serveur web (2002-2003)

C. Benuto, J. Romain, Etude d'un système de surveillance en ligne à l'aide d'une technique basée sur les réseaux de neurones : modélisation VHDL du RBF dynamique (2002-2003)

C. Benuto, T.-T. Le, M. Servelle, Création d'une base de connaissances des défaillances sur le système flexible du LAB (2003-2004)

2000-2003 Encadrement de projets de fin d'étude ENSMM (Ecole Nationale Supérieure de Mécanique et Microtechniques de Besançon)

S. Pross, S. Tararykine, Programmation du superviseur Monitor Pro du transfert flexible du LAB (2000-2001)

C. Philippe, Télémaintenance et diagnostic préventif, projet en collaboration avec la société Predict de Nancy (2001-2002)

J. Fenolar, J. Del Vecchio, Système de surveillance temps réel basé sur les réseaux de neurones dynamiques, accessibles à distance par un serveur web, développement du prototype et étude du marché, projet en collaboration avec la société Avensty Ingénierie Besançon (2002-2003)

2003-2005 Tuteur d'un étudiant IUP SAPIAA 1^{ère} année, puis en 2^{ème} année, en apprentissage chez Nestlé Dijon (soutien technique, visites régulières dans l'entreprise).

- 1999-2005 Tutorat de stages DESS PA et Master Professionnel Mécatronique (1 à 2 stages par an). Soutien technique, bibliographique, méthodologique et visites des stagiaires
- 2004-2005 Encadrement d'un binôme Licence GPI (gestion de production intégrée) en projet industriel :

Entreprise	Sujet du Projet Licence SDPI	Année
FCI SA, Besançon (25)	Flux composants inter-ateliers	2004-2005

- 1999-2000 Encadrement d'un projet Licence EEA (20h de projet par étudiant)

2.3.5 RELATIONS INDUSTRIELLES PAR L'ENSEIGNEMENT

Les activités pédagogiques m'offrent l'opportunité d'être fréquemment en contact avec le tissu industriel national (une quinzaine de visites par an) notamment par l'intermédiaire des projets et stages industriels Master Mécatronique Professionnel (anciennement DESS Productique Appliquée) et du suivi des apprentis d'IUP SAPIAA (Systèmes Automatisés de Production dans le Industries Agro-Alimentaires).

Dans le cadre des projets de Master MT, il s'agit de travailler dans nos locaux sur un problème industriel posé par une entreprise régionale. Mon rôle consiste alors à encadrer les étudiants à l'UFR et à me rendre régulièrement avec eux dans l'entreprise concernée pour réaliser un audit, effectuer une étude ou mettre en place une solution technique ou organisationnelle.

Par ailleurs, la création en 1998 de l'IUP SAPIAA s'est effectuée en lien très étroit avec l'industrie, dans la mesure où la grande majorité des étudiants de ce cursus sont en contrat d'apprentissage. A ce titre, je suis notamment tuteur enseignant d'un étudiant en contrat d'apprentissage pour trois ans (IUP1 → IUP 3) avec l'entreprise NESTLE de Dijon. Je suis donc amené à visiter régulièrement l'entreprise pour suivre cet apprenti et si nécessaire lui fournir une aide dans ces travaux.

2.3.6 AUTRES ACTIVITES ET RESPONSABILITES POUR L'ENSEIGNEMENT

A partir de 2004, j'ai pris en charge la responsabilité de la gestion des projets industriels d'innovation de 2^{ème} année de Master Mécatronique Professionnel. Cette activité consiste à contacter les entreprises régionales afin de faire un bilan de leur besoins dans nos domaines de compétences et de leur proposer un soutien via ce type de projet, dans lequel nous apportons – du côté pédagogique – un soutien en terme d'outils de calcul, d'analyse, de simulation, de plates-formes de test (automatique, productique, robotique, maintenance, vision industrielle) et bien évidemment un soutien méthodologique et bibliographique. Ces projets impliquent un volume de 200 h d'études, avec une soutenance intermédiaire et une soutenance finale, ainsi qu'un compte rendu de lancement et des comptes-rendus mensuels d'avancement, éléments indispensables à un bon apprentissage de la gestion de projet.

Toujours à partir de 2004, je suis le responsable au sein du Groupe Automatique et Productique du module découverte EEA dans le cadre du semestre d'intégration dans le monde universitaire et d'orientation dit STARTER et notamment du STARTER ST (Sciences et Techniques). Je participe à la gestion par le GAP de ce module très important pour nos formations car permettant de donner envie aux jeunes étudiants de découvrir les métiers de l'EEA, donc d'alimenter nos différentes filières d'enseignement. Environ 280

étudiants sont concernés en 2004-2005 par ce module qui consiste à effectuer un ensemble de TD et de TP autour des métiers de l'EEA (11h TD, 9h TP). La partie dont nous nous occupons plus particulièrement consiste à effectuer des interventions sur nos plates-formes de robotique, productique et asservissement.

La mise en place du LMD à l'UFR Science s'est accompagnée de certaines autres responsabilités nouvelles, associées à ce nouveau mode de gestion de l'enseignement supérieur. C'est ainsi que je fais partie de la commission de recrutement et de la commission de validation d'acquis du master mécanique.

A partir de 2000 j'ai mis en place un TP de temps réel nouveau dans le cadre des formations maîtrise EEA et IUP Génie Informatique. L'illustration de ce concept n'est pas une chose facile en dehors du monde de l'électronique pure. Après une réflexion approfondie et de multiples discussions avec les industriels (participation aux workshops du salon annuel national du temps réel) et les scientifiques (réunions du GdR STRQDS) du domaine - j'ai opté pour des travaux qui s'orientent vers la spécification, la vérification, le test et la génération automatique de code d'applications temps réel. Ceci rend la réflexion relativement générique, indépendante de la cible choisie, en utilisant des outils modernes, situés aux limites de la recherche dans ce domaine, comme le synch-chart mis à notre disposition dans le cadre d'un accord avec la société Esterel Technologies et INRIA.

Entre septembre 1999 et août 2004, j'ai assuré la responsabilité de l'option Automatique et Productique de la maîtrise EEA de l'UFR Sciences.

Pour finir, je suis le responsable d'un projet d'échange et de collaboration Erasmus entre l'Université de Franche-Comté et l'Université Valahia de Targoviste. Dans ce cadre, à partir de l'année universitaire 2004-2005, un échange d'étudiants et d'enseignants est mis en place. Dans ce contexte, 2 professeurs roumains et 3 collègues enseignants-chercheurs du GAP ont effectué une mission d'enseignement et d'échange d'expériences. Ce projet a été reconduit pour l'année universitaire 2005-2006.



Partie II - Contribution Scientifique

Introduction générale

Les industriels, habitués – dans les années passées – à des marges relativement confortables, se trouvent actuellement face à un marché de plus en plus concurrentiel, sur lequel la compétitivité est conditionnée, dans une mesure de plus en plus importante, par l'exploitation optimale des moyens humains et matériels de l'entreprise. Dans ce contexte, la maintenance industrielle connaît actuellement un essor spectaculaire, dû à la possibilité de puiser dans les ressources existantes de l'entreprise afin d'améliorer ses performances.

Un environnement international à forte connotation TIC (Technologies de l'Information et de la Communication), permet à l'heure actuelle la mise en place de techniques innovantes de gestion de la maintenance, en passant par la télémaintenance vers la e-maintenance, concept coopératif de partage des connaissances du métier. Loin d'être une simple instrumentation d'un métier déjà existant, la e-maintenance ouvre ainsi des perspectives innovantes de création de nouveaux services, capables d'accompagner les clients sur la durée et avec une compétence soucieuse de sa capitalisation et de son évolution permanente.

Cet environnement technologique permet aux entreprises soit d'externaliser la fonction maintenance pour mieux se concentrer sur leur activité principale, soit de passer de la maintenance classique à la maintenance à distance et en temps réel. Ceci est un enjeu important, au centre duquel la surveillance joue un rôle majeur, dans un contexte correspondant à une intelligence de plus en plus répartie, sujet à une migration vers les niveaux opérationnels (capteurs et actionneurs intelligents).

On voit naître par conséquent un intérêt croissant pour la maintenance intelligente, dans laquelle la surveillance et sa composante principale – le diagnostic – occupent une place fondamentale, en glissant de plus en plus vers des applications de surveillance dynamique et par conséquent, vers le pronostic.

Dans ce sens, l'utilisation des techniques de l'intelligence artificielle fait l'objet de nombreuses recherches. La possibilité de surveiller sans modéliser le système, dans un environnement ou celui-ci est soumis à des adaptations et des reconfigurations permanentes, présente un intérêt évident pour les industriels.

Notre approche exploite ces caractéristiques essentielles, en focalisant sur les capacités d'apprentissage des réseaux de neurones, avec comme objectif la souplesse des architectures et des algorithmes développés ainsi que la possibilité d'évoluer dans un contexte dynamique. Dans cet esprit, une première partie de nos recherches est consacrée à la mise au point de structures et algorithmes d'apprentissage dynamiques, dédiés à la surveillance dynamique des systèmes industriels. La capacité de répondre « je ne sais pas » à une aide au diagnostic nous paraît ainsi essentielle, en guidant nos études vers des approches neuronales locales et plus précisément vers des structures issues du réseau à fonctions de base radiales (RFR).

Dans le souci de mettre au point des outils capables de fournir une aide à la recherche de cause d'une défaillance ainsi que d'interpréter des connaissances de maintenance dans

un format industriel standard, nous avons étudié la possibilité de développer des outils de surveillance dynamique de la propagation des défauts. Nous avons ainsi focalisé une partie de nos recherches sur l'exploitation de l'information fournie par l'arbre de défaillance (ADD). La succession d'événements discrets que caractérise la propagation des défauts des systèmes de production manufacturiers nous a ainsi incité à utiliser un outil de type réseau de Petri. Nous y avons associé la logique floue, afin de prendre en compte un élément essentiel : le retour d'expérience. C'est ainsi que nous avons transformé un arbre de défaillance classique en un outil de pronostic de défauts, capable de signaler l'apparition et la possibilité de propagation des dégradations, bien avant que la défaillance apparaisse.

Enfin, des études approfondies ont été menées sur l'aide au diagnostic en utilisant les systèmes neuro-flous. Ces systèmes présentent la capacité de prendre en compte une connaissance incertaine et imprécise – souvent seuls éléments dont disposent les opérateurs dans la pratique – tout en permettant l'apprentissage dynamique des situations nouvelles. Ce type d'outil semble très intéressant dans ce contexte d'utilisation. Le nombre réduit de travaux sur ce sujet nous laisse envisager une suite prometteuse de ce type d'approche.

Ce rapport scientifique est structuré en cinq parties.

Afin d'introduire le lecteur à nos travaux de recherche, nous allons commencer par présenter des généralités sur le domaine de la télémaintenance et de son évolution : la e-maintenance, pour focaliser par la suite nos propos sur la partie « intelligente » de la maintenance, notamment celle qui utilise les techniques de surveillance par Intelligence Artificielle (IA) et qui constitue le cœur de nos travaux.

Par la suite, des développements concernant un réseau de neurones dynamique innovant font l'objet de la deuxième partie. Parmi toutes les architectures neuronales étudiées, celle qui apparaît la mieux adaptée aux problématiques de la surveillance dynamique est l'architecture des réseaux de neurones récurrents, seuls réseaux de neurones qui possèdent une mémoire dynamique interne. L'inconvénient majeur de ces réseaux récurrents – la lourdeur et la complexité d'apprentissage – a été contourné en prenant en compte l'aspect temporel par des récurrences locales. Ces récurrences locales sont tolérées uniquement au niveau du neurone lui-même, sans trop compliquer l'architecture globale du réseau de neurones. Cette partie est donc consacrée à l'étude des réseaux à représentation locale du temps qui sont appelés architectures *LRGF* : *Locally Recurrent Globally Feedforward* ou architectures *Localement Récurrente Globalement Feedforward*. Le but est de démontrer, grâce à des développements mathématiques appuyés par des simulations informatiques, que les architectures *LRGF* possèdent des caractéristiques dynamiques très intéressantes. Cette étude nous permet de justifier le choix du type de mémoire dynamique afin de proposer une nouvelle architecture d'un réseau *RFR* récurrent. Le Réseau Récurrent à Fonctions de base Radiales que nous proposons profite donc des performances des réseaux *RFR* avec l'aspect dynamique à la fois performant et simple des architectures *LRGF*.

L'évaluation des performances du réseau récurrent proposé, fait l'objet de la troisième partie. Cette évaluation se fait sur trois types d'applications dynamiques : *la reconnaissance de séquences booléennes et réelles*, *la prédiction de séries temporelles* et *la reproduction de séries temporelles*. Ces trois catégories d'applications sont très importantes en surveillance dynamique. Nous démontrons que l'architecture *LRGF* adoptée permet au réseau *RRFR* d'acquérir des propriétés dynamiques, tout en gardant la simplicité et l'efficacité des réseaux *RFR*. La phase de paramétrage du réseau *RRFR* est effectuée par une version améliorée de l'algorithme d'apprentissage des *k-moyennes* que nous proposons. En effet, cette version dynamique de cet algorithme d'apprentissage procure une stabilité du résultat, ainsi qu'une robustesse accrue par rapport à la phase de paramétrage. Cet avantage est mis en évidence à travers un développement de

e-surveillance, mené en collaboration avec un partenaire industriel bisontin – Avensty Ingénierie.

La quatrième partie du rapport scientifique est dédiée à une étude de surveillance dynamique en utilisant des outils de type réseaux de Petri flous. Nous proposons ainsi une approche globale homogène de modélisation utilisant deux outils intégrés de type réseau de Petri flous, dédiés à la surveillance et la maintenance des systèmes de production. Un mécanisme d'interfaçage approprié sera intégré à chaque outil, afin de rendre possible la mise en place de la boucle système - surveillance – décision. Dans la pratique de la maintenance industrielle, cette boucle est utilisée en tant que système d'aide à la décision pour la maintenance et comprend deux étapes supplémentaires fondamentales : la validation par le responsable de maintenance et la mise en œuvre par un opérateur spécialisé. L'intégration de la logique floue dans un tel système d'aide à la décision permet la mise en œuvre d'une formalisation proche de la réalité, du savoir-faire et de la culture de l'entreprise concernée, ainsi qu'une flexibilité et une dimension prédictive très intéressantes.

Enfin, la dernière partie aborde nos derniers développements concernant l'utilisation des systèmes neuro-flous pour l'aide au diagnostic et au pronostic. Démarré dans le cadre d'un projet européen sur la e-maintenance, ce travail permet l'exploitation des avantages des réseaux de neurones dynamiques (apprentissage dynamique) et de la logique floue (modélisation du raisonnement humain), sous la forme d'un service d'aide au diagnostic faisant partie d'une plate-forme de e-maintenance. La spécification du système suit le formalisme UML, en débouchant sur un prototypage mis au point dans le cadre d'un projet Anvar, avec la collaboration de notre partenaire industriel Cegelec.

Chapitre 3 - Problématique scientifique : Surveillance intelligente

3.1 INTRODUCTION

Mon recrutement au Laboratoire d'Automatique de Besançon en 1999 avait pour but de renforcer le groupe Productique et participer à la création et à l'organisation dans ce laboratoire d'une activité autour de la e-maintenance et plus particulièrement la maintenance intelligente. Ces domaines émergeaient alors au plan national, d'autre pays comme les Etats-Unis et le Japon ayant enregistré déjà quelques initiatives dans ce sens.

A titre d'exemple, nous pouvons citer le programme de l'armée américaine, intitulé « Machinery Information Management Open Systems Alliance » - MIMOSA¹⁵ qui développe et encourage l'adoption de standards ouverts d'information pour les opérations et pour la maintenance dans les systèmes de production, les flottes (maritimes, aériennes, ...) et les installations. L'armée américaine ayant développé des outils logiciels très sophistiqués (mais aussi très chers) pour supporter ses activités de maintenance, il est apparu que ces outils ne communiquaient pas et étaient peu ou pas intégrés du tout. La mise en réseau et l'intégration des multiples systèmes utilisés par l'armée américaine, ainsi que le support de ses relations de maintenance avec leurs différents fournisseurs, sont les principaux acquis de ces développements. Mimosa vise donc le développement des solutions de maintenance ouvertes, moins chères et plus rapides à installer et à utiliser. Les projets développés englobent dans un seul système, les informations nécessaires pour la maintenance de base, corrective, mais aussi préventive. A tout cela s'ajoute bien évidemment une standardisation très poussée.

Un autre exemple important sur le plan international est constitué par le centre Intelligent Maintenance Systems¹⁶ (IMS) de coopération industrie/université autour de la maintenance, crée sous l'égide du National Science Foundation¹⁷, avec la participation d'Universités comme celle de Wisconsin – Milwaukee et de Michigan et d'un nombre important d'industriels de marque (Automation, Intel, Harley-Davidson, We Energies, General Motors, United Technologies, USPS, Hitachi). Dans le cadre de ce centre, l'axe fondamental est représenté par la surveillance prédictive et le pronostic, dans le but de permettre le rapprochement d'un fonctionnement « zéro pannes » d'un système. Le centre IMS développe ainsi des technologies dans quatre domaines de recherche incluant les systèmes de type « chien de garde »¹⁸ pour le pronostic, ainsi que des outils d'aide à la décision en maintenance au niveau système.

¹⁵ Machinery Information Management Open Systems Alliance - MIMOSA : <http://mimosa.org/>

¹⁶ Center for Intelligent Maintenance Systems - IMS : <http://wumrc.engin.umich.edu/ims/>

¹⁷ National Science Foundation – NSF : <http://www.nsf.gov/>

¹⁸ WATCHDOG AGENTTM – marque déposée par le centre IMS

En ce qui concerne le Japon, le sujet semble constituer actuellement un enjeu national pour les industriels et pour l'état. A titre d'exemple, à Waseda University¹⁹ un projet industriel intitulé « Development of New Maintenance Technologies and Web based Services : Construction of a Virtual Community for Next Generation Maintenance » a comme finalité (en collaboration avec la compagnie d'électricité ASAKI), la capitalisation de l'expertise en diagnostic et maintenance, face aux évolutions démographiques dans les entreprises (sauvegarde du savoir-faire).

Dans ce contexte, la création au LAB d'une équipe de recherche dans le domaine de *l'ingénierie de maintenance et e-maintenance* a été favorisée par un environnement national et international à forte connotation NTIC (Nouvelles Technologies de l'Information et de la Communication), marqué au niveau national par la création en automne 2000 du pôle Sciences et Technologies de l'Information et de la Communication (STIC) du CNRS.

Profitant de cette évolution technologique très impressionnante, l'intelligence se trouve de plus en plus distribuée et se rapproche des niveaux bas, opérationnels des systèmes automatisés. Dans ce sens, nos contributions concernent le développement de techniques de l'intelligence artificielle pour la surveillance, étape fondamentale de la fonction maintenance.

Afin d'introduire le lecteur à nos travaux de recherche, nous allons commencer par présenter des généralités sur le domaine de la télémaintenance et de son évolution : la e-maintenance, pour focaliser par la suite nos propos sur la partie « intelligente » de la maintenance, notamment celle qui utilise les techniques de surveillance par Intelligence Artificielle (IA) et qui constitue le cœur de nos travaux.

3.2 MAINTENANCE ET SURVEILLANCE INDUSTRIELLE DANS LE CONTEXTE DE LA PRODUCTION MODERNE : TELEMaintenance, E-MAINTENANCE, MAINTENANCE ET SURVEILLANCE INTELLIGENTE

La maintenance,

ensemble de toutes les actions techniques, administratives et de gestion, durant le cycle de vie d'un bien, destinée à le maintenir ou le rétablir dans un état dans lequel il peut accomplir une fonction requise,

a longtemps été « l'enfant pauvre » des stratégies industrielles. Mis à part quelques secteurs comme l'aéronautique, les raffineries, le nucléaire et l'espace (Dekker *et al.*, 1998) qui – soumis à des contraintes très strictes de sécurité – ont été obligés de prendre à bras le corps ce problème depuis longtemps, l'industrie des biens de consommation (Richet *et al.*, 1996) se « réveille » depuis peu, en trouvant dans cette fonction une source nouvelle de profit et d'optimisation des moyens de production existants, dans un contexte économique très exigeant, dans lequel les entreprises européennes se voient contraintes d'exploiter les moindres niches technologiques et organisationnelles afin de rester compétitives.

¹⁹ Waseda University, Graduate School of Information, Production and Systems - <http://www.waseda.ac.jp/index-e.html>

Dans cet environnement industriel, les contraintes de coûts de réalisation technologique, l'accessibilité des sites et la répartition géographique obligent à distribuer les données ou les traitements de maintenance. Les accès sont donc distants donnant ainsi lieu à la *télémaintenance*.

Par définition, la télémaintenance (en anglais : remote maintenance) représente la maintenance d'une unité fonctionnelle, assurée par télécommunication directe entre cette unité et un centre spécialisé.

La télémaintenance s'impose progressivement pour toutes les machines ou les chaînes de production pilotées par un microprocesseur : celui-ci concentre en effet toutes les informations provenant des multiples capteurs nécessaires au contrôle du processus et il peut être facilement interrogé à distance. Il est également possible depuis le point de contrôle de modifier les paramètres de réglage ainsi que les programmes qui le pilotent.

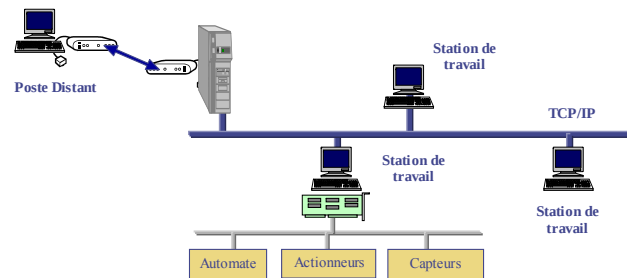


Figure 7. Schéma générique de la télémaintenance.

Depuis l'émergence des NTIC, les concepts de la Télémaintenance ont évolué pour aboutir à la *e-Maintenance* :

- ✚ La Télémaintenance est un concept de récupération de données à distance ou de prise de contrôle à distance (éphémère)
- ✚ La e-Maintenance est un concept lié au principe de *web-services*, de *coopération* et de *partage des connaissances*

La e-Maintenance (Leger, 2004) intègre le principe de base de la Télémaintenance en lui associant une dimension forte, constituée par la *coopération* au niveau des informations mais aussi des hommes, des services (ingénierie, exploitation, maintenance, sûreté, achats, comptabilité, ...) et des sociétés (client / fournisseur, inter fournisseurs, inter clients, ...).

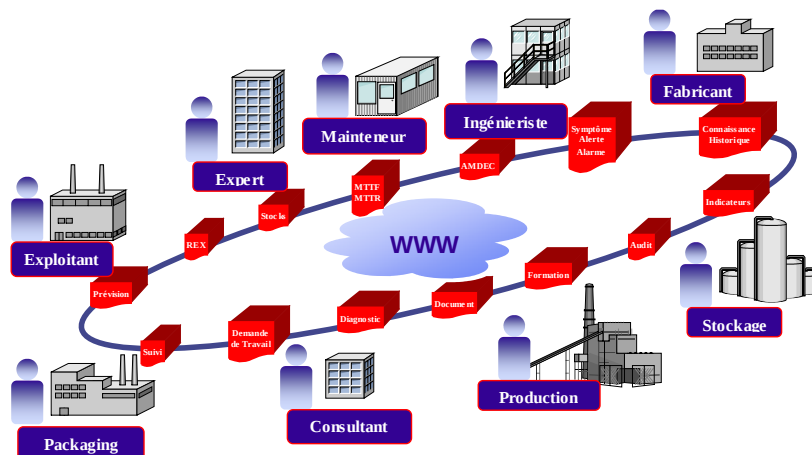


Figure 8. Schéma générique de la e-maintenance.

En effectuant un bilan des différentes évolutions des politiques de maintenance dans le temps, nous pouvons donner à titre indicatif les repères suivants :

✚ Années 60 : Maintenance réactive / corrective

Intervient après la détection et la localisation d'un défaut. Elle revient à corriger entièrement (curative) ou en partie (palliative) le défaut observé

✚ Années 70 : Maintenance préventive (préventive systématique)

Effectuée dans l'intention de réduire la probabilité de défaillance d'un bien ou la dégradation d'un service rendu. C'est une intervention de maintenance prévue, préparée et programmée avant la date probable d'apparition d'une défaillance. La maintenance préventive la plus courant (surtout dans cette période) est la maintenance préventive systématique. Elle représente une maintenance préventive effectuée selon un échéancier établi suivant le temps ou le nombre d'unités d'usage

✚ Années 80: Maintenance prédictive
(prévisionnelle, préventive conditionnelle)

Basée sur la surveillance en continu de l'évolution du système, afin de prévenir un dysfonctionnement avant qu'il n'arrive. Elle n'implique pas la connaissance de la loi de dégradation. La décision d'intervention préventive est prise lorsqu'il y a évidence expérimentale de défaut imminent, ou approche d'un seuil de dégradation prédéterminé. Elle impose des traitements en ligne, au moins en partie. Dans la pratique industrielle, cette maintenance est basée sur l'analyse des mesures de certains paramètres de l'équipement: température, vibration, qualité des huiles, ..., pour tenter d'éviter un dysfonctionnement.

✚ Années 90 : Maintenance proactive (PRM)

Implique la surveillance et la vérification continues des causes primaires de défaillance du système surveillé

✚ Années 2000 : E-maintenance

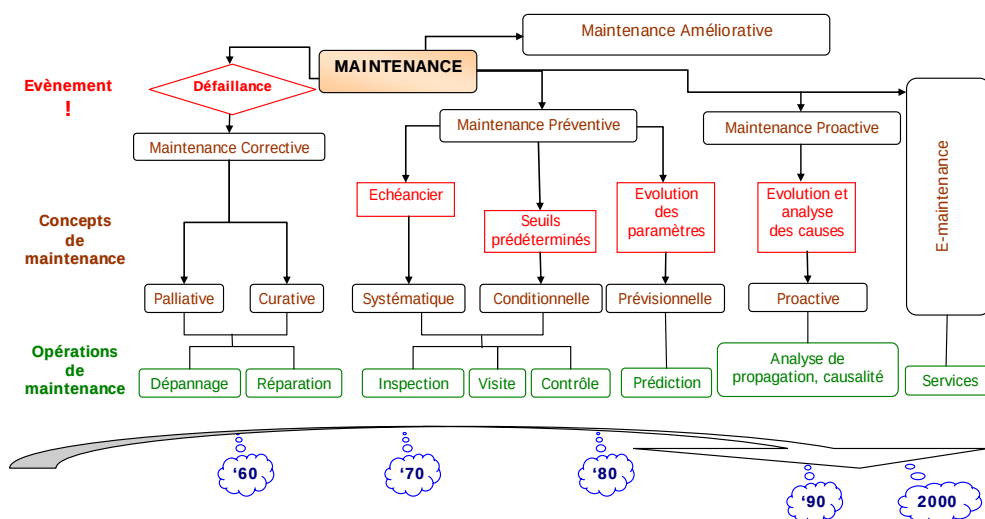


Figure 9. Différentes politiques de maintenance

Remarque :

Vu l'importance accordée à la propagation des défaillances/dégradations ainsi qu'à la recherche de causes durant nos recherches, nos études se positionnent essentiellement

dans le cadre de la maintenance proactive, avec des applications et des développements effectués en évoluant vers la création de services distribués et donc vers la e-maintenance.

Le développement actuel des Technologies de l'Information et de la Communication (TIC), facilite la mise en place d'une vraie stratégie de e-maintenance. Ceci a provoqué une évolution importante de l'organisation des entreprises. Celles-ci ont tendance soit à externaliser la fonction maintenance pour mieux se concentrer sur leur activité principale, soit à passer de la maintenance classique à la maintenance à distance et en temps réel. Ceci est un enjeu important, au centre duquel la surveillance joue un rôle majeur, dans un contexte correspondant à une intelligence de plus en plus répartie :

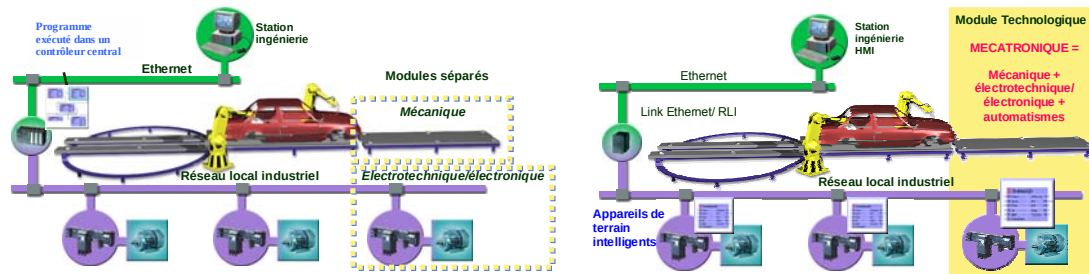


Figure 10. Hier : concept modulaire avec intelligence centralisée - Demain : concept modulaire avec intelligence répartie

Par ailleurs, dans le domaine du contrôle industriel et des automatismes, la tendance actuelle est à la migration de l'intelligence vers le bas (capteurs et actionneurs intelligents) :

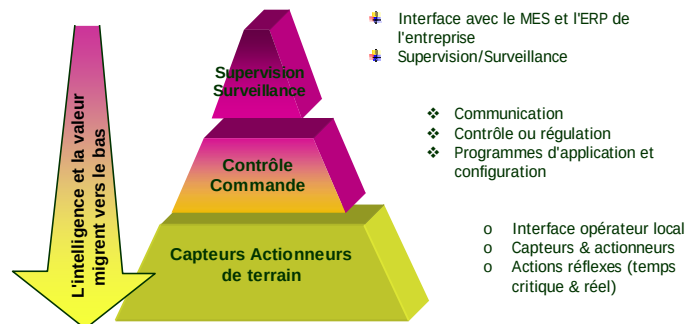


Figure 11. Tendances dans les systèmes de contrôle industriel et d'automatismes

Les composantes du système de surveillance se caractérisent ainsi par une autonomie de plus en plus importante, en travaillant dans des systèmes distribués et en intégrant souvent une intelligence embarquée.

D'une importance capitale dans la maintenance industrielle – aussi bien dans sa dimension conditionnelle, prévisionnelle, que du point de vue d'un retour d'expérience efficace, structuré et systématique –, la surveillance prend une dimension particulière grâce à la distribution de l'intelligence aux niveaux les plus bas.

Dans le contexte actuel d'évolution des systèmes de production automatisés et de la maintenance moderne, le rôle de la surveillance industrielle évolue en acquérant une dimension dynamique voir même prédictive, grâce à la possibilité d'exploiter l'intelligence répartie dans le système concerné, afin d'en extraire un maximum d'informations élaborées, en temps réel et d'augmenter la réactivité du système de surveillance.

3.3 SURVEILLANCE INDUSTRIELLE : DEFINITIONS, ETAT DE L'ART

3.3.1 INTRODUCTION

Dans un grand nombre d'applications industrielles, une demande croissante est apparue en matière de remplacement des politiques de maintenance curative par des stratégies de maintenance préventive. Cette mutation d'une situation où on « *subit les pannes* » à une situation où on « *maîtrise les pannes* », nécessite quelques moyens technologiques ainsi que la connaissance de techniques d'analyse appropriées. La fonction *surveillance* en continu de l'évolution de l'équipement à travers des données quantifiables et qualifiables permet ainsi de prévenir un dysfonctionnement avant qu'il n'arrive et d'écarter les fausses alarmes qui peuvent ralentir la production (Basseville *et al.*, 1996). De nombreux auteurs ont abordé le domaine de la surveillance industrielle mettant ainsi en évidence l'intérêt croissant manifesté par la communauté scientifique et les industriels par rapport à cette problématique. Sans souci d'exhaustivité, nous pouvons citer les travaux suivants : (Combacau, 1991), (Devauchelle, 1991), (Toguyeni, 1992), (Poulard, 1996), (Cussenot, 1996), (Evsukoff, 1998), (Weber, 1999), (Zhang, 1999), (Combastel, 2000), (Lefebvre, 2000).

Dans la littérature associée au domaine de la surveillance d'équipements industriels, on peut trouver plusieurs définitions quelquefois divergentes. C'est pour cette raison que nous considérons important d'affirmer notre point de vue, en donnant des définitions des notions de base indispensables au positionnement de nos travaux de recherche.

3.3.2 DEFINITIONS DES TERMES DE BASE UTILISES EN SURVEILLANCE

La diversité des définitions trouvées dans différents travaux fait que nous avons jugé important d'établir un lexique sur les termes qui seront utiles pour la compréhension du présent rapport. Ces définitions ont été extraites pour certaines à partir des références suivantes : (Villemeur, 1988), (Dubuisson, 1990), (Combacau, 1991), (Toguyeni, 1992), (Zwingelstein, 1995), (Basseville *et al.*, 1996), (Lefebvre, 2000). On peut toutefois trouver en littérature des définitions qui sont complètement différentes de celles que nous proposons, mais ceci nous permet de présenter notre point de vue sur la surveillance industrielle.

Dégradation

Une dégradation représente une perte de performances d'une des fonctions assurées par un équipement.

Si les performances sont au-dessous du seuil d'arrêt défini dans les spécifications fonctionnelles de cet équipement, il n'y a plus dégradation mais défaillance.

Défaillance

Une défaillance est l'altération ou la cessation de l'aptitude d'un ensemble à accomplir sa ou ses fonctions requises avec les performances définies dans les spécifications techniques.

Panne

Une panne est l'inaptitude d'une entité (composant ou système) à assurer une fonction requise.

Si nous écartons la possibilité d'erreurs de conception, la définition précédente implique que toute défaillance entraîne une panne. La défaillance correspond à un événement et la panne à un état. Sur le plan temporel, la défaillance correspond à une date et la panne à une durée comprise entre la date d'occurrence de la défaillance et la date de fin de réparation.

Mode de fonctionnement

L'un des états possibles d'un élément défaillant, pour une fonction exigée donnée

Surveillance

*La surveillance est un dispositif **passif, informationnel**, qui analyse l'état du système et fournit des indicateurs. La surveillance consiste notamment à **détecter** et **classer** les défaillances en observant l'évolution du système, puis à les **diagnostiquer** en **localisant** les éléments défaillants et en **identifiant** les causes premières.*

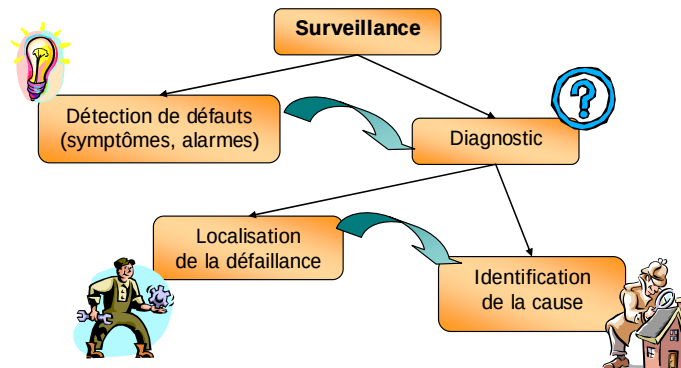


Figure 12. Composantes de la surveillance industrielle

La surveillance se compose donc de deux fonctions principales qui sont la *détection* et le *diagnostic* :

Détection

*Pour **détecter** les défaillances du système, il faut être capable de **classer** les situations observables comme étant **normales** ou **anormales**.*

Cette classification n'est pas triviale, étant donnée le manque d'information qui caractérise généralement les situations anormales. Une simplification communément adoptée consiste à considérer comme anormale toute situation qui n'est pas normale.

Diagnostic

L'objectif de la fonction diagnostic est de rechercher les causes et de localiser les organes qui ont entraîné une observation particulière.

*La fonction de diagnostic se décompose en deux fonctions élémentaires : **localisation** et **identification des causes**. La localisation permet de déterminer le sous-ensemble fonctionnel défaillant tandis que l'identification de la cause consiste à déterminer les causes qui ont mené à une situation anormale.*

A partir de l'observation d'un état de panne, la fonction diagnostic est chargée de retrouver la faute qui en est à l'origine. Ce problème est difficile à résoudre. En effet si, pour une faute donnée, il est facile de prédire la panne résultante, la démarche inverse qui consiste à identifier la faute à partir de ses effets, est beaucoup plus ardue. Une défaillance peut généralement être expliquée par plusieurs fautes. Il s'agit alors de confronter les observations pour fournir la bonne explication.

Dans la plupart des configurations industrielles, la surveillance se fait en ligne, dans ce cas, la figure suivante synthétise l'ensemble des fonctions connexes, en positionnant la surveillance dans le cadre de la supervision (Basseville et al., 1996) :

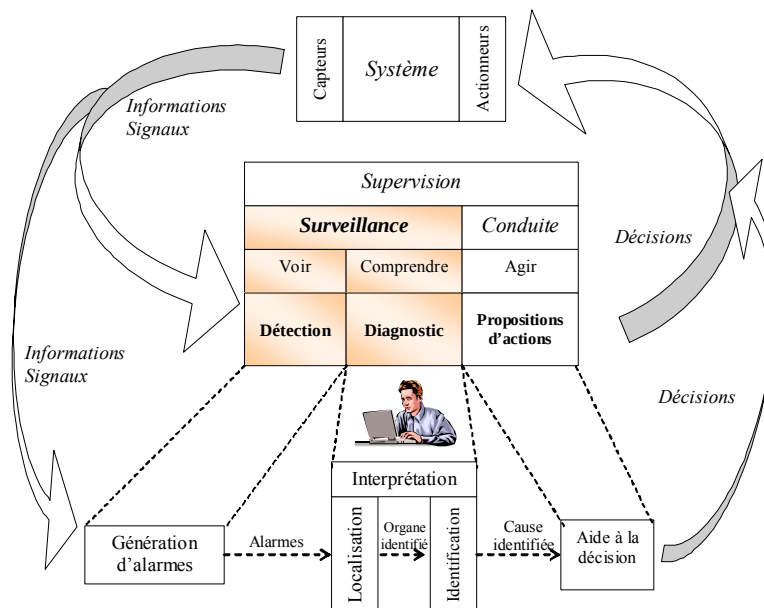


Figure 13. Architecture générale d'un système de supervision en ligne

Pour surveiller en continu l'évolution d'un système et déterminer les causes des événements indésirables, il semble donc intéressant d'utiliser au mieux toutes les informations disponibles, qu'elles soient numériques (données capteurs), ou symboliques (historiques, environnements, réparations effectuées,...). On retrouve donc dans la littérature le terme "d'approche unifiée" pour la surveillance, qui caractérise les système de surveillance prenant en compte les données numériques et symboliques :

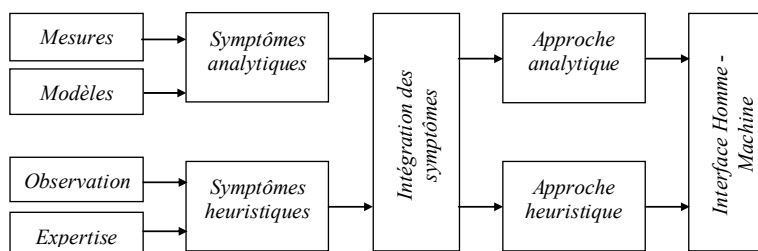


Figure 14. Approche unifiée pour la surveillance

Avec l'approche analytique, on considère les informations quantifiables issues d'un modèle numérique fournissant des symptômes analytiques. L'approche heuristique permet de compléter la première en prenant en compte les informations qualitatives sur les systèmes surveillés tirées de l'expertise humaine.

Surveillance prédictive (dynamique)

Comme pour la surveillance classique, la surveillance prédictive est un dispositif **passif, informationnel**, qui analyse l'état présent et passé du système et fournit des indicateurs sur les tendances d'évolution futur du système. La surveillance prédictive se compose de : la **Détection prédictive (dynamique)** et du **Diagnostic prédictif, appelé aussi pronostic**.

Détection prédictive (dynamique)

La détection prédictive consiste à prédire une défaillance future. En d'autres termes, le but de la détection prédictive est de détecter une dégradation au lieu d'une défaillance.

Diagnostic prédictif (pronostic)

L'objectif du diagnostic prédictif est d'identifier les causes et de localiser les organes qui ont entraîné une dégradation particulière.

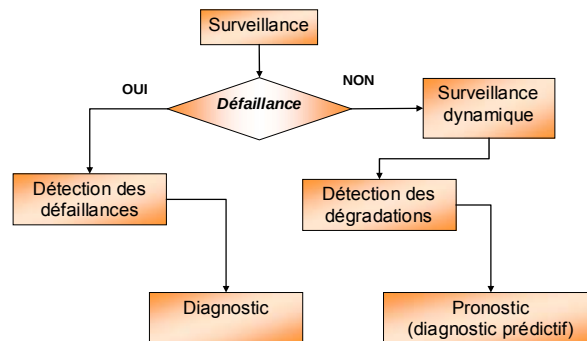


Figure 15. Surveillance classique et surveillance dynamique

Après avoir présenté les éléments de base du domaine concerné par nos contributions, nous présentons un aperçu des principales techniques liées à la surveillance industrielle, afin de dégager par la suite les idées génératrices de nos contributions scientifiques.

3.3.3 ETAT DE L'ART DES METHODES DE SURVEILLANCE. INTERET DE L'UTILISATION DES TECHNIQUES DE L'INTELLIGENCE ARTIFICIELLE

a) CLASSIFICATION DES METHODES DE SURVEILLANCE. PROBLEMATIQUE DU DIAGNOSTIC

Les méthodologies de surveillance sont généralement divisées en deux groupes : méthodologies de surveillance *avec modèle* et *sans modèle* (Dash *et al.*, 2000) :

- Les premières se basent sur l'existence d'un modèle formel de l'équipement et utilisent généralement des techniques de *l'Automatique* - (Combacau, 1991).
- Les techniques de *surveillance sans modèle* sont divisées en deux parties. La première partie correspond aux outils statistiques de *Traitement du Signal* qui sont généralement qualifiés d'outils de traitement de bas niveau, parce qu'ils sont en contact direct avec le signal capteur, et ne servent généralement que pour la génération d'alarmes brutes, sans aucune information concernant leur signification. Les outils statistiques établissent des tests sur les signaux d'acquisition, tests qui ne sont capables d'assurer que la fonction détection de défaillances. La deuxième partie est celle des techniques dites de haut niveau et qui sont plutôt orientées vers la communication avec l'expert. Celles-ci représentent les techniques de *l'Intelligence Artificielle* (IA) et servent comme outil de base pour l'aide à la décision. Leur réponse est donc plus élaborée que celle des techniques de bas niveau et elles sont capables de détecter, interpréter (association à un mode) et diagnostiquer les défaillances.

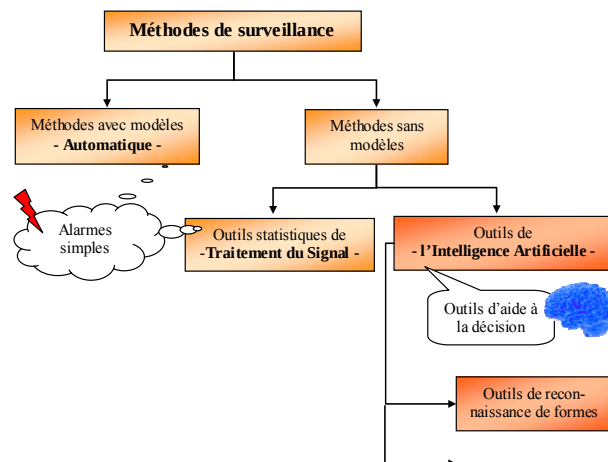


Figure 16. Classification des méthodes de surveillance - première approche

Notons que les deux approches Automatiques / Intelligence Artificielle peuvent être combinées pour profiter de certains avantages de chacune et avoir ainsi une certaine complémentarité (Dubuisson *et al.*, 2001). Les méthodes de l'Automatique sont par nature proches du système surveillé, puisqu'elles travaillent directement à partir des données issues des capteurs ; elles sont ainsi principalement utilisées pour la génération d'alarmes. Les méthodes de l'Intelligence Artificielle sont, elles, plus tournées vers la communication avec l'opérateur et se focalisent sur la transformation d'un ensemble d'informations brutes et non reliées entre elles en une information interprétable directement par l'opérateur chargé de la conduite ; elles sont donc utilisées pour l'interprétation des alarmes et l'aide à la décision (Basseville *et al.*, 1996). D'autres réflexions sur la complémentarité entre ces deux domaines peuvent être trouvées dans (Dubois *et al.*, 1994). On peut citer également quelques travaux où les deux techniques Automatique/Intelligence Artificielle ont été conjointement utilisées : (Katsillis *et al.*, 1997), (Loiez, 1997), (Washio *et al.*, 1998), (Hines *et al.*, 1995), (Vemuri *et al.*, 1997, 1998).

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

Un problème de diagnostic peut se définir comme un problème de reconnaissance de forme. L'ensemble des états est homologué à un ensemble de classes et le vecteur forme est le vecteur des composantes des paramètres observés sur le système.

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

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

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

Nous considérerons cette définition pour la suite du travail, elle semble mieux adaptée aux problèmes de diagnostic dans la mesure où elle met en évidence le raisonnement qui doit être suivi pour réaliser la fonction "comprendre" et l'approche considérée est largement reprise par d'autres auteurs, on peut citer notamment (Zwingelstein, 1995), (Grosclaude, 2001a), (Bouchon-Meunier, 2003). De plus, la décomposition du diagnostic en deux fonctions : *localisation* et *identification de la cause*, s'applique également à cette définition. Enfin, le diagnostic est réalisé à partir d'observations, ce qui revient bien à considérer la détection comme ne faisant pas partie de la phase de diagnostic.

Au regard de ces définitions, le diagnostic – de même que l'ensemble de la surveillance – traite à la fois des données numériques (exploitation des observations s'il s'agit de capteurs par exemple) et des données symboliques (connaissances sur le système considéré). Dans (Dubuisson, 1990) ces deux types de données sont considérés comme des connaissances nécessaires à l'opération de diagnostic. D'une part, on trouve la *connaissance globale*, que l'on peut qualifier de connaissance "a priori" sur le système et reposant sur le passé du système. D'autre part, une *connaissance instantanée* est nécessaire, correspondant à l'ensemble des éléments dont on dispose à un instant donné pour prendre une décision et l'exploiter.

Une autre notion se dégage également de la définition de (Peng, 1990). Elle rend compte de l'exploitation des connaissances causales sur le système. En effet, un "dysfonctionnement" peut être décrit de façon simple par des relations associant ses causes initiales (défaillances de composants, etc.) à ses manifestations observables, les symptômes. Si on dispose d'une théorie modélisant de telles relations, un problème de diagnostic consiste à rechercher à l'aide de cette théorie des *explications* satisfaisantes aux symptômes observés. L'inférence de base mise en jeu dans ce type de raisonnement qui "remonte des effets aux causes" est appelée *abductive*²⁰.

On peut la schématiser ainsi:

Etant donné le fait "B" et l'association (la relation de causalité) " $A \rightarrow B$ " ("A" cause "B"), inférer "A possible".

Le diagnostic ainsi réalisé sera qualifié de *Diagnostic Abductif*.

Un point très important pour le diagnostic est soulevé : dans quelle mesure peut-on considérer que les connaissances sur le systèmes qu'elles soient causales, globales ou instantanées sont suffisantes, et comment faut-on pour les acquérir et les valider ?

Compte tenu de la complexité de la tâche, de nombreuses méthodes existent pour réaliser un diagnostic faisant appel à des techniques variées. La partie suivante s'attache à fournir une classification des méthodes de surveillance, en prenant en compte plus particulièrement les critères associés au diagnostic.

²⁰ Les principaux types de raisonnement sont : inductif, déductif ou abductif. L'induction infère quelque chose de différent que ce qui est observé; ce raisonnement qui va des cas particuliers à la règle permet ainsi en quelque sorte de découvrir des lois ou régularités. La déduction s'inscrit dans un processus de démonstration où on conclut en utilisant des propositions prises pour prémisses, déjà démontrées ou auto-posées, d'autres propositions. Quant à l'abduction, elle est définie comme « ... une méthode pour former une prédiction générale sans assurance positive qu'elle réussira dans un cas particulier ou d'ordinaire, sa justification étant qu'elle est le seul espoir possible de régler rationnellement notre conduite future, et que l'induction fondée sur l'expérience passée nous encourage fort à espérer qu'elle réussira. » (Pierce, 1867)

b) METHODES DE SURVEILLANCE AVEC MODELE

La catégorie de méthodes la plus utilisée par les industriels à l'heure actuelle est vraisemblablement celle des méthodes par modélisation fonctionnelle et matérielle. Cette catégorie de méthodes comprend des outils industriels comme l'AMDEC²¹ et l'arbre de défaillances (ADD), utilisés essentiellement pour le diagnostic.

b.1 Les méthodes de diagnostic par modélisation fonctionnelle et matérielle

Le principe de ces méthodes consiste à établir a priori et de la manière la plus complète possible, les liens entre les causes initiales des défaillances et leurs effets mesurables. Les méthodes les plus couramment rencontrées sont l'Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité (AMDEC) et les Arbres De Défaillances (ADD).

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

b.2 Méthodes de surveillance par modélisation physique

Les méthodes de surveillance avec modèle physique ont pour principe de comparer les mesures effectuées sur le système aux informations fournies par le modèle (Frank, 1990). Tout écart est alors synonyme d'une défaillance. Les outils de la théorie de la décision sont ensuite utilisés pour déterminer si cet écart est dû à des aléas normaux, comme par exemple le bruit de mesure ou s'il traduit une défaillance du système. Les méthodes avec modèle physique sont les méthodes les plus familières aux automaticiens. En règle générale, ces méthodes peuvent être séparées en deux techniques : techniques de redondance physique et analytique, et techniques d'estimation paramétrique. Nous renvoyons le lecteur soucieux de plus de détails, aux références suivantes : (Willsky, 1976), (Isermann, 1984), (Basseville, 1988), (Basseville et al., 1993), (Gertler, 1988), (Gertler, 1998), (Patton et al., 1989), (Frank, 1990), (Combacau, 1991), (Cussenot, 1996), (Weber, 1999), (Tromp, 2000), (Combastel, 2000), (Ploix et al., 2001) ainsi que (Zemouri, 2003).

c) METHODES DE SURVEILLANCE SANS MODELE

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

c.1 Surveillance en utilisant des outils statistiques du traitement du signal

Les outils statistiques de détection de défaillances consistent à supposer que les signaux fournis par les capteurs possèdent certaines propriétés statistiques. On effectue alors quelques tests qui permettent de vérifier si ces propriétés sont présentes dans un échantillon des signaux mesurés. Une grande variété de tests, applicables sur un échantillon de mesures, peut être trouvée dans (Basseville, 1988). Parmi les plus importants, nous mentionnons : le test de franchissement de seuils, le test de moyenne et le test de variance (Zemouri, 2003).

²¹ AMDEC - Analyse des Modes, de leurs Effets et de leur Criticité

²³ RCE = Restricted Coloumb Energy

c.2 Surveillance par intelligence artificielle

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

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

Globalement, ces méthodes seront regroupées sous l'expression : "*Méthodes par modélisation symbolique*". Là encore, plusieurs approches existent dans la littérature : l'approche associatif - prédictif - explicatif (INRIA) (Basseville *et al.*, 1996) et l'approche reconnaissance - modèle opérationnel - modèle de pannes (IRISA) (Aghasaryan, 1998). Nous invitons les lecteurs intéressés par plus d'information à ce sujet, à consulter le rapport DEA (Monnin, 2004).

La classification que nous proposons regroupe les deux approches précédentes avec une définition unique pour chacun des trois types de méthodes. Nous avons également regroupé sous ces trois types de méthodes les outils de l'Intelligence Artificielle correspondants.

Etant donnée la difficulté de donner une définition claire et synthétique pour les méthodes de diagnostic, nous avons fait le choix de les caractériser par des mots clés qui rendent mieux compte de leurs objectifs et de leurs particularités respectifs. On distingue donc parmi les modèles symboliques, les méthodes à base de modèles comportementaux, les méthodes de reconnaissance de formes et les méthodes à base de modèles explicatifs.

Les méthodes à base de modèles comportementaux

Ces méthodes se caractérisent par la possibilité notamment de simuler le comportement du système, à partir d'une modélisation de son comportement. Le plus souvent il s'agit de modèles "de bon fonctionnement" qui, contrairement aux modèles numériques, ne sont pas basés sur la physique du système mais sur une considération en terme de mode de fonctionnement.

Ces méthodes regroupent notamment des outils tels que les réseaux de Petri et les automates d'états finis et seront caractérisées par les termes *modélisation/simulation*.

Les méthodes de reconnaissance de formes

Ces méthodes regroupent les modèles associatifs et les méthodes de reconnaissances dans le sens où elles sont caractérisées par les termes *apprentissage et reconnaissance* qui

s'appliquent aussi bien aux systèmes de reconnaissance de formes qu'aux systèmes à base de règles tels que les systèmes experts.

Dans ces méthodes, on retrouve principalement des outils tels que les méthodes statistiques de reconnaissance de formes, les réseaux neuronaux, la logique floue, les réseaux neuro-flous, les systèmes expert et le raisonnement à partir de cas. Les mots clés qui caractérisent le mieux ces approches sont *apprentissage/reconnaissance*.

Les méthodes à base de modèles explicatifs

Ces méthodes reprennent le terme de modèles explicatifs introduit dans (Basseville et al., 1996), et englobent de par leur définition même les notions de modèles de pannes de (Aghasaryan, 1998). Il s'agit donc de méthodes qui permettent de fournir une représentation de l'analyse causale des liens entre les défaillances, leurs causes et leurs effets observables.

Dans ces méthodes, on considèrera particulièrement des applications basées sur des outils tels que : les graphes d'influence, les graphes causaux, les graphes contextuels, les réseaux de Petri et la logique floue. Le mot clef associé est *analyse causale*.

La figure suivante regroupe notre point de vue (Monnin, 2004), (Monnin et al., 2004) sur la classification des outils de surveillance par IA (modélisation symbolique) :

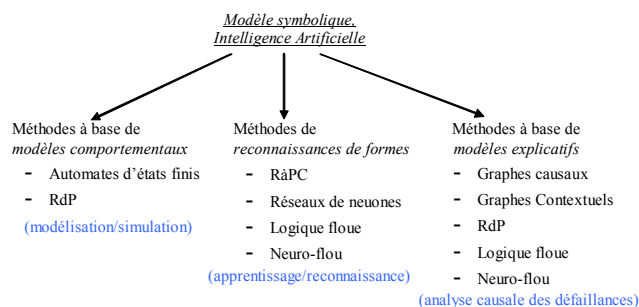


Figure 17. Classifications des outils d'IA suivant les méthodes de diagnostic.

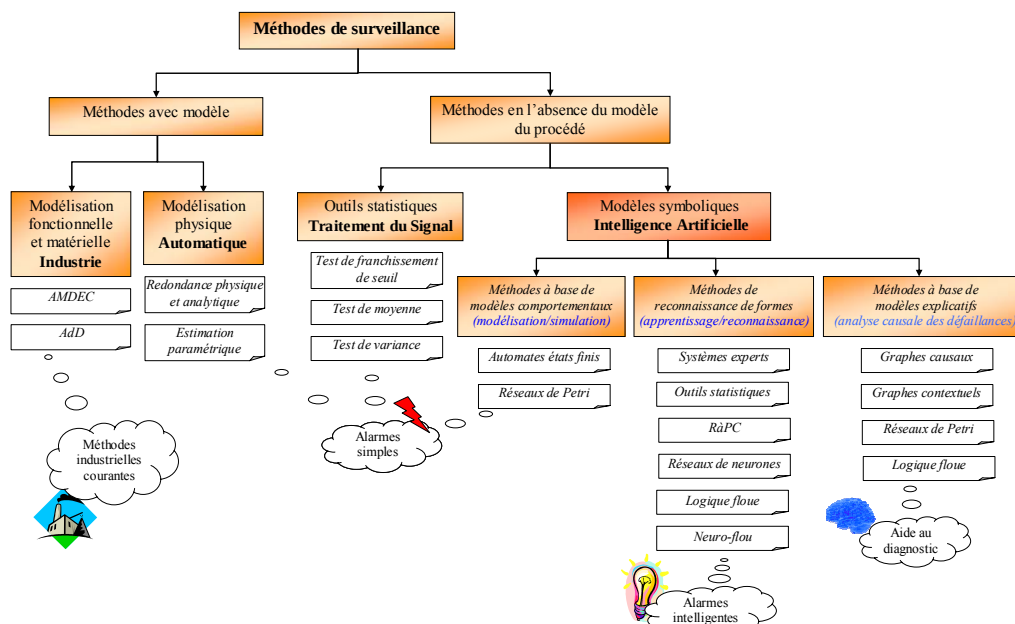


Figure 18. Classification globale des méthodes de surveillance

A la lumière de la classification proposée et au vu des précisions concernant l'importance et les spécificités du diagnostic, nous obtenons une vue d'ensemble des méthodes de surveillance, synthétisée par le schéma de la Figure 18.

Afin d'avoir une idée plus claire des avantages et inconvénients de chaque méthode, nous présentons par la suite un bref aperçu des outils utilisés dans l'approche par modélisation symbolique, en indiquant l'application en diagnostic qui en est faite. Pour chacune des méthodes, nous verrons ainsi dans quelle mesure le diagnostic réalisé satisfait la définition de (Peng, 1990), en essayant de mettre en évidence les applications, spécificités et limites des outils proposés. Nous renvoyons toutefois le lecteur soucieux de plus de détails et de références au travail de DEA de M. Monnin (**Monnin, 2004**).

Méthodes à base de modèles comportementaux

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

- Les automates d'états finis

Ils permettent de modéliser directement le fonctionnement du système, grâce à un automate global obtenu par composition d'automates élémentaires correspondant à des systèmes locaux (composants du système). Cette représentation est donc directement adaptée à la simulation et à la détection. Il existe cependant des systèmes pour lesquels cette représentation est également utilisée pour le diagnostic.

Une méthode présentée dans (Aghasaryan, 1998) et (Sampath *et al.*, 1996), se caractérise par deux étapes pour effectuer le diagnostic. Dans un premier temps, il s'agit de développer le modèle du système à l'aide des automates d'états finis, puis dans un deuxième temps, un "diagnostiqueur" correspondant également à un automate d'états finis est construit à partir du modèle global. Ce dernier effectue un diagnostic en observant en ligne une séquence d'événements. Pour chaque événement consécutif, le diagnostiqueur fournit une estimation de l'état du système et des événements non observés, d'où les occurrences des pannes sont déduites.

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

- Les réseaux de Petri

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

Dans cette approche, le diagnostic est réalisé par un tirage arrière du réseau, qui permet de remonter au marquage source (cause) à partir du marquage correspondant à l'état (défaillant) observé.

Le raisonnement de type chaînage arrière est particulièrement intéressant puisqu'il permet a priori de réaliser une réelle recherche de causes. Cependant, les contraintes liées à une application réelle ne sont pas prises en compte et la validité du diagnostic n'est pas évaluée. De plus, l'approche pour le diagnostic est complètement déterministe. Il serait donc intéressant d'apporter de la finesse au diagnostic réalisé afin d'apporter une aide au diagnostic en prenant en compte les imprécisions et les incertitudes inhérentes à la réalité industrielle.

- ***Autres formalismes***

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

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

Les automates d'états finis et les réseaux de Petri constituent donc des outils relativement bien adaptés pour construire des mécanismes de détection lorsque le fonctionnement normal du système est décrit par ces formalismes. En revanche, leurs utilisations en diagnostic sont encore limitées. Pour les automates, les principales difficultés étant liées à la taille importante de l'espace d'état, ceci conduit donc à des problèmes de mémoire et de vitesse d'exécution du diagnostic. Comme il est souligné dans (Valette *et al.*, 1994), les réseaux de Petri constituent un outil puissant de modélisation et peuvent être considérés comme un outil de description de la connaissance nécessaire au diagnostic.

Méthodes de reconnaissance de formes pour la surveillance

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

- ***Systèmes experts***

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

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

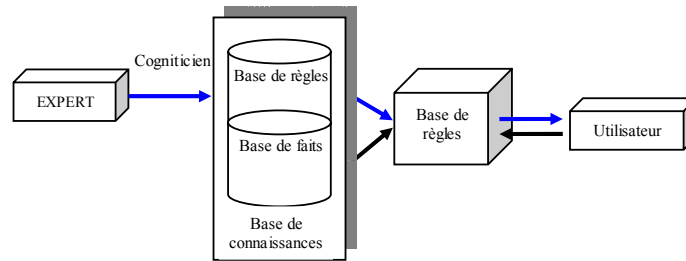


Figure 19. Structure d'un système expert

Le rôle du *cogniticien* est de soutirer leurs connaissances aux experts du domaine et de traduire ces connaissances dans un formalisme se prêtant à un traitement automatique, c'est-à-dire en règles. Ces deux tâches sont aussi délicates l'une que l'autre. En effet, un expert est la plupart du temps inconscient de la majeure partie de son savoir, et s'il arrive à en exprimer une partie, c'est souvent sous une forme difficile à formaliser.

Les systèmes experts sont subordonnés à une base de règles logiques représentant l'ensemble des combinaisons importantes d'états, sélectionnées parmi l'explosion combinatoire des états possibles du système analysé. Les systèmes experts permettent l'exploitation d'une base de données informationnelle considérable. Ils présentent le désavantage d'un comportement rigide, dans un contexte qui s'avère souvent variable et évolutif.

- Outils statistiques de reconnaissance de formes

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

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

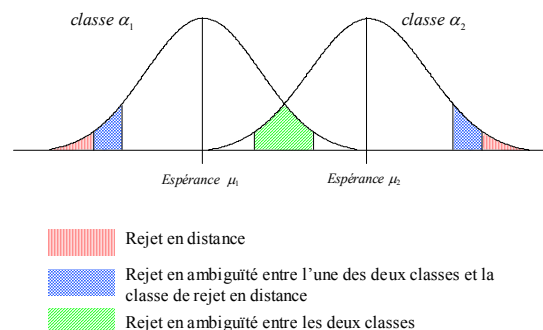


Figure 20. Représentation des différentes zones de deux distributions gaussiennes.

La qualité de la décision des classes d'affectation est quantifiée par un coût de décision (Duda, 1973). La décision peut être aussi qualifiée par un autre coût qui peut jouer un rôle particulier : c'est le coût entraîné par le fait de ne pas prendre de décision. Ce coût est appelé *rejet d'ambiguïté*. Sa valeur est égale à (-1) quelle que soit la vraie classe du vecteur forme d'entrée. La décision finale est jugée par un indicateur de performance, appelé *risque moyen de décision R*, dont on cherchera à obtenir la valeur minimale. La règle minimisant ce critère est appelée *règle de Bayes* ou *règle du risque minimum* (Fukunaga, 1990). Comme nous l'avons déjà précisé, dans le cas de la surveillance, nous préférons ne pas prendre de décision plutôt que d'associer le vecteur à une mauvaise classe ou de le rejeter en distance comme membre d'une nouvelle classe.

- **Raisonnement à partir de cas**

Le raisonnement à partir de cas - RàPC (Case Based Reasoning – CBR) est une approche récente pour résoudre et apprendre des problèmes. Il correspond à la résolution d'un nouveau problème en se rappelant une situation précédente similaire et en réutilisant les informations et les connaissances de cette situation (Aamodt, 1994). Il constitue donc une méthodologie pour modéliser le raisonnement et la pensée humaine ou pour développer des systèmes intelligents.

Le raisonnement à partir de cas recouvre un ensemble de méthodes de résolution de problèmes à partir d'expériences passées plutôt qu'à partir de connaissances générales comme le raisonnement à partir de règles, (Malek, 1996). Le principe de fonctionnement de la méthode consiste à stocker les expériences précédentes (cas) dans une mémoire afin de résoudre un nouveau problème :

- retrouver l'expérience similaire au nouveau problème dans la mémoire,
- réutiliser cette expérience dans le contexte de la nouvelle situation (complètement, partiellement ou en l'adaptant selon les différences),
- mémoriser la nouvelle expérience dans la mémoire (apprentissage).

Le RàPC constitue donc un processus cyclique pour la résolution de problèmes (Aamodt, 1994), (Malek, 1996), (Lamontagne, 2002). Dans ce processus, on identifie les phases suivantes :

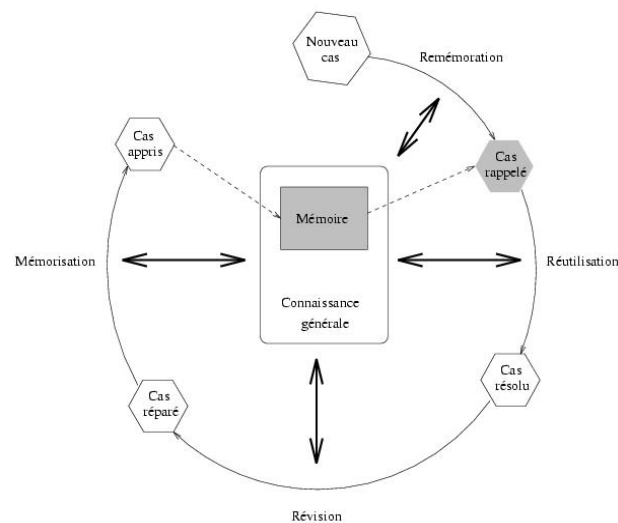


Figure 21. Cycle du raisonnement à partir de cas.

Un nouveau problème ou cas est résolu en se rappelant (*remémoration*) des cas similaires déjà pré analysés et stockés dans la mémoire. La solution trouvée est ensuite adaptée (*réutilisée*) au nouveau problème. Le nouveau cas est ensuite *révisé* ou réparé (par l'expert ou par l'utilisation de la connaissance générale du système). Ce nouveau cas peut aussi être appris au système (*mémorisation*) comme nouvelle expérience.

Le RàPC s'articule donc autour d'une base de connaissance qui contient deux parties :

- *la connaissance générale*, souvent représentée par une base de règles et qui peut intervenir dans toutes les phases du RàPC,
- *la mémoire*, qui contient les cas et les structures nécessaires qui constituent l'expérience du système; elle est utilisée lors des phases de remémoration et de mémorisation.

Un système de RàPC se caractérise donc par : la structure des cas utilisés, la structure de la base de cas (indexation des cas) et l'organisation de la mémoire.

Structure de cas

La structure des cas va dépendre des domaines d'utilisation et des tâches à accomplir. Adaptée au diagnostic, la structure des cas est donc la suivante :

<i>Problème</i>	↔ Symptômes (description de la situation particulière de diagnostic)
<i>Solution</i>	↔ Origines (plusieurs origines possibles)
<i>Conclusion</i>	↔ Actions (stratégie de maintenance)

Dans notre équipe de recherche, quelques travaux plus récents (ne faisant pas l'objet de ce rapport) ont été développés sur l'utilisation du RàPC pour le diagnostic (Rasovska *et al.*, 2005), (Morello-Chebel *et al.*, 2005).

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

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

Son utilisation en diagnostic apparaît donc comme relativement aisée, avec comme propriété, une structure de cas adaptée. Cependant, la difficulté tient justement à cette structure de cas et aux informations qu'elle doit contenir. En effet, l'extraction des connaissances et leurs représentations, sont primordiales dans ce type d'application.

- Reconnaissance de formes par réseaux neuronaux

Les réseaux de neurones sont des outils capables d'effectuer des opérations de perception, classification et prédiction. Leur fonctionnement est basé sur les principes de fonctionnement des neurones biologiques. Leur principal avantage par rapport aux autres

outils est leur capacité d'apprentissage et de généralisation de leurs connaissances à des entrées inconnues.

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

Leur utilisation est principalement guidée par leurs propriétés suivantes:

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

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

Apprentissage supervisé :

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

Apprentissage non supervisé :

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

Les architectures les plus utilisées pour ces problèmes de surveillance des systèmes industriels sont le modèle de Hopfield (basé sur la *mémoire associative*), le réseau de Kohonen (carte topologique *auto-adaptative*), le Perceptron Multicouches (PMC ou MLP pour Multi layer Perceptron) et le Réseau à Fonctions de base Radiales (RFR ou RBF pour Radial Basis Function). Une description détaillée pourra être trouvée dans la thèse de R. Zemouri (**Zemouri, 2003**).

La figure suivante montre l'architecture générale d'une application de surveillance par reconnaissance des formes avec réseaux de neurones. L'expert humain joue un rôle très important dans ce type d'application. Toute la phase d'apprentissage supervisé du réseau de neurones dépend de son analyse des modes de fonctionnement du système. Chaque mode est caractérisé par un ensemble de données recueillies sur le système. A chaque mode on associe une expertise faite par l'expert. Cette association (ensemble de données - modes de fonctionnement) sera apprise par le réseau de neurones. Après cette phase d'apprentissage, le réseau de neurones passe en phase de classification, dans laquelle il associera les classes représentant les modes de fonctionnement aux formes d'entrée caractérisées par les données du système.

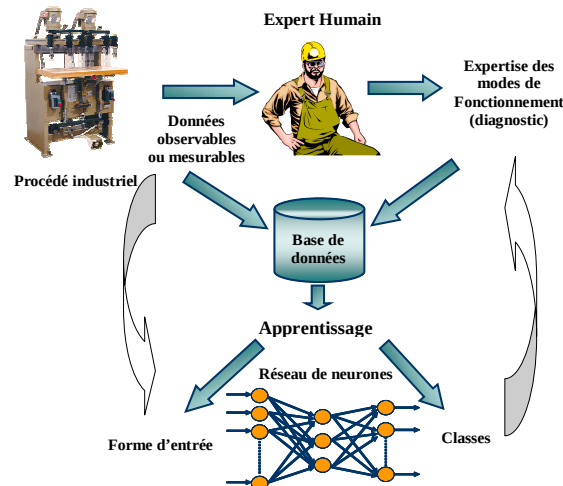


Figure 22. Reconnaissance des formes par réseau de neurones.

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

Le modèle perceptron multicouche se caractérise donc par une architecture globale (tous les neurones participent à la sortie après la phase d'apprentissage), qui lui confère de mauvaises propriétés en classification et donc le rend moins intéressant pour une utilisation en surveillance. Cependant il existe un nombre non négligeable de travaux en surveillance et diagnostic qui utilisent cette architecture, de nombreuses références sont fournies dans (Zemouri, 2003).

Celui qui semble se prêter le mieux à l'utilisation en surveillance est le réseau à base de fonctions radiales (RFR – en anglais Radial Basis Function - RBF). Ceci est dû en grande partie à son approche locale et à sa capacité de travailler dans un « monde ouvert », en enrichissant son expérience au fur et à mesure de son fonctionnement. Ce type de réseau a fait l'objet de nombreuses études de notre part. Parmi les plus importantes, nous pouvons citer les rapports DEA de N. Palluat (2002), de S. Ould Yahia (2001), J. Xu (2004), ainsi que le premier chapitre de la thèse de R. Zemouri (2003). Les algorithmes d'apprentissage les plus importants du RFR sont le RCE²³ et le DDA²⁴. Nous allons revenir sur ces outils et notamment sur les aspects temporels des réseaux de neurones dans le chapitre suivant.

- Reconnaissance de formes par logique floue

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

²⁴ DDA = Dynamic Decay Adjustment

incomplétudes, et le seul cadre dans lequel on peut traiter des connaissances numériques et des connaissances exprimées symboliquement par des qualifications du langage naturel.

Les applications de la logique floue sont extrêmement nombreuses et variées. Les plus courantes sont la commande floue, les systèmes experts flous, le raisonnement à partir de cas et la reconnaissance floue de formes. Dans le cadre de la surveillance et du diagnostic, on trouve principalement les systèmes experts, le raisonnement à partir de cas et la reconnaissance de formes. Dans ces différents contextes (aide au diagnostic, aide à la décision), l'expert humain exprime des connaissances ou des données dans un langage naturel fondamentalement imprécis; la logique floue permet donc d'une part de prendre en compte les imprécisions inhérentes aux données et d'autre part de rendre compte de l'expression des règles qui permettent de formuler un diagnostic ou de déterminer une action. On trouve par exemple dans (Rahamani *et al.*, 1998) l'architecture d'un outil de détection/diagnostic d'antennes, dans lequel la logique floue intervient sous forme d'un système expert flou et dans les étapes de classification :

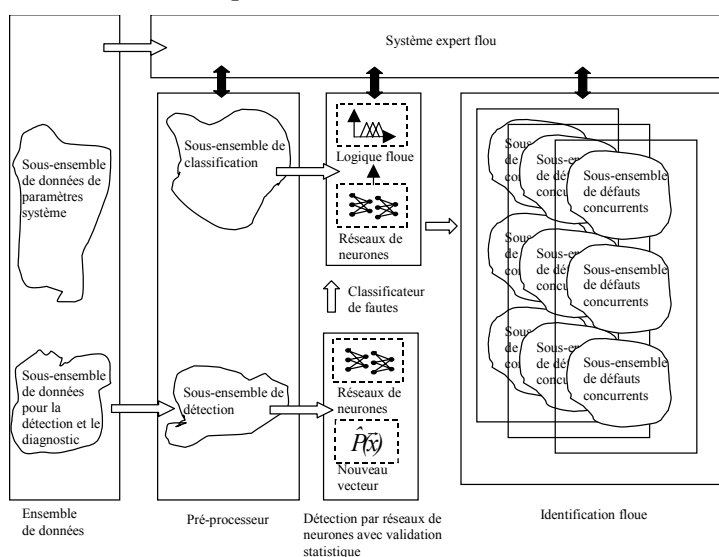


Figure 23. Exemple d'architecture de système de diagnostic par logique floue.

Dans le cadre de la surveillance des systèmes industriels, la logique floue se trouve également associée à d'autres outils et techniques d'analyses. A titre d'exemple, la logique floue a trouvé des applications en combinaison avec les arbres de défaillances afin d'évaluer l'apparition de l'événement sommet (Cheng, 2000).

Dans ces diverses applications, l'utilisation de la logique floue est assez naturelle dans la mesure où elle permet de traiter l'imprécision, l'incertitude et l'incomplétude liées aux connaissances du domaine. En plus, la logique floue leur confère une capacité d'utilisation en pronostic. Cependant, même si la logique floue fournit des résultats satisfaisants, on ne peut considérer ces applications comme de véritables applications de la logique floue pour le diagnostic dans la mesure où ces différents outils ne s'appliquent pas à la localisation et à l'identification des causes expliquant un défaut. Utilisée avec les arbres de défaillances, cette dernière devrait fournir une évaluation sur l'occurrence ou la présence des événements de base de l'arbre de défaillances qui sont eux à l'origine de l'événement sommet. On obtiendrait ainsi l'évaluation des causes à l'origine d'un dysfonctionnement.

²⁶ http://kmcenter.free.fr/technologies/IA/KMCenter_RB.pdf

- Les réseaux neuro-flous

Les réseaux neuro-flous sont nés de l'association des réseaux de neurones avec la logique floue, de manière à tirer profits des avantages de chacune de ces deux techniques. La principale propriété des réseaux neuro-flous est leur capacité à traiter dans un même outil des connaissances numériques et symboliques d'un système (Uppal, 2002). Ils permettent donc d'exploiter les capacités d'apprentissage des réseaux de neurones d'une part et les capacités de raisonnement de la logique floue d'autre part. Différentes combinaisons de ces deux techniques d'intelligence artificielle existent et mettent en avant des propriétés différentes. On peut identifier les combinaisons suivantes (**Ould Abdeslam, 2002**) :

- Réseau flou neuronal
- Système neuronal/flou simultanément
- Modèles neuro-flous coopératifs
- Modèles neuro-flous hybrides

Des structures neuro-floues pour la modélisation, la prédiction, le contrôle ou le diagnostic, peuvent être réalisées par une grande variété d'architectures, pour un même type de combinaison donné (Dourado *et al.*, 1999). On trouve par exemple dans (Lee *et al.*, 2000, 2001) et (Wang *et al.*, 2002), une utilisation d'un système neuro-flou R-SANFIS (Recurrent Self-Adaptive Neuro-Fuzzy Inference System) pour la commande d'un véhicule sous-marin autonome. Une autre utilisation des réseaux neuro-flous est présentée dans (Nauck *et al.*, 1998) où l'architecture NEFPROX (NEuro Fuzzy function apPROXimator) est utilisée pour l'approximation de fonctions.

Dans des applications de diagnostic, on trouve principalement des modèles neuro-flous hybrides, pour lesquels réseau de neurones et système flou sont combinés de manière homogène. Une définition d'un tel système neuro-flou est donnée dans (Palade *et al.*, 2002) :

Un système neuro-flou est un réseau de neurones qui est topologiquement équivalent à la structure d'un système flou. Les entrées/sorties du réseau ainsi que les poids sont des nombres réels, mais les nœuds implémentent des opérations spécifiques aux systèmes flous : fuzzyfication, opérateurs flous (conjonction, disjonction), défuzzyfication. En d'autres termes, un système neuro-flou peut être vu comme un système flou pour lequel les opérations sont implémentées de façon parallèle par un réseau de neurone.

Les systèmes neuro-flou les plus courants, sont basés sur deux types de modèles flous, les modèles TSK (Tagaki, Sugeno, Kang) et Mamdani, combinés avec des algorithmes d'apprentissage neuronaux. On trouve également le NEFCLASS (NEuro Fuzzy CLASSification) reconnu pour ces capacités de classification, (Palade *et al.*, 2002), (Uppal *et al.*, 2002), (Nauck *et al.*, 1995).

De nombreuses structures neuro-floues ont été appliquées avec succès dans de nombreux domaines en raison de leurs avantages. Parmi ces caractéristiques on peut citer :

- la conception à base de règles,
- l'application aux systèmes complexes et incertains non-linéaires,
- les capacités d'apprentissage,
- le calcul parallèle et
- la tolérance aux fautes

Cependant, l'implantation de telles structures dépend fortement des connaissances a priori sur le système et des données empiriques.

On retrouve principalement deux utilisations des réseaux neuro-flous en surveillance. Ces utilisations reposent la plupart du temps (Palade *et al.*, 2002), (Uppal *et al.*, 2002) sur l'étude des résidus qui sont générés par différence d'un signal estimé donné par un observateur neuro-flou avec les valeurs réelles du signal. Ces résidus sont ensuite classés et évalués (toujours par un système neuro-flou). Le neuro-flou intervient donc dans ces deux étapes du diagnostic, l'idée étant d'obtenir un classificateur de défauts interprétable. Pour établir formellement un diagnostic, il faut interpréter les résidus pour remonter jusqu'au défaut concerné, les résidus permettant la détection de l'état de panne. Pour ce faire, l'utilisation du classificateur neuro-flou doit permettre d'identifier sans ambiguïté le type de défaut, compte tenu de la valeur des résidus.

Les réseaux neuro-flous apparaissent comme des outils puissants combinant des grandes capacités d'approximation pour la modélisation des systèmes dynamiques non-linéaires pour lesquels le modèle mathématique est inconnu avec la possibilité d'obtenir des résultats possédant un certain niveau d'interprétation.

Plusieurs méthodes sont disponibles pour développer de tels systèmes mais quelques points négatifs sont tout de même à souligner. Tout d'abord au niveau de la conception de l'outil, il est nécessaire d'avoir des connaissances a priori suffisantes sur le système à diagnostiquer pour déterminer les variables d'entrées, les fonctions d'appartenances, ainsi que les règles. De plus les systèmes flous sont relativement gourmands en ressources, même si les réseaux de neurones viennent améliorer leurs performances, il serait intéressant d'avoir des précisions sur la possibilité d'effectuer un diagnostic en ligne.

La plupart des applications rencontrées sont basées sur l'établissement d'un diagnostic à partir de la classification de résidus, elles nécessitent donc de pouvoir établir un modèle du système. De plus, elles nécessitent un réseau par résidus ce qui rend le système relativement complexe et ne permet de diagnostiquer qu'un nombre restreint de défauts. Il serait donc intéressant d'employer ces techniques compte tenu de leurs capacités en s'affranchissant complètement d'un modèle du système à diagnostiquer.

- Conclusion sur les méthodes de reconnaissance de formes

Les applications des techniques de reconnaissance de formes pour la surveillance sont nombreuses et pour certaines, les résultats sont globalement satisfaisants. Cependant, ces méthodes effectuent - pour la plupart - une classification par reconnaissance de formes. Le diagnostic revient donc à identifier un mode de fonctionnement du processus qui reflète l'état de panne. En ce sens, la surveillance réalisée ne permet pas d'identifier formellement les causes du dysfonctionnement à moins que celles-ci soient explicitement décrites dans le mode identifié; ce que réalise le raisonnement à partir de cas. Pour les autres outils, les applications s'apparentent plus à de la "détection intelligente", pour laquelle la sortie du système de diagnostic est porteuse d'information sur l'état du système, mais n'en donne pas les causes. Ces outils semblent donc mieux adaptés à la réalisation d'un module de détection « intelligente » dans une architecture complète de surveillance.

Les méthodes à base de modèles explicatifs

Ces méthodes sont principalement basées sur la représentation des relations entre les différents états de pannes et leurs effets (éventuellement observables). Elles reposent donc sur une analyse profonde du système, de manière à avoir les connaissances suffisantes à l'expression de ses relations de cause à effet. Les modèles ainsi obtenus permettent - pour certains - une approche abductive, qui consiste à remonter aux causes des pannes à partir des observations correspondant aux symptômes. Plusieurs outils de l'Intelligence

Artificielle permettent une telle formalisation des connaissances disponibles sur un système. Il s'agit notamment des graphes causaux, des graphes contextuels, techniques auxquelles se joignent également des approches basées sur la logique floue ou les réseaux de Petri. Ces outils, par leurs capacités de modélisation et d'expression, permettent de fournir des modèles explicatifs dans le cadre d'applications au diagnostic.

- Les graphes causaux

L'exploitation de connaissances causales est assez naturelle pour le diagnostic. En effet, un "dysfonctionnement" peut être assez simplement décrit par les relations associant ses causes à ses manifestations observables.

Les graphes causaux constituent un formalisme bien adapté à la représentation de ces liens causaux. Dans une utilisation de diagnostic ils permettent d'exprimer les enchaînements causaux régissant le fonctionnement du système à surveiller en cas de panne. Il s'agit de graphes orientés acycliques. Les nœuds du graphe correspondent aux causes et effets et sont reliés par des arcs orientés (Grosclaude, 2001a).

Cet outil appartient donc aux méthodes à base de modèles explicatifs leur but étant de fournir une explication à des observations d'un fonctionnement anormal d'un système. Ils représentent des connaissances profondes sur le système capables de donner les relations causales entre les différents états de pannes (Grosclaude *et al.*, 2001b). La construction du modèle repose sur une expertise telle que des catalogues de pannes, des AMDEC ou des arbres de défaillances (Basseville *et al.*, 1996) qui expriment pour une panne l'ensemble des symptômes observables.

Leur principale utilisation en diagnostic consiste en un diagnostic abductif. Le graphe causal est utilisé de manière abductive, "*à partir des symptômes, on recherche les causes*" de manière à relier les symptômes observés à un ensemble de causes possibles (Grosclaude, 2000). Pour ce faire, plusieurs extensions des graphes causaux existent.

La première extension repose sur l'introduction de contraintes temporelles, on parle alors de "diagnostic abductif temporel" (Brusoni *et al.*, 1995). Une relation causale relie alors une occurrence de la cause à une occurrence de l'effet, des contraintes temporelles décrivent la durée de la cause, le délai entre l'occurrence de la cause et celle de l'effet, ou encore la durée de l'effet (Grosclaude, 2001a). Des algorithmes existent pour retrouver dans le graphe les causes expliquant les observations (Brusoni *et al.*, 1995), (Brusoni *et al.*, 1997).

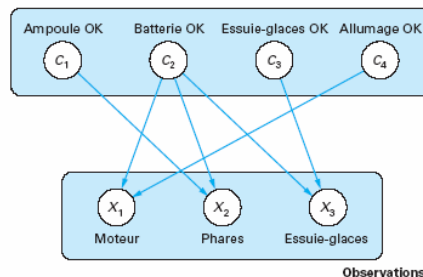


Figure 24. Exemple de réseau bayésien pour le diagnostic de voiture (Piechowiak, 2004).

Une autre approche des graphes causaux repose sur l'introduction de probabilités. On parle alors de réseaux bayésiens pour lesquels les nœuds représentent des variables aléatoires et les arcs, les dépendances causales et probabilistes entre ces variables²⁶. Un cas particulier de réseau Bayésien est représenté par les chaînes de Markov cachées.

Le graphe causal représente un outil particulièrement intéressant pour le diagnostic, dans le sens où ils peuvent apporter une justification du diagnostic proposé par le système au travers du chemin causal suivi dans le graphe. De plus, les algorithmes de diagnostic abductif permettent à partir de l'observation de symptômes de rechercher en ensemble de causes possibles qui expliquent par le biais de relations causales les observations. Enfin, l'introduction de contraintes temporelles, d'effets contradictoires et la prise en compte des interactions entre les pannes rend compte de manière plus juste de la réalité physique du système à diagnostiquer.

Les graphes causaux reposent donc sur la formalisation des liens causaux qui régissent les états de pannes et nécessite de ce fait une grande connaissance du système pour établir ces liens causaux ainsi que les contraintes temporelles. De plus, les algorithmes de diagnostic abductif temporel sont relativement complexes et imposent des longs temps de calculs. Un diagnostic en ligne est donc difficilement réalisable en utilisant directement le modèle causal.

Dans (Grosclaude, 2001a), des possibilités de pronostic sont envisagées : l'objectif étant de déterminer à partir du diagnostic de l'état courant les évolutions possibles du système. Une possibilité d'obtenir des arbres de décision à partir du modèle causal est également étudiée afin d'orienter le diagnostic au fur et à mesure de l'apparition des symptômes. Dans (Grosclaude *et al.*, 2001b), une heuristique est introduite, de manière à réduire le temps de calcul.

Plusieurs limitations voient le jour lorsque l'on choisit d'implémenter une approche de type graphe causal et plus particulièrement de type réseau Bayésien :

- L'approche Bayésienne se fonde sur 3 postulats qui ne sont pas réalistes :
 - o hypothèses mutuellement exclusives,
 - o exhaustivité
 - o indépendance conditionnelle de l'observation dans les hypothèses.
- Elle exige de pouvoir estimer toutes les probabilités a priori,
- Les contraintes sur les valeurs des probabilités (en particulier, la somme doit être égale à 1) rendent difficile l'évolution de la base de connaissance,
- La propagation des probabilités est coûteuse en temps,
- Ne sait pas gérer l'ignorance.

- Les graphes contextuels

Dans un contexte d'aide à la décision, les outils créés doivent assister un opérateur dans ses activités quotidiennes, sans toutefois prendre de décisions à sa place. Pour ce faire, l'analyse de l'activité des opérateurs permet de mettre en évidence la structure des raisonnements suivis ainsi que l'organisation des sous tâches et des actions. Cette structure peut être modélisée par des graphes. Les graphes ainsi créés donnent les différentes méthodes pour atteindre un but et ces méthodes se distinguent par le contexte dans lequel elles s'appliquent ; on parle alors de graphes contextuels (Brézillon *et al.*, 2001, 2003). Il s'agit en effet d'un formalisme bien adapté, car basé sur une expression du raisonnement que suivent les opérateurs en pratique.

Les graphes contextuels ont été introduits à partir des arbres de décision pour lesquels on passe d'une représentation basée sur le contexte de l'incident à une représentation basée sur le contexte de résolution de l'incident. Les branches de l'arbre qui conduisent à la même action terminale sont rassemblées dans le graphe et un branchement temporel est

introduit dans le graphe pour rendre compte des actions et des décisions qui peuvent être réalisées en parallèle.

Le graphe contextuel est constitué d'*actions élémentaires*, *activités*, *nœuds contextuels*, *nœuds de recombinaisons* et *branchements temporels*. Ces éléments conduisent à la définition suivante du *graphe contextuel* :

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

Remarque :

Le fait que le graphe soit acyclique garantit la finitude de l'algorithme de parcours du graphe.

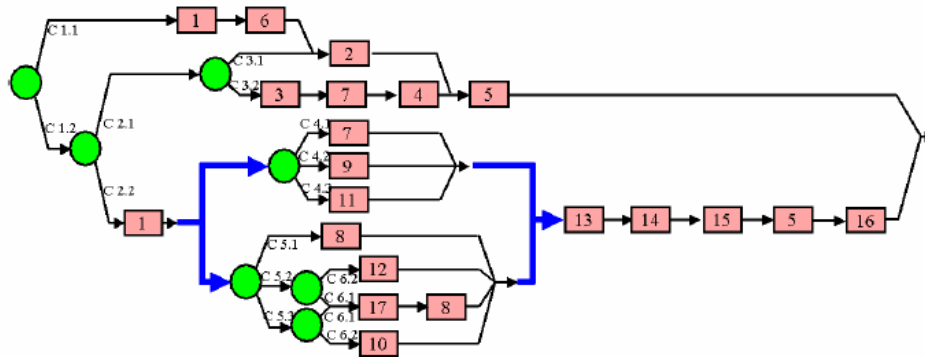


Figure 25. Exemple de graphe contextuel

Un graphe contextuel représente ainsi un raisonnement contextuel local construisant une stratégie pour la réalisation d'un but en fonction de la situation et de son évolution. En effet les graphes contextuels sont conçus pour évoluer au cours de leur utilisation.

Les graphes contextuels permettent la représentation d'actions multiples dépendant du contexte. De plus, cette représentation prend en compte la dynamique du contexte dans son évolution. Ils présentent également l'avantage de pouvoir traiter des grandes structures telles que les applications industrielles. Leur représentation est compréhensible par des opérateurs puisqu'elle est similaire à leur mode de raisonnement. Enfin, leur flexibilité et leur modularité permettent l'acquisition incrémentale de connaissance de manière à intégrer des nouvelles pratiques.

Les graphes contextuels apparaissent donc comme un outil adapté pour la modélisation d'activités comportant une dualité procédure/pratique. Ils sont donc applicables dans des domaines où une interprétation ou une adaptation de règles générales est nécessaire pour prendre en compte la richesse du contexte réel d'application. Pour le diagnostic, ils s'appliqueraient dans des domaines où les causes des défaillances d'un système seraient fortement liées au contexte d'apparition de la panne.

Dans le cadre d'une application de supervision, ils pourraient s'appliquer dans des cas où le contexte prend une place importante dans le lien entre diagnostic de défauts et les actions de reprise.

- Les réseaux de Petri

Dans le cadre du diagnostic, les réseaux de Petri permettent également une approche en terme de *modèle de pannes*. Les places constituent alors des états de pannes et l'architecture du réseau permet de rendre compte des relations existant entre ces pannes. Plusieurs techniques se basant sur un modèle de pannes ont été développées, avec en particulier, l'utilisation de réseaux de Petri stochastiques (SPN) ou partiellement stochastiques (PSPN). Ceci permet une bonne capacité d'expression du non déterminisme.

Dans (Aghasaryan *et al.*, 1997), (Aghasaryan, 1998), (Tromp, 2000), (Fabre, 2001), une approche par réseaux de Petri aux problèmes de détection de pannes et de diagnostic est proposée. Dans cette approche, le diagnostic est présenté de la manière suivante :

Les alarmes générées par le processus de détection souvent n'identifient pas les pannes et ne déterminent pas exactement leur localisation. Le problème de diagnostic consiste donc à trouver une explication à la présence de symptômes en utilisant les connaissances du domaine (Aghasaryan, 1998).

Les réseaux de Petri sont un modèle puissant pour l'expression de la concurrence ; des événements concurrents pouvant apparaître sur l'axe du temps dans n'importe quel ordre, ou simultanément. Dans un RdP, les événements sont seulement partiellement ordonnés, ce qui définit une *sémantique d'ordre partiel*. Cette sémantique d'ordre partiel est particulièrement bien adaptée pour les environnements répartis. De plus, l'approche est fondée sur une description explicite de la propagation des alarmes et des défauts en utilisant les réseaux de Petri saufs. Cette approche peut donc être vue comme appartenant aux méthodes à base de modèles explicatifs (Aghasaryan, 1998).

Dans (Tromp, 2000), l'approche pour la construction du modèle est un peu différente, les réseaux de Petri utilisés pour le diagnostic sont établis à partir des arbres de défaillances du système considéré. De plus, l'approche globale est hybride puisqu'elle utilise aussi les résidus issus d'un modèle continu du système.

Le réseau de Petri permet de modéliser les différentes dépendances entre les défauts :

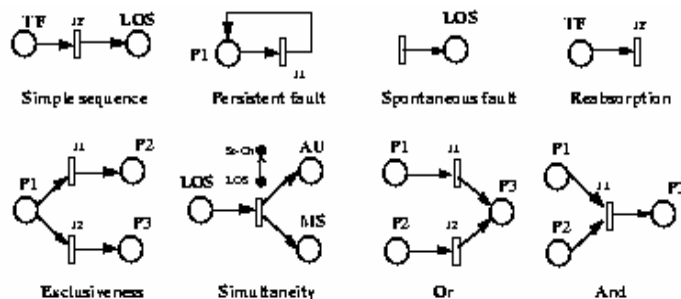


Figure 26. Dépendances entre les défauts

Dans l'utilisation qui est faite, les alarmes sont associées aux transitions, les places qui correspondent aux états de pannes ont une capacité de 1, avec une fonction d'étiquetage. Ces descriptions élémentaires sont la base pour la construction d'un « réseau de défauts », pour lequel les places sont bornées à un jeton et les arcs ont un poids de 1.

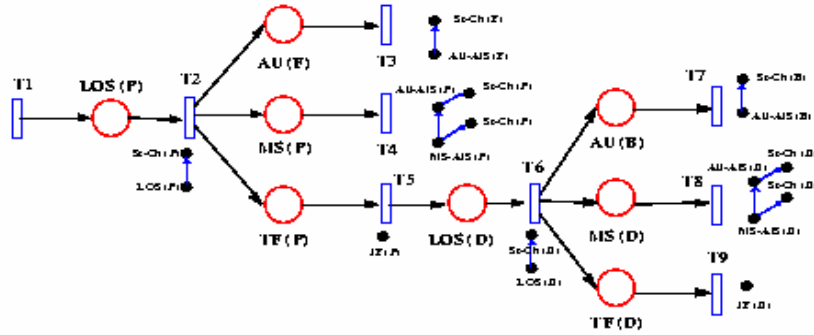


Figure 27. Réseau de défauts (fault net)

Un franchissement exprime une propagation élémentaire de panne et est accompagné par l'observation d'une alarme choisie dans une étiquette.

(Tromp, 2000) propose également une construction des RdP à partir des arbres de défaillances :

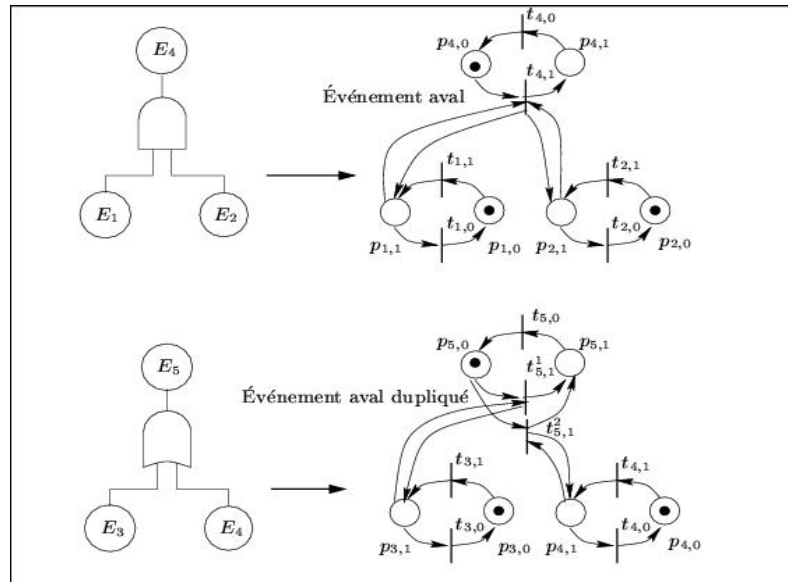


Figure 28. Conversion des portes logiques de l'arbre de défaillance.

Le réseau de Petri définit donc une relation de causalité basée sur le passage de ressource dans une séquence de franchissement. Cette relation de causalité définit un graphe orienté acyclique (graphe de causalité). La sémantique d'ordre partiel du RdP est obtenue à partir de ce graphe de causalité (GdC). La trajectoire du RdP est un ordre partiel obtenu par fermeture transitive de la relation de causalité à partir du graphe de causalité.

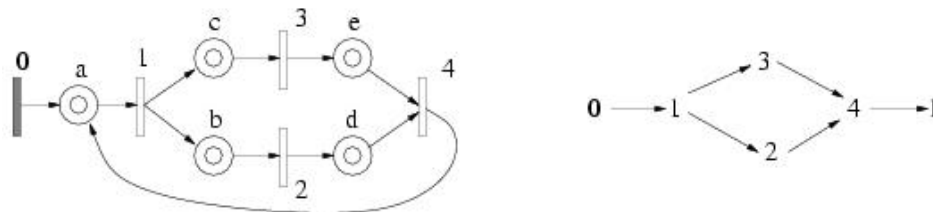


Figure 29. RdP et graphe de causalité

Pour représenter une trajectoire du réseau partiellement stochastique et pour calculer sa vraisemblance, une technique de dépliage du temps, est utilisée. Sur ce réseau déplié, le GdC est obtenu en ne gardant que les connexions qui correspondent aux transitions tirées.

En conclusion, l'utilisation des RdP pour le diagnostic offre plusieurs aspects intéressants. Tout d'abord, les travaux ont été étendus au diagnostic décentralisé. Dans ce cas, les solutions locales sont vues comme des alarmes intelligentes puis mises ensembles au niveau d'un superviseur central. L'outil développé permet d'exprimer la concurrence, la causalité et le non déterminisme, trois notions essentielles pour le diagnostic.

L'utilisation d'un modèle de réseaux de Petri partiellement stochastique permet de conserver l'indépendance probabiliste des événements concurrents et ne nécessite pas l'exploration de tout l'espace d'état car seul les contextes locaux sont nécessaires à la propagation des événements ou à l'estimation de vraisemblance. De plus, cet aspect de contexte local rend la méthode plus robuste aux changements de configuration du réseau.

En revanche, les aspects de l'approche locale posent des problèmes quant aux résultats dans (Tromp, 2000) et conduisent à des résultats expérimentaux qui ne sont pas satisfaisants

L'approche de (Tromp, 2000) semble bien adaptée pour le diagnostic des systèmes industriels, car la modélisation est réalisée à partir d'un arbre de défaillances. Par contre, les traitements sont réalisés en utilisant une approche hybride qui fait appel à des résidus et conduit à une complexité diminuant les performances du diagnostic.

- *La logique Floue*

Une autre utilisation de la logique floue en diagnostic est décrite dans (Bouchon-Meunier *et al.*, 2003). Il s'agit ici d'un *diagnostic orienté explication* qui permet d'expliquer la présence d'un ensemble de symptômes et de remonter aux causes à l'origine de ces observations. Pour ce faire, on utilise la modélisation d'une règle graduelle (*R*) de causes à effets entre les dysfonctionnements et les symptômes. De plus, on prend également en compte l'incertitude (ou le caractère vague) de l'observation de certains symptômes.

Selon cette approche, le diagnostic est réalisé grâce à la confrontation entre les connaissances et les observations. Ce diagnostic est réalisé en deux étapes : il utilise un *indice de cohérence* raffiné ensuite par un *indice de pertinence*.

Le *diagnostic par cohérence* permet de rejeter tous les dysfonctionnements incohérents avec les observations. Ce premier principe se base sur le modus tollens qui s'énonce comme suit :

si $d \Rightarrow s$ et $\neg s$ alors $\neg d$,

De cette manière, seuls les dysfonctionnements pour lesquels les observations ne contredisent aucun de leurs symptômes sont gardés. Le diagnostic par cohérence est permis par un *indice de cohérence (coh)* qui calcule la possibilité d'un événement flou au sens du filtrage flou. En utilisant l'indice de cohérence, un dysfonctionnement sera d'autant plus rejeté que son indice de cohérence sera petit. Ensuite, pour orienter le diagnostic vers les dysfonctionnements les plus plausibles, on met en œuvre un *test abductif*.

Le *diagnostic abductif* permet la sélection des dysfonctionnements qui expliquent effectivement les observations. Elle peut se formuler de la façon suivante :

sachant $d \Rightarrow S$,

Ayant observé l'ensemble de symptômes S , on conclut que d est une explication pertinente des observations. Ainsi, parmi les dysfonctionnements sélectionnés par la cohérence, l'abduction met l'accent sur les seuls dysfonctionnements qui sont pertinents. Pour ce faire, on utilise un *indice flou d'inclusion (per)*

Le diagnostic est alors établi de la manière suivante : *coh* permet tout d'abord de rejeter les dysfonctionnements incohérents avec les observations. Puis *per* vient affiner le diagnostic afin de sélectionner et de classer par ordre de suspicion les dysfonctionnements restants.

- Conclusion sur les méthodes à base de modèles explicatifs

A travers les différents exemples décrits, la logique floue apparaît comme un outil puissant pour traiter de l'imprécision, de l'incertitude et du raisonnement. Les possibilités qu'elle offre de traiter à la fois des données symboliques et numériques en font un outil tout indiqué pour le diagnostic. En effet, le diagnostic nécessite de traiter avec des connaissances et des raisonnements humains exprimés de ce fait sous forme linguistique. L'utilisation de ces connaissances à travers les raisonnements doit se faire de façon qualitative mais aussi quantitative pour fournir une aide au diagnostic pertinente et efficace.

D'autres travaux sur la logique floue s'intéressent également à la réalisation d'un raisonnement abductif par l'inversion du modus ponens. Etant donné une règle floue du type "*si U est A alors V est B* ", et une observation de type " *V est B'* ", les auteurs cherchent à caractériser les hypothèses de type " *U est A'* " répondant à la question "*pourquoi V est-il B'* "? Cette approche abductive basée sur l'inversion du modus ponens généralisé doit permettre de retrouver les hypothèses satisfaisant une observation donnée. Si la règle de départ modélise une relation causale pour laquelle la prémisse A est une cause et la conclusion B est un effet, l'approche abductive partant de l'observation B' de l'effet permet de caractériser la cause par l'hypothèse A' et réalise donc un diagnostic, (Mellouli *et al.*, 2000a,b, 2003).

d) CARACTERISTIQUES FONDAMENTALES D'UN OUTIL DE SURVEILLANCE ET D'AIDE AU DIAGNOSTIC

Suite à la présentation succincte des différents outils de l'intelligence artificielle appliqués à la surveillance, nous allons revenir sur quatre points essentiels énoncés dans (Basseville *et al.*, 1996), (Monnin, 2004) pour évaluer et comparer les différentes solutions proposées :

- 1) difficultés liées à l'acquisition des informations / du modèle,
- 2) capacité à prendre en compte l'incertain et l'imprécision,
- 3) généricité des outils et capacité à évoluer avec le système,
- 4) validation (ou évaluation) des résultats obtenus.

En effet, ces quatre points clés reviennent dans quasiment tous les outils présentés et permettent de mettre en avant les avantages et les inconvénients des différentes méthodes de diagnostic rencontrées. Nous allons donc faire le point sur chacune de ces caractéristiques.

d.1 Acquisition de l'information / du modèle

Le diagnostic tel qu'il est défini dans (Peng *et al.*, 1990), définition sur laquelle nous nous sommes appuyés, nécessite des connaissances sur les relations liant symptômes et pannes. Nous avons présenté les principaux outils et méthodes qui permettent de réaliser un diagnostic au plus proche de cette définition. Les problèmes liés à l'acquisition des modèles sont donc communs à toutes les méthodes de diagnostic rencontrées.

Plus particulièrement, pour les méthodes à base de modèles comportementaux, les modèles décrivent le système dans son fonctionnement normal (avec éventuellement des états de panne) qui est donc le mieux connu, ce qui rend l'acquisition du modèle un peu moins délicate et notamment pour la complétude du modèle. En revanche, la pertinence du modèle dépend fortement de son niveau d'abstraction.

Pour les méthodes de reconnaissance telles que celles basées sur les réseaux de neurones, la logique floue ou le neuro-flou, la définition du vecteur forme est un point crucial qui nécessite des connaissances suffisantes pour définir les éléments les plus pertinents qui constitueront les composantes de ce vecteur. De plus, pour les structures à couches comme les réseaux de neurones ou le neuro-flou, le nombre d'éléments par couche entre également dans les difficultés rencontrées lors de l'acquisition du modèle. Pour ces méthodes, du modèle va également dépendre l'interprétabilité des résultats de diagnostic. Par exemple, pour un système de RàPC, dans la structure des cas, la définition des symptômes et des origines est fortement liée au niveau d'analyse auquel on se place pour les définir²⁷ et de cette définition va également dépendre l'efficacité du système. En effet, la définition des cas est primordiale et conditionne complètement le système, elle nécessite donc une très bonne expertise du système et de ces dysfonctionnements. En revanche, un point fort pour toutes ces méthodes est la notion d'apprentissage qui permet de garantir une meilleure complétude du système puisque les cas non rencontrés peuvent être appris au cours de l'utilisation.

Les méthodes à base de modèles explicatifs reposent sur l'expression des relations causes-effets entre les dysfonctionnements, les pannes et leurs effets observables. Le diagnostic réalisé par le modèle dépend donc directement de l'expression de ces relations. Un des problèmes sera donc de trouver le bon formalisme pour traduire de façon efficace et exploitable la connaissance de ces relations. Cependant, il apparaît que même si le modèle dépend uniquement des connaissances de son comportement en cas de dysfonctionnement, des outils tels que l'AMDEC, les arbres de défaillances ou les catalogues de pannes facilitent son acquisition et garantissent une certaine complétude. Le problème se complique en revanche lorsqu'on y ajoute des contraintes temporelles qui elles, ne sont pas fréquemment explicitées dans les outils cités plus haut.

L'extraction de la connaissance disponible sur un système et sa formalisation constitue donc une étape très importante dans la conception d'un système de diagnostic. De cette étape va dépendre l'efficacité du système et, surtout, son interprétabilité.

²⁷ Prenons par exemple une pièce dans laquelle la lumière ne s'allume plus, cette constatation sera donc le symptôme et une origine serait l'ampoule ne fonctionne plus. Si l'ampoule est réparable, le symptôme sera : ampoule ne fonctionne plus et une origine pourrait être : filament cassé. On met ainsi en évidence la notion de point de vue dans l'analyse pour l'acquisition du modèle

²⁹ On rencontre souvent en littérature les termes classification ou discrimination associés au domaine de la reconnaissance des formes.

d.2 Prise en compte de l'incertain

Dans les activités de surveillance et de diagnostic, les imprécisions et les incertitudes interviennent à différents niveaux. Un système de diagnostic qui se veut efficace se doit donc de les prendre en compte. Nous avons vu que plusieurs solutions sont proposées dans la littérature. Les réseaux neuronaux par exemple – de par leur nature même – permettent de traiter des données bruitées ou incomplètes. De plus, la plupart des réseaux rendent également compte des incertitudes par la valeur de la sortie du réseau qui correspond à une valeur qui peut être comprise entre 0 et 1 en fonction de l'appartenance du vecteur d'entrée à une classe. Avec les réseaux de Petri partiellement stochastiques, on trouve des algorithmes qui permettent d'associer une vraisemblance à une trajectoire du réseau et constituent donc un modèle probabiliste. Associées aux graphes causaux, les probabilités constituent les réseaux Bayésiens, ce qui permet la prise en compte des incertitudes. Cependant, comme pour les réseaux de Petri stochastiques, les algorithmes associés aux réseaux Bayésiens sont souvent coûteux en terme de ressources et donc difficilement applicables au diagnostic.

La logique floue tient une place un peu particulière en ce qui concerne la prise en compte de l'incertain. En effet, en plus des systèmes basés uniquement sur la logique floue comme la reconnaissance floue de forme, il est fréquent qu'elle soit associée à d'autres outils pour introduire la prise en compte des incertitudes et des imprécisions sur le système à diagnostiquer. On trouve par exemple les réseaux neuro-flous pour lesquels elle fournit une meilleure interprétabilité des résultats tout comme pour les systèmes d'aide à la décision utilisant la logique floue. Elle est également introduite en extension des réseaux de Petri pour la prise en compte de l'incertitude et, enfin, dans les méthodes à base de modèle explicatifs, elle permet d'introduire des niveaux de certitude sur les liens entre des dysfonctionnements et leurs symptômes associés introduisant ainsi l'imprécision sur les données au cœur même du raisonnement à la base du diagnostic. Elle apparaît donc comme indispensable à tout système de diagnostic.

d.3 Généricité des outils

Comme nous l'avons vu, l'acquisition du modèle est une tâche complexe, de plus le diagnostic nécessite des connaissances qui sont le plus souvent propres aux systèmes considérés et nécessite donc une bonne expertise du système en particulier pour les méthodes de reconnaissance. En effet, pour ces méthodes, il est difficile de mesurer l'impact d'une modification du système sur l'ensemble de l'expertise. Ce problème se retrouve également pour les méthodes à base de modèles explicatifs. A plusieurs reprises, les réseaux de Petri sont mis en avant pour leur capacité de modélisation. Ils permettent par exemple d'obtenir un modèle global du système par composition de modèles élémentaires et limitent ainsi les répercussions d'une modification du système. Ils permettent également des modélisations qui s'appuient sur des expertises déjà réalisées comme les arbres de défaillances par exemple. Cependant, les connaissances nécessaires à la construction de tout modèle pour le diagnostic sont propres aux systèmes à diagnostiquer, il semble donc plus pertinent de fournir une méthode générique pour la construction du modèle. Cette méthode pouvant utiliser les connaissances sur le système pour la construction du modèle et ce quel qu'il soit. La généricité des outils passe donc plus par leur exploitation pour l'obtention d'un modèle que par le modèle construit lui-même.

d.4 Validation des résultats

Le diagnostic doit fournir une explication aux différents états de panne du système, une première solution consiste donc à confronter le diagnostic avec un retour d'expérience s'il est disponible. Se pose alors le problème des pannes éventuelles qui ne sont encore jamais survenues sur le système. En effet, il peut être dangereux de fournir une explication dans tous les cas si les pannes sont encore inconnues dans l'historique du système et que les connaissances liées à ces pannes sont trop incomplètes. La confrontation des diagnostics à ceux émis par les experts et les opérateurs de conduite est une des solutions les plus couramment employées (Basseville *et al.*, 1996).

3.4 CONCLUSION ET SYNTHESE

Comme nous l'avons montré dans les paragraphes précédents, la tendance industrielle actuelle en matière d'automatismes et d'équipements industriels nous mène vers des systèmes mécatroniques de plus en plus complexes, travaillant dans un environnement incertain, évolutif, correspondant à une recherche permanente d'amélioration, d'optimisation et de productivité.

Dans ce contexte, l'utilisation des techniques de surveillance avec modèles semble de plus en plus difficile et coûteuse, en offrant des performances de moins en moins satisfaisantes. En plus, la surveillance à base de modèle du système est souvent opérée hors ligne, empêchant ainsi des traitements en temps réel de l'information, une exigence de plus en plus présente chez les industriels.

En revanche, l'Intelligence Artificielle offre des outils totalement découplés de la structure du système, ne nécessitant pas la modélisation préalable de ce dernier et permettant un suivi temps réel de son évolution. Par ailleurs, le raisonnement en ligne fait que l'approche de l'Intelligence Artificielle est plus robuste à des changements de modes opératoires, comme pour les systèmes ayant plusieurs configurations ou étant obligés de changer régulièrement de configuration. Cette approche s'avère par conséquent évolutive.

L'approche de surveillance par IA, munie des notions de rejet (d'ambiguïté et de distance) et de la possibilité d'adaptation, s'est montrée très performante pour résoudre des problèmes de surveillance industrielle. Les systèmes de surveillance utilisant des techniques de l'Intelligence Artificielle peuvent donc représenter d'excellents systèmes d'aide à la décision pour l'expert humain. Ceci étant, un des aspects fondamentaux de cette approche de surveillance nous semble assez peu abordé dans la littérature : il s'agit de l'aspect diagnostic, avec une de ses fonctions fondamentales : la recherche de la cause d'une défaillance. C'est par conséquent sur cette problématique faisant partie de la maintenance proactive – tendance marquée dans l'évolution de la maintenance des années à venir –, que nous avons souhaité apporter une partie de notre contribution.

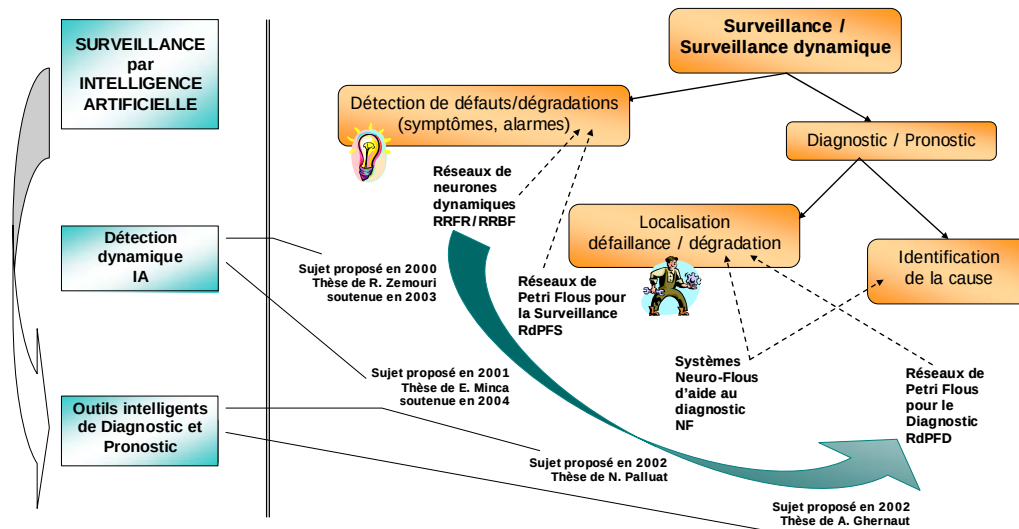


Figure 30. Thématiques de recherches abordées. Positionnement des travaux de recherche co-dirigés par rapport aux problématiques de la surveillance

En effet, dans la plupart des techniques de surveillance par reconnaissances de formes, on recherche des cas « similaires » aux nouveaux cas rencontrés lors de la surveillance « temps réel », afin de retrouver par rapport aux cas mémorisés, l'ensemble de son historique et d'en profiter pour en extraire des éléments d'aide pour le nouveau cas (diagnostic probable, inspiré des résultats du diagnostic précédent). Cette approche (correspondant au principe du raisonnement à base de cas) se prête à une base de données complète, presque exhaustive, dans laquelle en plus, une correspondance bijective est possible entre les symptômes observés et l'origine de la défaillance. Cette configuration existe rarement dans les applications industrielles réelles. Mise à part des informations souvent incertaines ou incomplètes avec lesquelles les responsables de maintenance sont obligés de travailler, le diagnostic industriel d'un système mécatronique complexe est souvent un tâche très difficile, dans laquelle on ne peut tout simplement « copier » les expériences du passé, telles qu'elles sont déjà apparues. Cette remarque est d'autant plus évidente lorsque l'historique du système est très faible ou inexistant, dans le cas – par exemple – d'un équipement nouveau ou peu connu.

Les arguments précédents font que nous avons orienté naturellement nos recherches vers les techniques de surveillance par modélisation symbolique / intelligence artificielle en abordant deux aspects fondamentaux :

- La détection dynamique des défaillances / dégradations et
- L'aide au diagnostic / pronostic

Les techniques de détection dynamique de dégradations/défaillances font appel à des méthodes de reconnaissances de formes, que nous avons développées pour le traitement dynamique des informations. Dans ce cadre, nous proposons des réseaux de neurones dynamiques d'un type nouveau, permettant une approche locale spécifique à la surveillance et une prise en compte de l'aspect dynamique des informations.

En ce qui concerne l'aide au diagnostic et au pronostic, le rôle central – de notre point de vue – est joué par la recherche de la cause de la défaillance ou de la dégradation. Ce volet de la surveillance ne peut pas être traité par les techniques de reconnaissances de formes et nécessite l'utilisation de modèles explicatifs.

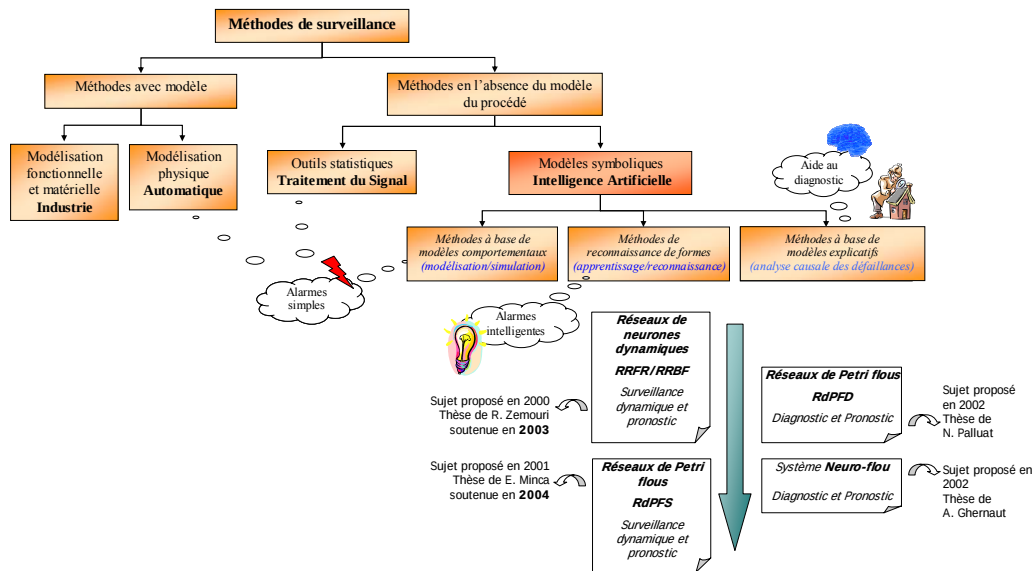


Figure 31. Positionnement de travaux co-encadrés par rapport aux différents outils de l'Intelligence Artificielle

Dans ce sens, nous nous sommes intéressés à la modélisation de la propagation des défauts et à la démarche de recherche de cause à l'aide des réseaux de Petri flous, afin de développer et tester par la suite des outils neuro-flous. Ces derniers sont capables (grâce à la logique floue) de prendre en compte aussi bien l'expérience humaine et l'incertain sous la forme de règles de propagation de défauts ou règles de recherche de cause, ainsi que de prendre en compte (grâce à sa composante neuronale) la notion d'apprentissage et l'adaptation à un contexte de surveillance évolutif.

Dans le chapitre suivant, nous commençons par présenter nos premières recherches, concernant l'étude d'un réseau de neurones pour la surveillance dynamique.

Chapitre 4 - Surveillance Dynamique par Réseaux de Neurones Récurrents

4.1 INTRODUCTION

Les réseaux de neurones peuvent fournir une solution intéressante pour des problématiques de surveillance d'équipements industriels. En effet, leur utilisation ne nécessite pas l'existence d'une modélisation formelle de cet équipement. Par ailleurs, leurs capacités de mémorisation, d'apprentissage, d'adaptation et le parallélisme du calcul représentent des fonctions très utiles à tout système de surveillance autonome.

Les techniques de surveillance par réseaux de neurones sont fondées sur l'existence d'une base de données d'apprentissage et non sur l'existence d'un modèle formel ou fonctionnel de l'équipement. Le principe d'une telle application est de trouver une relation entre une (des) variable(s) d'entrée et une (des) variable(s) de sortie. Les variables d'entrée peuvent être quantifiables (sorties capteurs) ou qualifiables (observations faites par l'opérateur). A partir de ces variables d'entrée, le réseau de neurones donne une réponse caractérisée par deux types de variables de sortie. Des variables de sortie réelles qui peuvent représenter une sortie estimée d'un paramètre de surveillance, ou des variables de sortie catégorielles qui représentent l'état de fonctionnement de l'équipement.

Selon la nature des données en sortie, il existe deux types d'applications. Le premier type est une application *d'approximation de fonctions*, qui consiste à estimer une sortie mesurée de l'équipement. Dans ce cas, les réseaux de neurones sont utilisés en tant qu'approximateur universel et fournissent un modèle sous la forme d'une boîte noire du système. Ceci n'est autre que de l'identification des processus industriels. Comme pour les méthodes à base de modèle, décrites au chapitre précédent, la comparaison de la sortie du réseau de neurones avec celle du système réel donne un résidu qui servira à déterminer si le système est dans un état défaillant ou pas. Le deuxième type d'application considère le problème de la surveillance comme un problème de *reconnaissance des formes*²⁹. La forme à reconnaître est caractérisée par l'ensemble des données (quantifiables et – ou – qualifiables) et les classes d'appartenance représentant les différents modes (de fonctionnement ou de dysfonctionnement). Le réseau de neurones doit nous fournir une réponse qui nous renseigne sur l'état de fonctionnement de l'équipement. Il assure la fonction de *détection* (fonctionnement normal ou pas), sous la forme d'une *détection intelligente* (reconnaître un mode de défaillance) qui ne représente pas – selon les définitions que nous avons retenues au chapitre précédent – à proprement parler un *diagnostic* mais qui peut faciliter sa mise en œuvre ultérieure, en précisant la classe de symptômes.

La figure ci-dessous illustre les deux types d'applications. Les avantages les plus importants que l'on peut donc attribuer à une application de surveillance par réseaux de neurones sont :

- La modélisation et l'estimation de fonctions non linéaires par apprentissage
- La fusion de données et le parallélisme

- La généralisation et reconstruction des signaux capteurs

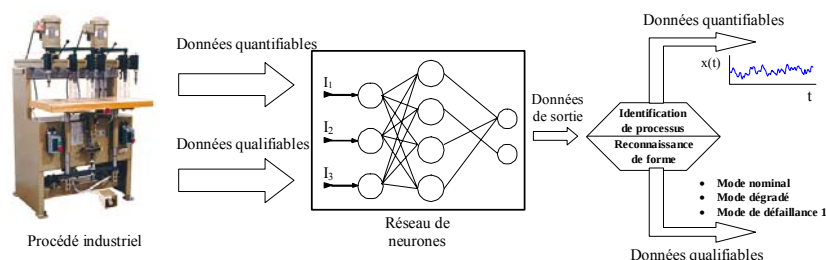


Figure 32. Application des réseaux de neurones en surveillance.

En prenant en considération les caractéristiques souhaitables d'un système de surveillance dynamique, nous avons noté dans le chapitre précédent que la flexibilité occupe une place importante. Dans cet objectif, parmi les structures classiques de réseaux de neurones statiques, le réseau de neurones à fonctions de base radiales RFR (RBF - radial basis function) semble de loin le plus efficace. En effet, son approche locale (due à la présence des gaussiennes en tant que fonctions d'activation) est d'un grand intérêt pour la surveillance. Ceci permet à ce type de structures de développer un apprentissage dynamique, en signalant les modes encore inconnus (élément fondamental en surveillance) et incluant éventuellement ces nouveaux modes, sans être obligé pour autant d'oublier tout ce qui a été appris précédemment.

Un autre problème d'une grande importance dans la surveillance industrielle, est constitué par la possibilité d'effectuer une détection des fausses alarmes et éventuellement de prédire une défaillance. Ces problématiques rentrent dans le cadre de la surveillance dynamique, méthode permettant à terme de s'orienter vers le concept de maintenance proactive et, plus particulièrement, vers le pronostic.

Dans ce contexte, nos travaux se sont orientés vers les réseaux de neurones temporels et plus précisément les réseaux de neurones dynamiques récurrents. Un état de l'art du domaine a ainsi été effectué, en focalisant très rapidement vers la catégorie des réseaux de neurones récurrents, la classe de réseaux temporels présentant une adéquation prononcée avec le domaine d'application visé – la surveillance dynamique.

A partir de ce constat, nos efforts de recherche se sont concentré sur la conception et la mise en œuvre d'une structure neuronale dynamique récurrente, faisant appel à une approche récurrente locale (*Localement Récurrentes Globalement Feedforward* – LRGF) et aux fonctions gaussiennes, très efficaces dans la surveillance statique. Nous avons ainsi proposé un concept nouveau – le *Réseau Récurrent à Fonctions de base Radiales* – RRFR (*RRBF-Recurrent Radial Basis Function*).

Ce chapitre est structuré en quatre parties. La première sera dédiée à une brève présentation des concepts fondamentaux liés aux réseaux de neurones, comme l'apprentissage et la généralisation. La deuxième partie présente un aperçu des principales contributions dans la surveillance et surveillance dynamique par réseaux de neurones. Un état de l'art synthétique sera ensuite introduit en troisième partie, pour finir avec l'introduction de notre nouvel outil neuronal récurrent – le RRFR.

4.2 PROPRIETES FONDAMENTALES DES RESEAUX DE NEURONES ARTIFICIELS

Dans ce paragraphe, nous nous proposons d'introduire brièvement les propriétés les plus importantes des réseaux de neurones artificiels, propriétés auxquelles nous allons faire référence tout au long de notre étude. L'ensemble des références bibliographiques étudiées, ainsi qu'un résumé étendu des différentes structures avec les techniques d'apprentissage correspondantes peuvent être consultées dans (**Zemouri, 2003**).

4.2.1 APPRENTISSAGE ET MEMOIRE

L'une des caractéristiques les plus complexes du fonctionnement de notre cerveau est bien la phase d'apprentissage. C'est une phase au bout de laquelle certaines modifications s'opèrent entre les connexions des neurones : certaines sont renforcées et d'autres affaiblies ou carrément inhibitrices. Le cerveau converge alors vers un comportement souhaité : par exemple l'apprentissage d'une langue, ou encore l'apprentissage par un enfant à reconnaître son environnement. Ceci nous emmène à la notion de mémoire qui donne au cerveau la capacité de retrouver des expériences passées. Le cerveau possède plusieurs types de mémoires. Nous ne nous attarderons pas sur ces différents types de mémoires mais ce que nous pouvons retenir c'est que le cerveau humain procède par association. Cela permet par exemple de retrouver une information à partir d'éléments incomplets ou imprécis (bruités). Par exemple, le fait de voir un bout d'une photographie qu'on connaît déjà est suffisant pour que notre cerveau soit capable de la reconnaître. Dans le paragraphe suivant, nous détaillerons d'avantage cette importante caractéristique des réseaux de neurones artificiels, plus connue comme *capacité de généralisation*. Le mécanisme de l'association permet aussi au cerveau de converger vers un état à partir d'un autre état. Par exemple, le fait de passer devant une boulangerie nous fait rappeler qu'on devait acheter du pain. Cette deuxième importante caractéristique est aussi connue sous le nom de mémoire adressée par le contenu, dont le modèle de Hopfield s'en inspire. Par analogie avec les réseaux de neurones biologiques, les réseaux de neurones artificiels tentent de reproduire les caractéristiques les plus importantes du comportement biologique, à savoir *l'apprentissage, la généralisation et l'association*.

L'apprentissage des réseaux de neurones artificiels est une phase qui permet de déterminer ou de modifier les paramètres du réseau, afin d'adopter un comportement désiré. Plusieurs algorithmes d'apprentissage ont été développés depuis la première règle d'apprentissage de Hebb (1949). Ces algorithmes d'apprentissage sont classés en deux catégories : *supervisé* et *non supervisé*.

Dans l'apprentissage *supervisé*, un superviseur (ou expert humain) fournit une valeur ou un vecteur de sortie (appelé cible ou sortie désirée) que le réseau de neurones doit associer au vecteur d'entrée. L'apprentissage consiste dans ce cas à modifier les paramètres du réseau de neurones afin de minimiser l'erreur entre la sortie cible et la sortie réelle du réseau de neurones.

Dans l'apprentissage *non supervisé*, les données ne contiennent pas d'informations sur une sortie désirée. Il n'y a pas de superviseur. Il s'agit de déterminer les paramètres du réseau de neurones suivant un critère à définir (fonction d'énergie à minimiser/maximiser).

4.2.2 SOUS-APPRENTISSAGE, GENERALISATION ET SUR-APPRENTISSAGE

La capacité de généralisation est une raison de base qui motive l'étude et le développement des réseaux de neurones artificiels. Elle peut être définie par la capacité d'élargir les connaissances acquises après apprentissage à des données nouvellement rencontrées par le réseau de neurones. C'est de cette façon que les réseaux de neurones sont capables d'approximer une fonction uniquement à partir d'une partie des données, ou encore d'associer un vecteur d'entrée qui n'a pas fait l'objet d'un apprentissage, à une classe. On peut distinguer deux types de généralisation : *locale* et *globale*.

Dans l'approche *locale*, chaque neurone est associé à une région d'activation (région d'influence) localisée dans l'espace des données. Seule une partie des neurones participe donc à la réponse du réseau. Deux types d'architectures neuronales possèdent cette particularité : les Réseaux à base de Fonctions Radiales et la carte de Kohonen.

Par contre, dans l'approche *globale*, l'ensemble des neurones du réseau participe à l'élaboration de la sortie du réseau. L'information est donc distribuée dans le réseau tout entier. C'est le cas des réseaux de neurones de type Perceptron Multi Couches ou le modèle de Hopfield. L'approche *globale* est supposée plus robuste aux *pannes* éventuelles de quelques neurones isolés. Par ailleurs, lors d'un apprentissage incrémental, des problèmes d'*interférences catastrophiques* peuvent apparaître : la modification des paramètres d'un neurone a des répercussions sur l'ensemble de la fonction modélisée par le réseau. Des régions de l'espace des données éloignées de la région de la donnée à mémoriser risquent d'être perturbées. En d'autres termes, tandis que le réseau apprend dans une région de l'espace des données, le modèle peut *oublier* ce qu'il a appris dans d'autres régions.

La figure ci-dessous illustre la différence de généralisation entre les deux architectures neuronales caractérisées par les deux fonctions d'activation (locale pour le RFR et globale pour le PMC) :

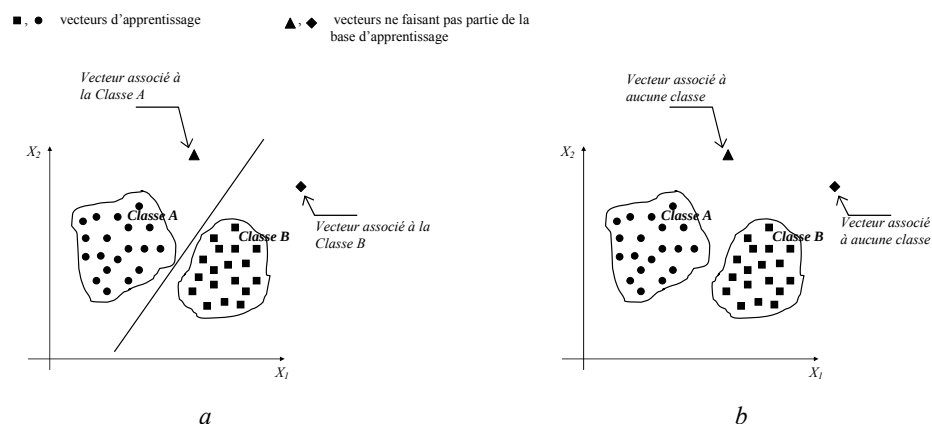


Figure 33. Différence des capacités de généralisation entre le PMC (a) et les RFR (b).

La capacité de généralisation est très liée à la notion de *surapprentissage*³⁰. Ces deux caractéristiques sont complètement antagonistes. On parle de sur-apprentissage quand le réseau a trop parfaitement appris les exemples proposés. Il sera donc incapable de généraliser.

³⁰ Over-fitting en anglais.

Un indicateur utilisé pour étudier ce phénomène est la *mesure de complexité k* du système d'apprentissage (indicateur lié à la taille du réseau de neurones). En pratique, on calcule alors la moyenne des erreurs quadratiques - appelée '*erreur base apprentissage*' - sur l'ensemble A de données d'apprentissage, et - '*erreur base test*' - sur l'ensemble T de données de test (différent de celui d'apprentissage). Plus on agrandit l'ensemble A , plus l'*erreur base apprentissage* diminue, plus l'*erreur base test* augmente, ce qui implique la perte des capacités de généralisation. La figure suivante illustre clairement ce compromis entre surapprentissage, sous-apprentissage et bonne généralisation.

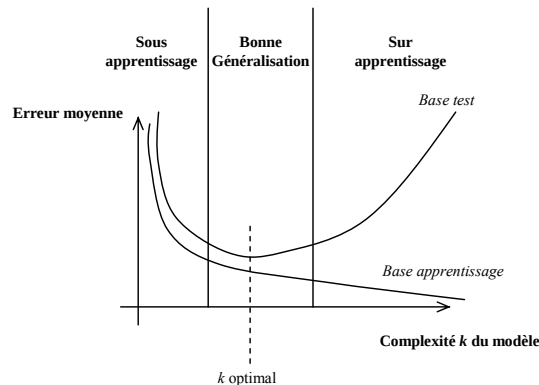


Figure 34. Complexité d'un réseau de neurones.

4.3 UTILISATION DES RESEAUX DE NEURONES POUR LA SURVEILLANCE INDUSTRIELLE

Ce paragraphe traite de manière non exhaustive quelques travaux sur les applications de surveillance industrielle. D'autres références non citées dans cette partie peuvent être trouvées dans (Koivo, 1994), (Bernauer, 1996), (Dubuisson *et al.*, 2001).

A titre d'exemple, un travail de synthèse de Koivo (1994) présente les architectures neuronales « statiques » les plus utilisées en surveillance industrielle, avec des résultats pratiques dans des applications statiques et dynamiques. Trois types de réseaux de neurones ont été testés : le Perceptron Multi Couche (*PMC*), le Réseau à Fonctions de base Radiales (*RFR*) et la carte de Kohonen. Les deux premiers donnent d'assez bons résultats en classification et en identification de processus. Les RFR peuvent se montrer plus performants que les PMC à condition de déterminer judicieusement leurs paramètres. Par contre, le réseau de Kohonen n'est pas aussi performant que les deux premiers mais ses capacités d'auto-adaptation (apprentissage non supervisé) sont très appréciées. L'auteur insiste dans sa conclusion sur la pertinence du choix du type de réseaux de neurones en fonction de l'application.

Nous introduisons par la suite quelques repères supplémentaires concernant l'utilisation de réseaux de neurones en surveillance, en abordant trois catégories d'applications : l'approximation de fonction pour la prédiction, la classification (discrimination) et la surveillance dynamique.

4.3.1 APPROXIMATION DE FONCTIONS ET PREDICTION PAR RN

La relation entrées-sorties d'un réseau de neurones a pour objet la reconstruction d'une variable réelle utile à une décision ultérieure. Les réseaux de neurones sont dans ce cas

utilisés comme un approximateur de fonctions non linéaires liant les variables de sortie aux variables d'entrée du système industriel. Ils représentent donc une boîte noire non linéaire modélisant le système. Cette boîte noire peut être *dynamique* ou *statique*, selon la nature du réseau de neurones (réseaux de neurones *temporels* ou *statiques*). L'avantage d'une telle technique de modélisation est qu'aucune connaissance (mathématique, physique ou autre) du système à modéliser n'est nécessaire. Le réseau de neurones intègre implicitement ces connaissances à travers le processus d'apprentissage. La réussite d'une telle technique de modélisation est donc tributaire du processus d'apprentissage.

Les travaux présentés par (Böhme *et al.*, 1999) constituent une application très intéressante de détection et localisation des défauts capteurs d'une centrale d'épuration hydraulique. Ceci est effectué par la reconstruction des signaux capteurs avec une comparaison de deux architectures neuronales : le Perceptron Multi Couches comparé à la carte de Kohonen. Le *PMC* est constitué de cinq couches, six neurones d'entrée et six neurones de sortie. Son objectif est de reconstituer six mesures de six sorties capteurs après un apprentissage par rétropropagation. Dans ce cas, ce réseau peut donc être considéré comme une mémoire auto associative. La détection ainsi que la localisation sont effectuées après une phase de comparaison avec seuillage de la sortie estimée avec la sortie réelle du capteur. Cette mémoire auto associative est donc comparée à la carte de Kohonen appelée carte topologique auto adaptative. Cette carte contient 15×15 neurones avec un vecteur d'entrée de 6 neurones (dimension de l'ensemble des sorties capteurs). Chaque neurone de la carte est caractérisé par un prototype et un paramètre définissant le rayon d'influence, déterminés par le processus d'apprentissage non supervisé. Pour chaque vecteur d'entrée, la réponse est donnée par un neurone gagnant qui correspond à celui dont la réponse de la fonction gaussienne est la plus importante. D'après la conclusion des auteurs, les deux techniques ont des performances similaires. Ces réseaux ont été testés sur deux types de fautes isolées et une succession de deux fautes. La carte de Kohonen se montre plus rapide pour la détection et l'identification du capteur défaillant. La technique devient insignifiante si plus de 50% des variables d'entrée sont erronées.

Une autre application de surveillance d'un moteur utilisant le même principe est présentée dans (Petsche *et al.*, 1996). Un Perceptron à trois couches utilisé comme mémoire auto associative sert à reconstituer le spectre du courant électrique. Après un apprentissage par l'algorithme de rétropropagation, la sortie de la mémoire auto associative est comparée à celle du spectre original. Tout écart supérieur à un seuil est synonyme de défaillance. Dans (Vemuri *et al.*, 1997), (Vemuri *et al.*, 1998) les réseaux de neurones ont été utilisés comme outils complémentaires à la modélisation d'un manipulateur robot. Un Perceptron à trois couches sert à estimer une fonction inconnue du modèle mathématique de fonctionnement du robot. Cette fonction représente des perturbations (défaillances) du manipulateur. L'apprentissage des paramètres du réseau s'effectue séquentiellement. La détection d'une défaillance se traduit par une génération de résidus. La variable de sortie du modèle de fonctionnement nominal (valeur estimée) est comparée à celle mesurée sur le robot. L'originalité de cette application se situe dans la modélisation non-linéaire du fonctionnement du manipulateur où les défaillances sont une fonction du temps, contrairement aux autres méthodes classiques où les modèles sont linéaires avec des défaillances additives. Dans (Lopes *et al.*, 1999) deux *PMC* à trois couches sont utilisés pour une application d'estimation et de prédiction de la qualité des pièces qui sortent d'un moule à injection. Un *PMC* sert à prédire (reconnaître) la classe représentant la qualité de ces pièces, et un autre *PMC* pour quantifier cette qualité. Les deux réseaux ont été appris par rétropropagation (1000 itérations pour le premier et 6000 pour le deuxième).

4.3.2 CLASSIFICATION OU DISCRIMINATION PAR RN

Lors de la mise en place d'un système de surveillance par reconnaissance des formes, l'expert est censé connaître les modes de bon fonctionnement et certains des modes de défaillances. Une grande partie des modes de bon fonctionnement est généralement fournie par les données du constructeur de l'équipement. Par contre, les informations concernant les modes de défaillance peuvent provenir de deux origines différentes : soit fournies par le constructeur ou par le bureau des études, soit collectées en cours de fonctionnement de l'équipement. Ces connaissances sont emmagasinées dans un historique de fonctionnement qui contient des informations sur les situations de dysfonctionnement d'un équipement. Les informations contenues dans l'historique de fonctionnement représentent la base d'apprentissage supervisé du réseau de neurones. La réussite d'une telle application est donc tributaire de la qualité des informations contenues dans l'historique de fonctionnement.

Les variables d'entrée peuvent être constituées par les différents paramètres mesurés sur le procédé. On cherche à associer un mode de fonctionnement (fonctionnement nominal, dégradé, ...) à ces variables d'entrée. Les variables de sortie sont alors des variables catégorielles, où chaque catégorie représente un mode de fonctionnement. Dans l'application décrite dans (Terstyanszky *et al.*, 2002) un réseau *RFR* est ainsi utilisé pour la surveillance d'un véhicule mobile autonome (*AMV* : Autonomous Mobile Vehicle). Le réseau de neurones associe cinq modes de défaillance à neuf variables mesurées en entrée. Les auteurs mettent en évidence les capacités du réseau *RFR* à reconnaître des nouvelles situations jusqu'alors non apprises. Cette capacité de généralisation locale des réseaux *RFR* est souvent appréciée dans les applications de surveillance industrielle.

Un autre exemple d'application des réseaux de neurones en surveillance par reconnaissance des formes est présenté dans (Keller *et al.*, 1994). Les auteurs ont utilisé deux réseaux de neurones pour la surveillance de l'environnement : le premier réseau de neurones de type *PMC* a pour but d'identifier et de quantifier les vapeurs chimiques de l'air. Le réseau de neurones associe neuf classes de sortie (huit classes représentant huit gazes ménagers et une classe de rejet en distance) à une signature de douze capteurs (neuf capteurs de gazes, un capteur de d'humidité et deux capteurs de température). L'apprentissage du réseau à trois couches (12 neurones d'entrée, 6 neurones cachés et 9 neurones de sortie) par l'algorithme de rétropropagation reste tout de même assez gourmand en temps (15000 itérations). Le deuxième réseau de neurones permet d'identifier et de quantifier un certain nombre d'isotopes radioactifs à partir des capteurs de radiations. Les auteurs ont utilisé une mémoire associative linéaire³¹ à deux couches : une couche d'entrée de 512 neurones et une couche de sortie de 8 neurones. Chaque neurone d'entrée est excité par l'amplitude d'une fréquence parmi les 512 fréquences d'un spectre de rayons gamma. L'avantage d'une telle application est de donner une réponse sur le type et la proportion d'isotopes contenue dans un objet et ceci en prenant en compte l'ensemble des fréquences du spectre. Cette propriété de fusion de données représente également un atout non négligeable pour les applications de surveillance par réseaux de neurones (Hashem *et al.*, 1995).

On peut trouver également dans (Wu *et al.*, 1994) et (Meador *et al.*, 1991) une application de surveillance de circuits intégrés par reconnaissance des formes. Les auteurs ont comparé les performances d'un *PMC* face à deux techniques de classification : l'estimateur gaussien avec le maximum d'appartenance et l'estimateur des *k* plus proches voisins. Les résultats de cette application semblent être plus favorables au *PMC*. La raison

³¹ Réseau de neurones dans lequel tous les neurones ont une fonction d'activation linéaire.

principale de ce résultat est que le *PMC* tend vers une structure qui minimise l'erreur quadratique moyenne lors du processus d'apprentissage. Cette caractéristique démarque l'utilisation des réseaux de neurones dans les problèmes de classification, d'autant plus que la procédure de calcul de la matrice de covariance de l'estimateur gaussien peut s'avérer très complexe. Le seul point négatif relevé par les auteurs est la lourdeur du temps d'apprentissage du Perceptron Multi Couches.

(Hines *et al.*, 1995) ont publié un travail assez original sur la surveillance des équipements d'énergie nucléaire. Leur méthode est basée sur une technique hybride qui combine méthode analytique à base de modèle de l'équipement et technique neuronale qui permet d'identifier la défaillance à partir des résidus obtenus par les techniques classiques de générations de résidus. Le réseau de neurones utilisé est un réseau à trois couches. Le nombre des neurones en entrée du réseau est donc égal au nombre de résidus. Chaque neurone de sortie représente une classe de fonctionnement (mode de fonctionnement de l'équipement). Les auteurs semblent hésiter entre deux types d'architectures neuronales : le Perceptron Multi Couches et le Réseau à Fonctions de bases Radiales. Le *PMC* est considéré comme une méthode de discrimination non-linéaire; par contre, le *RFR* ne couvre qu'une partie de son espace de données en fonction de ce qu'il a appris. Ce réseau est donc capable de dire « *je ne sais pas* », contrairement au *PMC* qui peut mal classer un mode nouvellement rencontré et induire ainsi l'expert en erreur. Les auteurs ont tout de même choisi le *PMC*, malgré les arguments précédemment évoqués. La raison essentielle de leur choix est que leur base de connaissance (base d'apprentissage) était assez exhaustive pour couvrir la quasi-totalité de l'espace des données d'entrée. Cette technique hybride semble donner des résultats assez satisfaisants. L'intérêt majeur des réseaux de neurones dans cette application par rapport aux autres méthodes classiques de détection des défaillances, concerne la résolution du problème de la redondance des alarmes.

Une autre application de surveillance par reconnaissance de formes concerne un problème d'hypovigilance d'un conducteur (Hernandez, 1999). Les auteurs ont utilisé deux techniques hybrides : la technique des Ensembles Flous Multidimensionnels³² et la technique des Réseaux de neurones à base de Fonctions Radiales Généralisés³³. Le problème consiste à associer des modes de conduite (normal, fatigue, alcool et inattention) à un ensemble de sorties capteurs. Ce problème devient très complexe si l'on souhaite modéliser le comportement du conducteur. Les techniques de l'*IA* sont donc indispensables dans une telle situation où les connaissances du système sont très minimes. Le traitement des chevauchements des classes de la base d'apprentissage (base de données superposées) est l'une des raisons du choix des auteurs pour les deux outils cités précédemment. Un test de chevauchement et décision finale permet de quantifier la proportion du chevauchement des classes de sortie. Les performances des deux techniques (*EFM* et *RFRG*) semblent être identiques avec un taux de réussite compris entre 78% et 90%.

4.3.3 SURVEILLANCE DYNAMIQUE PAR RN

Les applications précédentes montrent que les réseaux de neurones peuvent fournir des solutions très intéressantes pour les problèmes de surveillance sans modèle. Toutefois, un aspect fort important n'a pas encore été abordé jusque là, à savoir la dynamique du système à surveiller. En effet, toutes les applications présentées ne traitent que l'aspect statique des données capteurs. Ce type de traitement ne permet pas d'apprendre la dynamique d'un

³² *Multidimensional Fuzzy Sets*

³³ *Generalized Radial Basis Functions*

signal capteur afin de prédire une défaillance, ni même d'apprendre des séquences de fonctionnement d'un système à événements discrets. Les réseaux de neurones temporels offrent cette possibilité de prise en compte de l'aspect temporel des données.

Nous pouvons citer quelques travaux de surveillance par reconnaissance des formes avec réseaux de neurones temporels. Les travaux présentés dans (Bernauer *et al.*, 1993) et (Demmou *et al.*, 1995) traitent d'un problème de surveillance d'une cellule flexible d'assemblage. Un réseau de neurones récurrents permet d'apprendre des séquences temporelles booléennes qui représentent des gammes de bon fonctionnement. Après une phase d'apprentissage assez complexe, le réseau est capable de détecter des situations de défaillance système et de localiser l'origine de la défaillance. L'article présenté par (Rengaswamy *et al.*, 1995) constitue une autre application très intéressante des réseaux de neurones pour la surveillance dynamique d'une unité de craquage catalytique fluide (FCCU³⁴). Un Perceptron à trois couches permet de prendre en compte l'évolution temporelle des signaux capteurs. 21 défaillances sont associées aux variations de 7 sorties capteurs. Le diagnostic s'effectue grâce à une base de connaissance sous forme d'un arbre de défaillance reliant les effets constatés sur le système aux causes potentielles (les 21 défaillances du système). Les effets constatés sur le système sont caractérisés par le type d'évolution des sorties capteurs (brusque changement, évolution exponentielle, faible palier de dégradation, important palier de dégradation, ...etc.). Cette classification des signaux en fonction de leur évolution est effectuée par des réseaux de neurones à trois couches, un réseau par signal capteur. L'aspect temporel des signaux est pris en compte par cinq neurones d'entrée. Chaque neurone d'entrée représente une réponse d'un capteur pris à différents intervalles de temps ($x(t), \dots, x(t-4)$). Chaque neurone de la couche de sortie représente la classe du type d'évolution du signal. Le réseau est capable de distinguer sept classes différentes appelées primitives : trois droites (horizontale, croissante et décroissante) et quatre courbes (deux croissantes -convexe et concave- et deux décroissantes -convexe et concave-). L'apprentissage de 100 formes par l'algorithme de rétropropagation a nécessité près de 5000 itérations. La technique a été testée sur 63 cas de défaillances avec, pour chaque défaillance, trois degrés de sévérité. La technique semble donner de bons résultats, par contre aucun degré de sévérité n'a pu être estimé.

Les réseaux de neurones peuvent aussi être des outils assez efficaces pour des applications de prédiction ou pronostic où l'aspect dynamique (temporel) joue un rôle très important. Les résultats présentés dans (Chang *et al.*, 2001) semblent être très prometteurs pour des applications de prédiction neuronale des systèmes non-linéaires. Les auteurs ont appliqué un réseau *RFR* avec une fenêtre temporelle pour la prédiction des inondations d'un important fleuve situé à Taiwan sur un horizon de trois heures. Les paramètres du réseau sont déterminés en deux phases : une phase non supervisée avec l'algorithme Fuzzy Min-Max, et une phase supervisée avec la méthode de régression linéaire. Le réseau donne d'assez bons résultats sur plusieurs points tests. Nous pouvons encore citer les travaux de (Freitas *et al.*, 1999) pour la détection des défaillances d'un actionneur pneumatique. Le principe de la méthode consiste à comparer la sortie estimée avec celle du système réel. Après avoir intégré le résultat de la comparaison, le signal de sortie du système et le signal de commande, un PMC a pour rôle de reconnaître les modes de fonctionnement à partir de ces trois derniers signaux. L'estimation de la sortie de l'actionneur est assurée par un réseau de neurones récurrent (dynamique). L'estimation d'un tel système peut être très complexe avec d'autres outils mathématiques classiques car le système est non seulement non linéaire mais loin d'être périodique. La modélisation se fait par apprentissage. Notons que les réseaux de neurones statiques sont incapables de modéliser un tel système.

³⁴ Fluidized Catalytic Cracking Unit

4.4 REPRESENTATION DU TEMPS DANS LES RESEAUX DE NEURONES

Certaines fonctions de la surveillance industrielle, telle que la détection prédictive qui consiste à reconnaître le type de dégradation en fonction de l'évolution d'un signal (**Zemouri et al., 2003b**) et aussi à éliminer les fausses alarmes (**Zemouri et al., 2002b**), nécessitent la prise en compte d'un certain « *passé* » des informations analysées. Ce passé est nécessaire si l'on veut prédire l'évolution d'un signal à l'instant $t + \Delta t$ afin de prendre de l'avance sur les actions à entreprendre soit sur un équipement - comme la détection précoce des défaillances d'une colonne de distillation (Ploix et al., 1997) - soit sur tout un système donné - comme la prédiction du trafic de la téléphonie mobile (Legrand et al., 2002). La surveillance d'un système à événements discrets (SED) par réseau de neurones peut également exiger la prise en compte de l'aspect temporel pour l'apprentissage de séquences booléennes (Bernauer, 1996). Une autre application importante des réseaux de neurones nécessitant la prise en compte de la dimension temporelle est constituée par l'identification des systèmes dynamiques non linéaires (Urbani, 1995), (Yu et al., 2001), (Mirea et al., 2002), (Ferariu et al., 2002). Les applications où l'intégration du temps dans les réseaux de neurones est nécessaire sont bien évidemment plus larges que le domaine de la surveillance. Nous pouvons citer par exemple la reconnaissance en ligne de l'écriture (Garcia-Salicetti, 1996), (Chappelier, 1996) ou de la parole (Mellouk, 1994), (Adamson et al., 1996), ainsi que la prédiction de séries temporelles (Aussem, 1995), (Mangeas, 1996), (Atiya et al., 1999), (Rynkiewi, 2000), (Sinha et al., 2002) très utile pour la problématique des prédictions financières (Mozar, 1993).

Tous ces problèmes de détection de fausses alarmes, classification de dégradations, détection précoce des défaillances, surveillance des SED, etc. nécessitent la prise en compte de l'évolution temporelle des paramètres surveillés. La réponse en terme de réseaux de neurones est constituée par la classe des réseaux de neurones temporels.

4.4.1 ARCHITECTURES NEURONALES TEMPORELLES

La prise en compte de l'aspect temporel des données par les réseaux de neurones artificiels nécessite certaines modifications architecturales des modèles neuronaux statiques. Il existe en littérature deux façons distinctes d'aborder le temps par les réseaux de neurones (Chappelier et al., 1996), (Chappelier, 1996) : dans la première, le temps est représenté comme un *mécanisme externe* au réseau de neurones. Des retards (ou temporisations) servent à mémoriser les données d'entrée pendant une certaine durée τ_i . On présente au réseau un vecteur d'entrée comportant les données à l'instant t et ceux des instants $t - \tau_i$ (avec $i = 1, \dots, N$). On obtient alors une fenêtre temporelle de taille $N+1$. Cette technique a l'avantage de pouvoir utiliser les architectures de réseaux de neurones statiques. La prise en compte de l'aspect temporel est complètement transparente (Figure 35-a). Cette technique est aussi appelée *représentation spatiale* du temps selon Elman (1990). Par contre, dans la deuxième façon de prendre en compte le temps, le réseau de neurones est capable de traiter le temps sans aucun mécanisme externe (Figure 35-b). Cette représentation est appelée *représentation interne* selon Chappelier et *représentation dynamique* selon Elman.

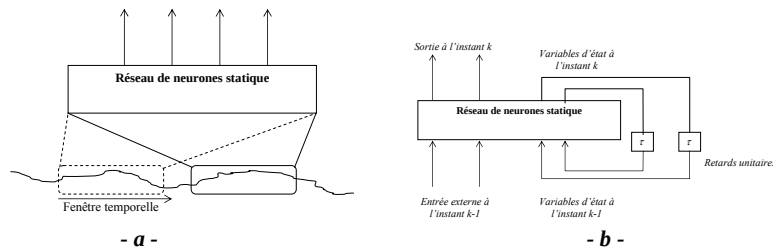


Figure 35. Exploitation des réseaux statiques pour des traitements dynamiques par utilisation de fenêtre temporelle (a) ou par utilisation de connexions récurrentes (b).

La Figure 36 présente les différentes façons de prendre en compte le temps dans les réseaux de neurones selon Chappelier et Grumbach. On peut également voir que la représentation interne se divise en deux possibilités : soit que le temps est pris en compte implicitement par la récurrence des connexions (réseaux de neurones récurrents), soit qu'il est pris en compte d'une manière explicite. Dans ce dernier cas, deux types de réseaux existent : dans le premier cas, les temporisations apparaissent au niveau des connexions. Dans ce type de réseaux, les connexions entre neurones possèdent non seulement des pondérations mais aussi des retards τ_i . L'apprentissage de ces réseaux consiste alors à trouver les valeurs des pondérations et des retards. Le deuxième cas de la prise en compte explicite du temps, se situe au niveau du neurone. On trouve alors soit des modèles biologiques ayant le souci de reproduire des comportements biologiques des neurones, soit des modèles où le temps est pris en compte par des mécanismes algébriques afin de résoudre des problèmes d'ingénierie, sans forcément se soucier de l'aspect biologique.

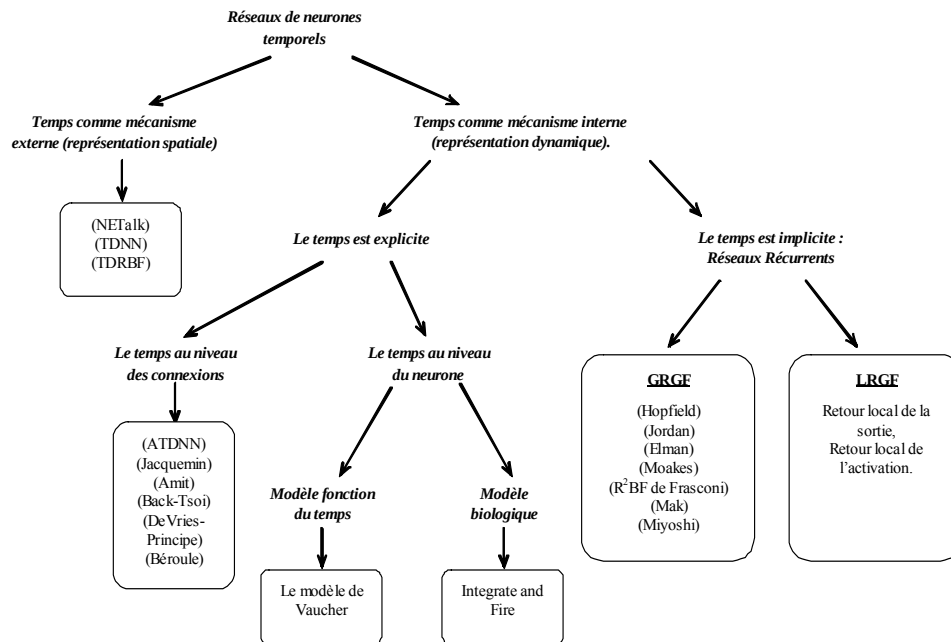


Figure 36. Représentation du temps dans les réseaux de neurones.

Sans soucis d'exhaustivité, nous indiquons quelques architectures de réseaux de neurones temporels pour chaque façon de représenter le temps. Certains types de réseaux seront détaillés dans ce rapport, tandis que d'autres seront simplement cités en indiquant les références bibliographiques correspondantes. Néanmoins, une présentation détaillée des principales techniques neuronales temporelles se trouve dans (Zemouri, 2003).

a) REPRESENTATION SPATIALE DU TEMPS

La façon la plus simple et immédiate de représenter le temps dans les réseaux de neurones est d'utiliser une représentation spatiale du temps. L'information temporelle contenue dans les données est alors transformée en une information spatiale, c'est à dire une forme qu'il s'agit de reconnaître. Dès lors, les techniques de classification par réseaux de neurones habituellement employées deviennent applicables. Cette transformation du temporel en spatial s'obtient par l'utilisation classique de ligne à retard. Au lieu de présenter au réseau chaque événement, dès son apparition, il convient d'attendre un certain temps avant de procéder à la classification de la forme obtenue. Chaque retard temporel représente une dimension de la représentation spatiale. Ce type de représentation du temps fait donc appel à un mécanisme externe qui est chargé de retarder ou de retenir un certain temps les données, ce qui conduit à l'appeler également représentation externe du temps. Les trois architectures neuronales les plus importantes qui utilisent ce principe sont : le *NETtalk* (Sejnowski et al., 1986), le *TDNN* (Waibel et al., 1989) et le *TDRBF* (Berthold, 1994a). Ces architectures sont principalement appliquées dans le domaine de la reconnaissance de la parole.

La représentation spatiale du temps présente plusieurs désavantages. Tout d'abord elle suppose l'existence d'une interface avec le monde extérieur dont le rôle est de retarder ou de retenir les données jusqu'au moment de leur utilisation dans le réseau. Se pose ainsi le problème de la connaissance de l'instant où les données doivent être traitées c'est-à-dire de la définition de la taille de la fenêtre. Ensuite, le fait d'utiliser une fenêtre temporelle de longueur finie et déterminée a priori soit par la plus longue information à traiter, soit en supposant la même longueur pour toutes les données, présente un second désavantage. Enfin, c'est dans la nature même de la représentation spatiale que se pose la difficulté de différencier une position temporelle relative d'une position temporelle absolue.

b) REPRESENTATION DYNAMIQUE DU TEMPS

Contrairement aux réseaux de neurones qui interprètent le temps comme un mécanisme *spatial* ou *externe*, les réseaux de neurones dits *dynamiques* traitent le temps d'une façon totalement interne au réseau. On peut distinguer deux manières de gérer le temps en interne : le temps est pris en compte *implicitement* ou *explicitement*. Dans les deux cas, le réseau possède la capacité de mémoriser des informations soit *implicitement* par la récurrence des connexions, soit *explicitement* par des retards au niveau des connexions.

b.1 Représentation explicite du temps dans les réseaux de neurones

Représentation explicite du temps au niveau des connexions : Les connexions à délais

Une des représentations explicites du temps dans les réseaux de neurones est l'utilisation des retards aux niveaux des connexions. La différence entre ce type de réseau de neurones et les réseaux récurrents se situe au niveau sens de la propagation du signal. Les réseaux à délais au niveau des connexions sont des réseaux *feedforward*. La Figure 37 montre un exemple d'un réseau de neurones avec des délais internes au niveau des connexions. Ce modèle introduit par Day et Davenport (Day et al., 1993) est appelé *Adaptive Time-Delay Neural Network* : *ATDNN*.

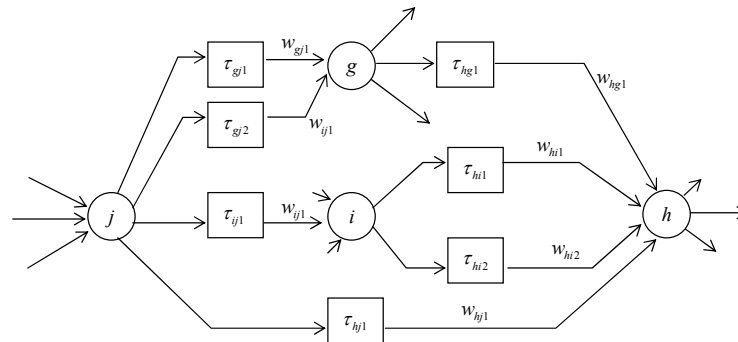


Figure 37. Exemple d'un réseau avec des connexions à délais.

L'algorithme d'apprentissage doit permettre non seulement une adaptation des poids des connexions mais également une adaptation des délais. Toute la difficulté que pose ce type de modèle à délais est la coordination entre la propagation du signal et la rétropropagation du gradient de l'erreur. Parmi les réseaux à retard que l'on peut trouver en littérature, on peut citer le modèle de Bérroule (Bérroule, 1985), le modèle d'Amit (Amit, 1988), le modèle de Back-Tsoi (Back *et al.*, 1990), le modèle de DeVries-Principe (DeVries *et al.*, 1991) et le réseau de Jacquemin (Jacquemin, 1994).

🚦 Représentation explicite du temps au niveau des neurones

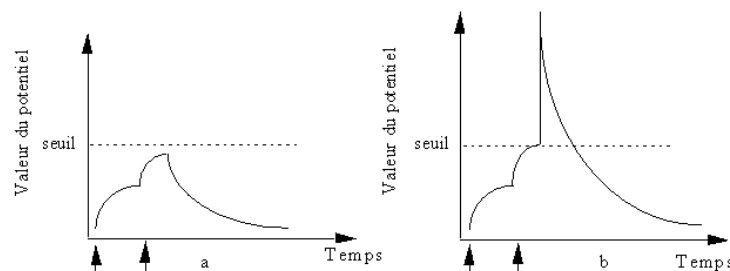


Figure 38. Sommation spatio-temporelle : addition des potentiels évoqués à la fois dans l'espace et dans le temps. a) 2 potentiels évoqués (repérés par les flèches) ne dépassent pas la valeur seuil. b) 2 potentiels évoqués qui dépassent la valeur seuil génèrent un potentiel d'action.

Le temps peut également être traité *localement* au niveau du neurone, ce qui permet d'avoir une certaine robustesse temporelle : deux entrées d'un neurone donné ne doivent plus être nécessairement synchrones pour avoir un certain effet (Chapellier, 1996). Ce modèle de neurone temporel peut être réalisé de deux manières différentes : soit en gardant les propriétés biologiques d'un neurone, soit en construisant un modèle pour résoudre un problème « d'ingénierie » sans autres types de préoccupations (ignorant complètement l'aspect biologique). Il existe de nombreux modèles d'inspiration biologique. La plus grande classe est ainsi formée par ce que l'on nomme les modèles « *Integrate and Fire* » (Koch, 1999), (Liu *et al.*, 2001). Leur principe est de sommer spatialement et temporellement « *Integrate* » les entrées leur provenant. Lorsque cette somme dépasse un seuil qui leur est propre, ils émettent « *Fire* » une impulsion (potentiel d'action) (Figure 38). Ces modèles possèdent ainsi des propriétés temporelles inspirées des neurones biologiques, tout en permettant un calcul rapide.

b.2 Représentation implicite du temps : Réseaux de neurones récurrents

La connectivité des unités dans les réseaux de neurones récurrents ne se limite pas, comme dans le cas des réseaux à propagation avant (*feedforward*), à des architectures dans

lesquelles l'information se propage de l'entrée vers la sortie, couche après couche. Tout type de connexion est admis, c'est à dire d'un neurone à n'importe quel autre, y compris lui-même. En d'autres termes, 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*. Une grandeur, à un instant donné, ne peut pas être fonction de sa propre valeur au même instant. Par conséquent, tout cycle du graphe des connexions d'un réseau de neurones récurrent doit comprendre au moins une connexion de retard unitaire (Dreyfus *et al.*, 2002). Ceci donne lieu à des comportements dynamiques fort intéressants mais qui peuvent être très complexes (Haykin, 1999).

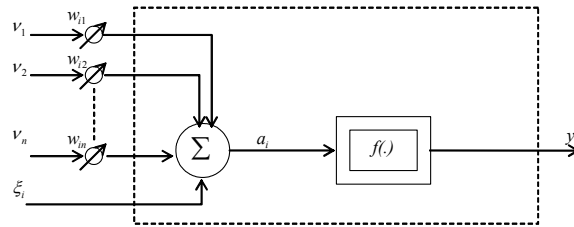


Figure 39. Représentation du comportement statique d'un neurone statique.

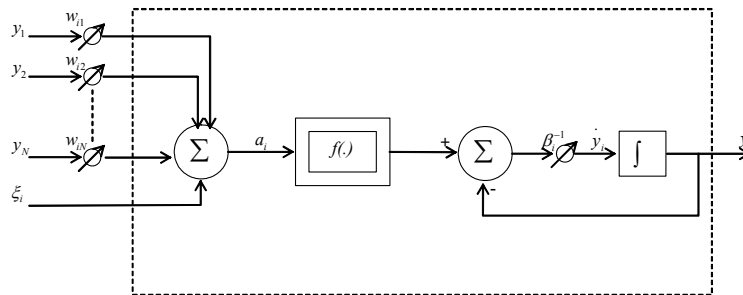


Figure 40. Représentation du comportement dynamique d'un neurone récurrent.

Les réseaux récurrents peuvent présenter deux types de comportements : le premier est que ces réseaux sont capables de se stabiliser dans un certain nombre de points de leur espace d'état, appelés points d'équilibre ou « *Fixed Point* ». Ces points constituent la réponse du réseau en présence d'une donnée et l'apprentissage consiste à affecter aux poids des connexions les valeurs permettant cette relaxation vers un point d'équilibre. L'architecture la plus connue et la plus ancienne ayant ce type de comportement est le modèle de Hopfield (Hopfield, 1982). Bien que ce réseau soit généralement utilisé pour apprendre un certain nombre d'associations statiques pour des problèmes d'optimisation multicritères, le réseau de Hopfield possède une certaine dynamique de convergence ; c'est-à-dire que le réseau passe par un certain nombre de points de son espace d'états avant d'atteindre un point d'équilibre. Ce type de réseau peut donc générer une séquence finie à partir d'un point donné de son espace d'état. Une variante de ce modèle a été proposée par Amit (Amit, 1988) pour la mémorisation de séquences temporelles. Son modèle se base sur celui de Hopfield pour stocker les diverses formes de la séquence, et le niveau temporel, représenté par des retards, est capable d'enchaîner les formes apprises.

Le second comportement temporel que produisent les réseaux récurrents est une succession d'états, ou plutôt de points dans l'espace d'états, mais sans qu'il y ait stabilisation en un point particulier. Il peut s'agir par exemple d'un cycle limite au cours duquel le réseau passe cycliquement par certains états. L'apprentissage appelé *Trajectory Learning* consiste alors à donner aux poids des connexions, les valeurs qui permettent au réseau de produire ce comportement particulier.

✚ Principales architectures de réseaux récurrents

- Architecture de Jordan

Dans l'architecture proposée par Jordan (1986) (Figure 41-a), les unités de la couche de sortie sont dupliquées sur une couche appelée *couche de contexte*. Les unités de cette couche tiennent également compte de leur propre état à l'instant précédent. Cette connexion récurrente d'une unité de contexte à elle-même lui confère une mémoire individuelle.

- Architecture d'Elman

Elman (1990) s'est inspiré en grande partie du réseau de Jordan pour proposer son architecture (Figure 41-b). Cette fois-ci, ce sont les unités de la couche cachée qui sont dupliquées dans la couche contexte avec un poids unitaire. L'apprentissage s'effectue par l'algorithme de rétropropagation et ne concerne que les poids de propagation avant.

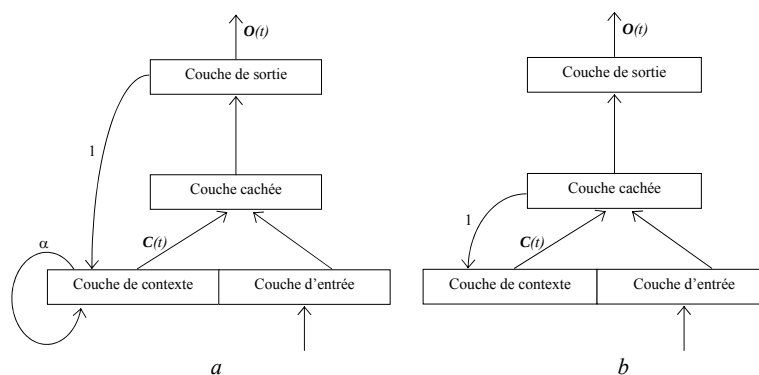


Figure 41. Architectures récurrentes : Jordan (a), Elman (b).

- Architecture de Moakes

Le modèle de la Figure 42-a introduit par Moakes (1994) représente une architecture particulière récurrente des réseaux *RFR* appliquée au traitement dynamique de la parole. Ce réseau combine récurrence (globale) des connexions et fenêtre temporelle. Des retards τ_n et τ_m sont associés respectivement au signal d'entrée $s(k-1)$ et au signal de sortie $y(k)$ rebouclé sur l'entrée du réseau. Cette architecture de *RFR* récurrent a été utilisée par (Billings *et al.*, 1995) comme un filtre non linéaire de bruit.

- Architecture de Mak

La Figure 42-b présente une autre architecture de réseau *RFR* récurrent (Mak, 1995). La récurrence des connexions se situe au niveau des sorties de chaque neurone gaussien. Les paramètres des fonctions gaussiennes sont déterminés par l'algorithme des *k-moyennes*, par contre, les poids des connexions récurrentes et de sortie sont déterminés par l'algorithme de rétropropagation du gradient. Le réseau a été testé sur une application d'apprentissage de séquences temporelles réelles d'une réponse impulsionnelle d'un filtre passe bas.

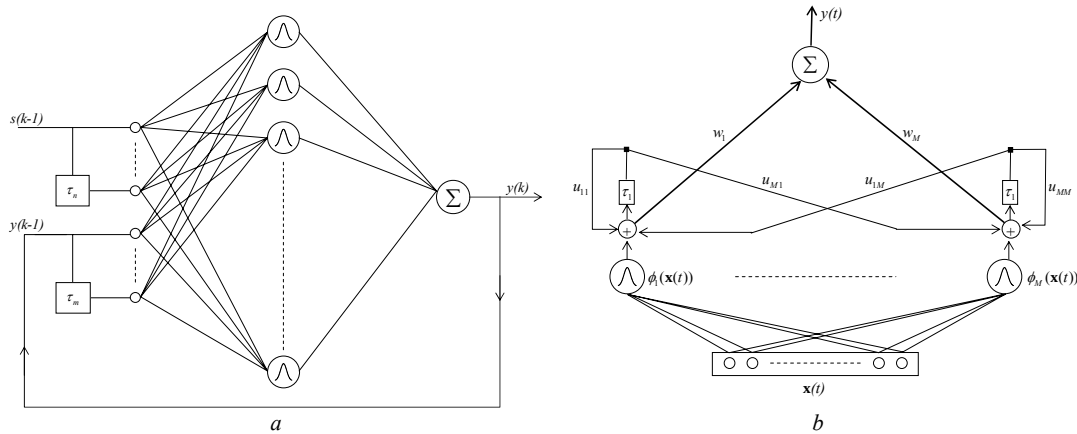


Figure 42. L'architecture de Moakes(a) et de M.W. Mak (b) pour un RFR récurrent.

- Architecture de Miyoshi

Le réseau proposé par Miyoshi (1995) (Figure 43) a été conçu pour l'identification et la prédiction des systèmes dynamiques chaotiques. Ce réseau récurrent se compose de plusieurs sous réseaux *RFR*. Chaque sous réseau r ($r = 1, \dots, N$) contient K cellules gaussiennes et un neurone de sortie de fonction d'activation linéaire. Cette architecture a été reprise par (Honda *et al.* 1998) qui a proposé un algorithme d'apprentissage.

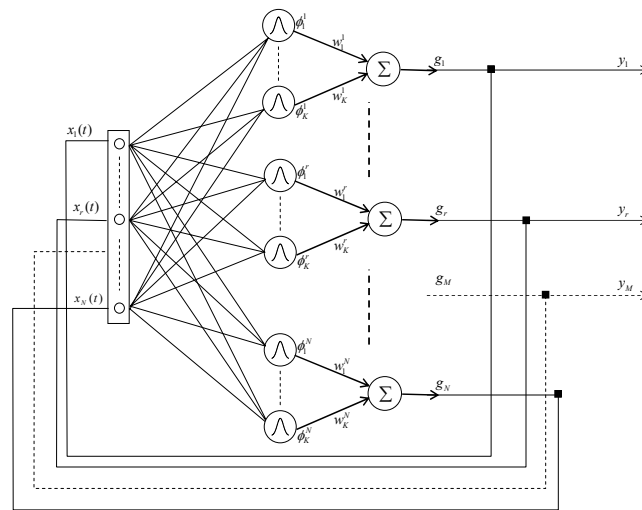


Figure 43. L'architecture de Miyoshi pour un RFR récurrent.

- L'architecture R^2BF

Le réseau R^2BF (Frasconi *et al.*, 1996)³⁵ représente (Figure 44) une architecture hybride entre neurones gaussiens (*représentation locale*) et neurones sigmoïdes (*représentation globale*). Une première couche cachée de neurones gaussiens est entièrement connectée à la deuxième couche cachée de neurones sigmoïdes. La sortie de ces neurones est connectée à un neurone de sortie, mais également réinjectée sur l'entrée des neurones gaussiens.

Ce réseau de neurones a été appliqué sur un problème d'apprentissage de séquences d'une grammaire d'un automate à états finis. La même application avec une architecture

³⁵ Recurrent Radial Basis Function

légèrement différente d'un *RFR* récurrent peut être trouvée dans (Sorel *et al.*, 2000), et avec un *PMC* récurrent dans (Giles *et al.*, 1992).

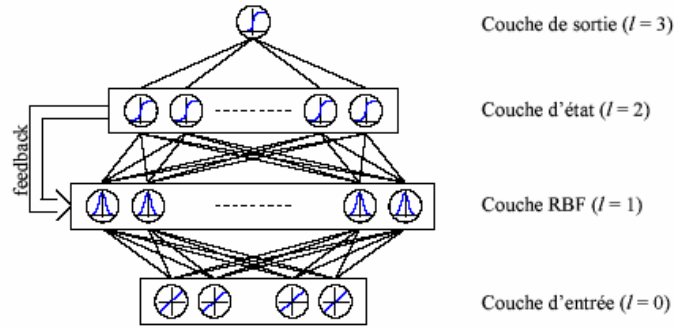


Figure 44. L'architecture de Frasconi pour un RFR récurrent (le R^2BF).

- Architecture DGNN

Ce réseau DGNN³⁶ (Ferariu *et al.*, 2002) présente un caractère générique, en incluant des neurones récurrents de type gaussiens, sigmoïdes et linéaires (Figure 45). A partir de l'application et des performances souhaitées, un algorithme génétique est mis en oeuvre afin de déterminer la topologie et les paramètres optimaux du réseau.

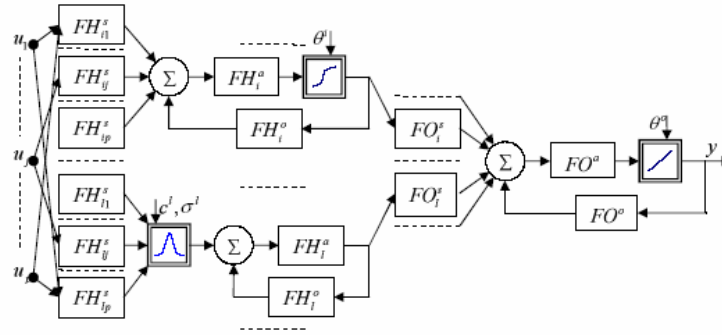


Figure 45. Topologie du DGNN

✚ Algorithmes d'apprentissage pour réseaux récurrents

- Fixed Point Learning

Ce type d'algorithme d'apprentissage permet d'affecter aux poids des connexions les valeurs assurant la convergence vers un point d'équilibre. Néanmoins, la convergence de ce type d'algorithme souffre dans certains cas d'une dépendance des conditions initiales. En effet, un changement infinitésimal des conditions initiales ou de la pente d'un point intermédiaire sur la trajectoire, peut changer le point d'équilibre vers lequel le système évolue (Pearlmutter, 1990).

- Rétropropagation récurrente

L'algorithme de rétropropagation récurrente a été proposé par Pineda (Pineda, 1987) et Almeida (Almeida, 1988) qui ont remarqué que l'algorithme de rétropropagation du gradient est un cas particulier d'un gradient de l'erreur plus globale.

³⁶ Dynamic General Neural Network

- *Trajectory Learning*

L'algorithme « *fixed point learning* » présenté précédemment est incapable de reproduire des séquences ou des trajectoires temporelles. On dira que ce sont des modèles qui *utilisent* le temps (Chappelier, 1996). En d'autres termes, la convergence du réseau vers un état stable suppose une certaine temporalité. Par contre, dans la deuxième catégorie d'algorithmes d'apprentissage, le temps est *traité* par le réseau de neurones. Le paramètre temps représente une donnée à *traiter* par le processus d'apprentissage. Leur but est donc d'apprendre au réseau de neurones à reproduire une séquence temporelle grâce à sa mémoire dynamique. On peut recenser cinq algorithmes d'apprentissage pour réseaux récurrents dit *Trajectory Learning* (Atiya *et al.*, 2000), (Aussem, 2002) : la Rétropropagation dans le temps ou *Back-Propagation Through Time : BPTT* (Werbos, 1990), (Rumelhart *et al.*, 1986) ; la Propagation Avant ou *Forward Propagation* appelé aussi *Real Time Recurrent Learning : RTRL* (Williams *et al.*, 1989) ; la Propagation Avant Rapide ou *Fast Forward Propagation : FFP* (Toomarian *et al.*, 1991) ; l'approche par Fonction de *Green* ou *Green Function : GF* (Sun *et al.*, 1990) et enfin l'approche par *Block Update : BU* (Schmidhuber, 1992). D'autres algorithmes d'apprentissage ont été proposés ces dernières années, parmi lesquels : Temporal Recurrent Back-Propagation qui se trouve à la croisée du *BPTT* et l'algorithme de rétropropagation récurrente par (Aussem, 1995) ; *Recursive Back-Propagation (RBP)* et sa version temps réel *Causal Recursive Back-Propagation (CRBP)* par (Campolucci *et al.*, 1999) ; une autre technique qui se base sur une approximation du gradient de l'erreur proposée par (Atiya *et al.*, 2000) ; et enfin une technique d'apprentissage appelée Statistical Approximation Learning (*SAL*) appliquée à une architecture de réseau récurrent bien particulière appelée Simultaneous Recurrent Networks (*SRN*) proposée par (Sakai *et al.*, 2002). Nous présentons brièvement les deux algorithmes d'apprentissage les plus utilisés pour les réseaux de neurones récurrents : *BPTT* et *RTRL*.

Rétropropagation dans le temps « Back-Propagation Through Time : BPTT »

Le principe de cette méthode consiste à déplier un réseau récurrent. Le réseau ainsi déplié est de type *feedforward* et peut faire l'objet d'un apprentissage par une version légèrement modifiée de l'algorithme de rétropropagation. La duplication (le dépliage) ne s'applique que pendant la phase d'apprentissage.

Le principal problème que pose l'algorithme de rétropropagation du gradient dans le temps est qu'il nécessite beaucoup de ressources informatiques. Pour de longues séquences ou des séquences de longueur inconnue, il devient inutilisable. Plus de détails sur ce type d'algorithme sont présentés dans (Pearlmutter, 1990), (Werbos, 1990), (Warwick *et al.*, 1992).

Le RTRL (Real Time Recurrent Learning)³⁷

L'algorithme de rétropropagation dans le temps représente une technique d'apprentissage hors-ligne. Williams et Zipser (Williams *et al.*, 1989) ont proposé un algorithme pour l'apprentissage des réseaux récurrents sans avoir à connaître la longueur de l'horizon temporel de la séquence d'apprentissage. La mise à jour des poids s'effectue au fur et à mesure que les données sont présentées au réseau sans avoir à dupliquer le réseau récurrent (Robinson *et al.*, 1987). Leur principal désavantage est que le réseau de neurones doit être entièrement connecté et souffre d'une certaine lenteur du temps d'apprentissage.

³⁷ On peut trouver en littérature l'appellation « *Forward Propagation : FP* »

Notons que cette démarche est analogue à celle utilisée habituellement sur les réseaux à propagation avant, qui consiste à appliquer les modifications aux poids après chaque exemple au lieu d'attendre la fin du cycle complet de présentation des données. Par ailleurs, cette technique ne garantit pas le suivi du gradient total de l'erreur de toute une séquence d'apprentissage (Tsoi *et al.*, 1994).

4.4.2 ANALYSE COMPARATIVE ENTRE LES REPRESENTATIONS DYNAMIQUE ET SPATIALE POUR LA SURVEILLANCE DYNAMIQUE

Les réseaux de neurones temporels se divisent donc en deux grandes catégories : réseaux de neurones *dynamiques* et *spatiaux*. Ces deux représentations du temps correspondent respectivement à une représentation *interne* et *externe* du temps. Les topologies des réseaux de neurones pour chaque représentation temporelle sont complètement différentes et chacune possède ses avantages et ses inconvénients.

Nous définissons trois types d'applications des réseaux de neurones temporels pour la surveillance dynamique industrielle : la reconnaissance de séquences temporelles (booléennes et réelles), la prédiction temporelle et la reproduction de séquences temporelles.

- *La reconnaissance de séquences temporelles :*

La reconnaissance de séquences temporelles consiste à produire une réponse spécifique lorsqu'une séquence particulière se produit à l'entrée du réseau. En d'autres termes, le rôle du réseau de neurones dans ce cas est de reconnaître une séquence temporelle bien particulière (application de reconnaissance des formes). C'est le cas par exemple de la reconnaissance de la parole où la sortie du réseau indique le mot qui vient d'être émis.

En surveillance, ce type d'application est utilisé selon deux manières différentes : la reconnaissance d'une séquence booléenne et la reconnaissance d'une séquence réelle. Pour le cas d'une séquence booléenne, le réseau de neurones est utilisé pour surveiller tout un Système à Événements Discrets (SED). Les variables d'entrée au réseau de neurones sont de type booléen (événementiel). Dans le deuxième cas, le réseau de neurones surveille un signal capteur d'un équipement industriel (variable de surveillance de type réel). Le but est de reconnaître un palier d'une dégradation précoce de l'équipement et d'éliminer les pics de changements brusques du signal, synonymes de fausses alarmes.

- *La prédiction temporelle :*

La prédiction temporelle consiste à donner une valeur future d'un signal capteur à partir des connaissances aux instants présents et passés de ce signal. En surveillance dynamique, ce type d'application est très important car prédire l'évolution d'un paramètre d'un équipement permet d'anticiper l'évolution future du signal d'un capteur afin de prendre des décisions préventives. Le réseau de neurones est dans ce cas utilisé comme approximateur universel (modélisation dynamique de l'équipement à surveiller).

- *La reproduction de séquences temporelles :*

La reproduction de séquences temporelles consiste à reproduire toute une séquence temporelle par le réseau de neurones. Ce type d'application est aussi très important en surveillance dynamique. Le réseau de neurones peut être utilisé pour reproduire un régime transitoire d'une évolution temporelle d'un paramètre d'un équipement industriel. La comparaison entre la sortie réelle et celle du réseau de neurones permet d'avoir un résidu pour la détection et le diagnostic de cet équipement. Souvent, le régime transitoire d'un

équipement, obtenu soit au démarrage de l'équipement soit entre changement de modes de fonctionnement, est très riche en informations utiles au diagnostic préventif.

Nous avons récapitulé sur le tableau ci-dessous les performances des principales architectures de réseaux de neurones temporels pour chacun des trois types d'application. Les réseaux spatiaux (*TDNN* et *TDRBF*) ainsi que les réseaux dynamiques à délais (*ATDNN*) se prêtent bien aux problèmes de reconnaissance de séquences. Les fenêtres temporelles dans le premier cas et les délais au niveau des connexions dans le deuxième cas permettent au réseau de neurones de prendre en compte un certain passé du signal afin de pouvoir donner une réponse par rapport à toute une séquence. Les réseaux biologiques en particulier les réseaux *Integrate and Fire* sont également capables de reproduire ce type de comportement. L'utilisation des réseaux spatiaux peut être plus souple que les deux autres architectures puisqu'un simple ajout d'une fenêtre temporelle peut rendre les architectures *feedforward statiques* capables de traiter le temps. L'inconvénient majeur de la représentation spatiale du temps est qu'elle suppose l'existence d'une interface avec le monde extérieur dont le rôle est de retarder ou de retenir les données jusqu'au moment de leur utilisation par le réseau : *comment connaître l'instant où les données doivent être traitées ?* La longueur de la fenêtre temporelle est finie et déterminée a priori, soit par la plus longue information à traiter, soit en supposant la même longueur pour toutes les données. C'est donc bien dans la nature même de la représentation spatiale que se pose la difficulté de différencier une position temporelle relative d'une position temporelle absolue (Elman, 1990). Les architectures à représentation spatiale du temps peuvent également être utilisées pour des applications de prédiction de séries temporelles (prédire la valeur de $x(t + \theta)$ à partir des connaissances des valeurs $[x(t - i)]_{i=0, \dots, \alpha}$) (Chang *et al.*, 2001).

	Représentation spatiale	Représentation dynamique			
		Le temps est explicite			Le temps est implicite -Réseaux récurrents-
		Le temps au niveau de connexions -Réseaux à retards-	Le temps au niveau du neurone		
			Modèle fonction du temps	Modèle biologique	
Reconnaissance de séquences	oui	oui	oui	oui	oui
Prédiction temporelle	oui	oui	oui	oui	oui
Reproduction de séquences	non	non	non	non	oui

Tableau 1. Résultats comparatifs entre les performances des architectures temporelles

Par contre, aucune des architectures neuronales temporelles, excepté les réseaux récurrents, n'est capable de reproduire des séquences temporelles. La raison est que seuls les réseaux récurrents possèdent des mémoires dynamiques grâce à la récurrence des connexions (Aussem, 1995). Le signal d'entrée ne se propage pas seulement de la couche d'entrée vers la couche de sortie comme dans les réseaux *feedforward*, mais se rétropropage également de la sortie vers l'entrée. Cette boucle fermée permet au réseau de garder en mémoire une trace interne d'un signal d'entrée, par exemple une impulsion, et de reproduire en sortie une séquence temporelle grâce aux algorithmes d'apprentissage de type *trajectory learning*. D'un autre côté, les temps d'apprentissage ainsi que les ressources informatiques nécessaires à leur mise en œuvre peuvent être relativement importants (Bernauer, 1996).

Le domaine d'application des réseaux récurrents se montre plus large que les autres architectures. Leur application dans des problématiques de surveillance dynamique est par

conséquent plus prometteuse que les autres architectures. Notre choix se porte donc naturellement vers ce type de réseaux de neurones temporels :

- d'un côté la prise en compte du temps est implicite, c'est-à-dire qu'on n'a pas besoin d'avoir un mécanisme externe pour retarder les données d'entrée (comme pour la représentation spatiale du temps),
- d'un autre côté, les réseaux récurrents sont bien les seuls réseaux à posséder une mémoire dynamique interne à travers la récurrence des connexions. Cette mémoire leur permet, non seulement de reconnaître des séquences temporelles et de faire de la prédiction de séries temporelles (comme les autres représentations temporelles) mais aussi d'apprendre à reproduire des séquences temporelles.

En contre partie, la présentation des principaux algorithmes d'apprentissage des réseaux récurrents montre que la phase de calcul des poids des connexions est très laborieuse. Les temps d'apprentissage et les ressources informatiques nécessaires à leur mise en œuvre sont relativement importants. Leur application en surveillance dynamique peut être très laborieuse, surtout pour des traitements en temps réel et pour un réseau entièrement récurrent (globalement récurrent). La solution envisagée pour contourner ces désavantages, passe par l'utilisation des réseaux partiellement récurrents et plus précisément des réseaux localement récurrents *LRGF* (*Locally Recurrent Globally Feedforward*).

4.5 UNE NOUVELLE STRUCTURE DE RESEAU DYNAMIQUE POUR LA SURVEILLANCE – LE RRFR – RESEAU RECURRENT A BASE DE FONCTIONS RADIALES (RRBF – RECURRENT RADIAL BASIS FUNCTION)

Comme nous l'avons constaté précédemment, une application de surveillance dynamique avec de réseaux de neurones récurrents peut être très compliquée à cause de l'apprentissage très complexe et coûteux en temps de calcul. Pour éviter la complexité du processus d'apprentissage, une façon simple d'avoir une mémoire dynamique interne au réseau de neurones est d'utiliser des récurrences locales au niveau du neurone lui-même. Ce type bien particulier de réseaux de neurones récurrents est appelé représentation *Localement Récurrente Globalement Feedforward* ou *Locally Recurrent Globally Feedforward* (*LRGF*). Dans ce type d'architecture, la récurrence des connexions est présente uniquement au sein du neurone lui-même. Ceci réduit considérablement le processus d'apprentissage tout en gardant un aspect dynamique fort important du réseau de neurones récurrent.

La mémoire de type *LRGF* nous permet de proposer un réseau *RFR* dynamique appelé *RRFR* : *Réseau Récurent à Base de Fonctions Radiales*. Cette architecture nouvelle de réseau *RFR* dynamique présente une structure plus souple avec un algorithme d'apprentissage simplifié. Ce réseau possède deux types de mémoire : une mémoire *statique* grâce aux neurones gaussiens de la couche cachée, et une mémoire *dynamique* grâce aux neurones localement récurrents de la couche d'entrée. Le réseau *RRFR* proposé profite de l'aspect dynamique des représentations *LRGF* tout en gardant la simplicité et l'efficacité des réseaux *RFR*.

4.5.1 ARCHITECTURE DU RRFR

En faisant appel aux récurrences locales de type LRGF, nous proposons un outil de surveillance capable de prendre en compte l'aspect dynamique des données d'entrée - afin de pouvoir détecter une dégradation ou prédire l'évolution d'une sortie capteur - et capable d'apprendre en continu les différents modes de fonctionnement d'un équipement sans avoir à oublier les connaissances précédemment acquises et, surtout, de garder une certaine simplicité d'utilisation du réseau de neurones pour des applications industrielles. L'architecture neuronale s'inspire des avantages des réseaux RFR et de ceux des réseaux récurrents. La Figure 46 présente l'architecture du RFR récurrent que nous proposons, appelé réseau RRFR : Réseau Récurrent à Base de Fonctions Radiales. Le réseau RRFR est composé de trois couches :

La couche ℓ_1 appelée *mémoire dynamique* du réseau, a comme rôle principal de prendre en compte la dynamique des données d'entrée (Figure 46). En d'autres termes, cette couche sert à mémoriser le « passé » du signal d'entrée du réseau. Cette mémoire dynamique est constituée par des récurrences locales au niveau des neurones d'entrée sans trop compliquer la topologie du réseau RFR. Cela revient, en quelque sorte, à « greffer » une mémoire dynamique aux réseaux RFR statiques. C'est pour cette raison que nous nous sommes intéressés aux réseaux Localement Récurrents connus sous l'appellation LRGF « Locally Recurrent Globally Feedforward ». Nous présentons une étude sur les architectures LRGF existantes en littérature avec des tests de performance pour chaque modèle afin de choisir celle qui serait la plus adaptée à la surveillance dynamique.

La couche ℓ_2 , appelée *mémoire statique* du réseau, a pour rôle de mémoriser les prototypes, comme pour les réseaux RFR classiques. Les entrées de cette couche ne proviennent plus directement des données d'entrée mais résultent de la sortie de la première couche, à savoir la mémoire dynamique. Un premier traitement des données d'entrée est donc effectué par la première couche avant que celles-ci soient mémorisées par la mémoire statique. Le prototype ainsi mémorisé par les neurones gaussiens aura implicitement pris en compte la dimension temporelle des données d'entrée. Nous détaillerons davantage les effets de la mémoire dynamique sur la mémoire statique au chapitre suivant.

La couche de sortie ℓ_3 qui donne la *décision de la classification* (dans le cas d'une application en reconnaissance de formes), ou la *valeur de la fonction à prédire* (dans le cas d'une approximation de fonction).

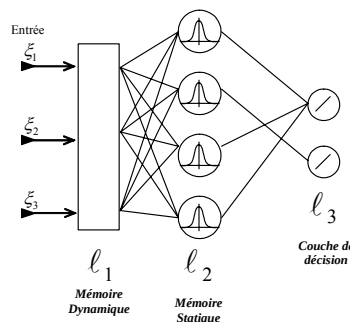


Figure 46. Réseau RRFR (Réseaux Récurrents à Fonctions de base Radiales).

4.5.2 MEMOIRE DYNAMIQUE DU RESEAU RRFR : ARCHITECTURE LRGF

L'appellation *LRGF* : *Locally Recurrent Globally Feedforward* de l'architecture *Localement Récurrence Globalement Feedforward*, reflète clairement le fait que les récurrences des connexions ne sont permises que localement au niveau du neurone (*Locally Recurrent*), la propagation du signal s'effectuant de la même manière que pour les réseaux statiques à propagation avant (*Globally Feedforward*). Une étude a été réalisée par A.C. Tsoi et A.D. Back (Tsoi *et al.*, 1994) sur les différentes architectures *LRGF* existantes en littérature. D'après cette étude, les auteurs ont classé les architectures *LRGF* en trois catégories :

- Les architectures à *retour local synaptique* (*Local Synapse Feedback*),
- Les architectures à *retour local de l'activation* (*Local Activation Feedback*),
- Les architectures à *retour local de la sortie* du neurone (*Local Output Feedback*).

Les architectures à retour local de l'activation et de la sortie interprètent bien la dimension temporelle implicitement par une récurrence des connexions. Par contre, concernant l'architecture à retour local synaptique, le temps n'est pas pris en compte par la récurrence des connexions, mais tout simplement par des retards synaptiques. Nous plaçons donc cette catégorie plutôt dans la représentation explicite du temps au niveau des connexions. Nous divisons donc les architectures *LRGF* en deux catégories : *retour local de l'activation* et *retour local de la sortie*. Ces deux catégories sont illustrées dans la Figure 47. D'après A.C. Tsoi et A.D. Back (Tsoi *et al.*, 1994), nous pouvons recenser deux modèles de neurones pour la représentation *LRGF* à retour local de la sortie : le modèle de *Frasconi-Gori-Soda* et celui de *Poddar-Unnikrishnan*, et un seul modèle pour l'architecture *LRGF* à retour local de l'activation, le modèle de *Frasconi-Gori-Soda*.

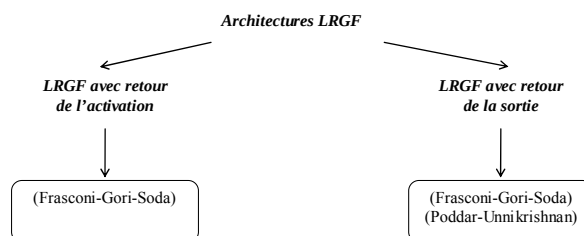


Figure 47. Représentation des différentes architectures *LRGF*.

Les conclusions de A.C. Tsoi et A.D. Back sont nettement en faveur des architectures *LRGF* dont les performances ont été comparées à celles des réseaux globalement récurrents - précisément le réseau récurrent de *Williams-Zipser*³⁸ (Williams *et al.*, 1989). En effet, les auteurs ont comparé quatre architectures neuronales : le réseau de *Back-Tsoi*, le neurone à retour local de la sortie *Frasconi-Gori-Soda*, le réseau entièrement récurrent de *Williams-Zipser* et le *TDNN*. Le réseau le plus performant est celui de *Frasconi-Gori-Soda* suivi par le *TDNN*. Le réseau le moins performant est le réseau *Williams-Zipser* avec la convergence la plus lente. Le réseau de *Williams-Zipser* souffre également de considérables difficultés pour la modélisation de signaux dynamiques (précisément des signaux vocaux).

a) ARCHITECTURE LRGF AVEC RETOUR DE LA SORTIE

La première représentation implicite du temps au niveau du neurone est caractérisée par un bouclage de la sortie du neurone vers son entrée. Ce type de neurone est appelé neurone à retour local de la sortie ou *Local Output Feedback network*. Nous pouvons

³⁸ Réseau récurrent dont l'algorithme d'apprentissage est le *RTRL*

recenser en littérature deux types de neurones à retour local de la sortie (Tsoi *et al.*, 1994) : le modèle de *Frasconi-Gori-Soda* (Frasconi *et al.*, 1992), (Gori *et al.*, 1989, 1990) et le modèle de *Poddar-Unnikrishnan* (Poddar *et al.*, 1991a, b).

a.1 Le modèle de Frasconi-Gori-Soda

L'élément qui différencie le modèle du retour local de l'activation et celui du retour local de la sortie introduits par Frasconi-Gori-Soda est, bien évidemment le point de retour qui se situe au niveau de l'activation dans le premier modèle et au niveau de la réponse du neurone dans le second modèle. Dans le modèle avec bouclage de la sortie, l'évolution de la sortie du neurone dépend de ses propres réponses antérieures :

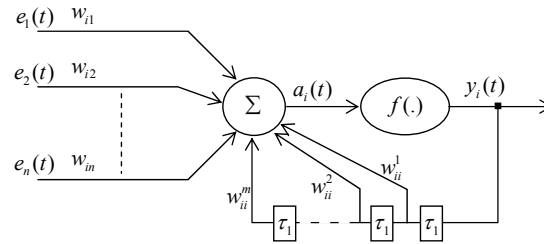


Figure 48. Architecture générale d'un réseau LRGF avec bouclage de la sortie (modèle de Frasconi-Gori-Soda). Les τ_1 représentent des retards unitaires.

En d'autres termes, l'activation du neurone est fonction de ses entrées pondérées (comme le neurone classique) et des valeurs retardées de sa sortie. Sa sortie $y_i(t)$ est donc définie par les équations suivantes :

$$y_i(t) = f(a_i(t))$$

$$a_i(t) = \sum_{j=1}^n w_{ij} e_j(t) + \sum_{q=1}^m w_{ii}^q y_i(t - q) \quad [1]$$

Lorsqu'il s'agit d'un neurone d'entrée au réseau, et si l'on simplifie le modèle de la Figure 48 en ne considérant qu'un seul retard, son activation peut être ramenée à l'équation ci-dessous. Ce type de neurone est communément appelé *neurone bouclé* (Figure 49).

$$a_i(t) = \xi_i(t) + w_{ii} y_i(t - 1) \quad [2]$$

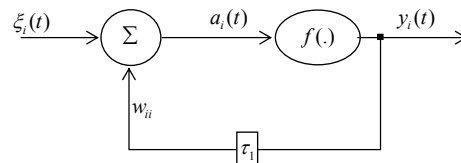


Figure 49. Architecture simplifiée d'un Réseau LRGF avec retour de la sortie.

La fonction d'activation $f(\cdot)$ du neurone bouclé est la sigmoïde³⁹ (Figure 50) définie par :

$$f(x) = \frac{1 - \exp^{-bx}}{1 + \exp^{-bx}} \quad [3]$$

³⁹ Tangente Hyperbolique. Nous supposons que $b > 0$ ce qui rend cette fonction strictement croissante.

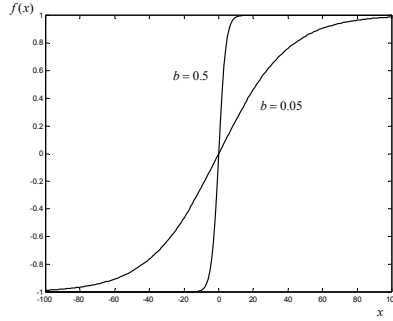


Figure 50. Allure de la sigmoïde en fonction du paramètre b .

Points d'équilibre

Pour étudier l'impact du bouclage de la sortie sur le comportement dynamique du neurone, nous allons considérer son évolution en l'absence de toute excitation extérieure ($\xi_i(t) = 0 \quad \forall t > 0$) (Frasconi *et al.*, 1992), (Bernauer, 1996). Dans ces conditions, l'activation du neurone bouclé est définie par l'équation suivante :

$$a_i(t) = w_{ii} f(a_i(t-1)) \quad [4]$$

La recherche des points d'équilibre de cette équation, c'est-à-dire de l'ensemble des points a^* tels que $a^* = w_{ii} f(a^*)$, revient à chercher les racines de la fonction g définie par⁴⁰ :

$$g(a) = w_{ii} f(a) - a \quad [5]$$

D'après la définition de la fonction $f(x)$, $f(0) = 0$. $a_0 = 0$ est une solution évidente de $g(a) = 0$. La recherche des autres racines de $g(a)$ se fait par l'étude de ses variations.

- Si $bw_{ii} > 2$, le neurone bouclé possède trois points d'équilibre a_0^- , a_0 et a_0^+
- Si $bw_{ii} \leq 2$ la fonction $g(a)$ est toujours décroissante et donc l'équation $g(a) = 0$ ne possède qu'une seule solution. Le neurone bouclé ne possède alors qu'un seul point d'équilibre a_0 .

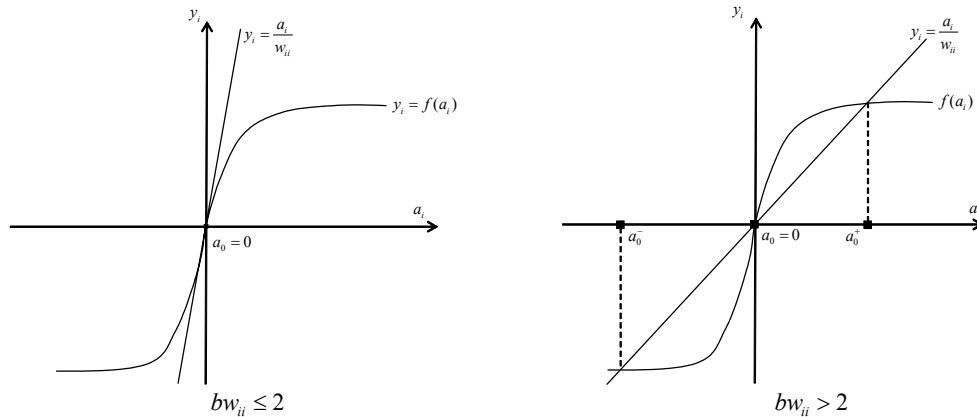


Figure 51. Points d'équilibre du neurone bouclé en fonction du produit bw_{ii} .

⁴⁰ Nous considérons bien évidemment que l'auto-connexion $w_{ii} \neq 0$, sinon on ne peut pas parler de réseaux LRGF.

✚ Stabilité des points d'équilibre

L'étude de la stabilité des points d'équilibre se fait en utilisant la fonction de Lyapunov définie par $V(a) = a^2$. Nous concluons ainsi que :

Le neurone bouclé de Frasconi-Gori-Soda possède

- un seul point d'équilibre stable $a_0 = 0$ si $bw_{ii} \leq 2$ avec $w_{ii} > 0$,
- deux points d'équilibre stables a_0^- et a_0^+ , et un point d'équilibre instable $a_0 = 0$ si $bw_{ii} > 2$.

✚ Comportement d'oubli

Considérons le neurone bouclé de la Figure 48 dont l'évolution est donnée par l'équation suivante :

$$\begin{aligned} y_i(t) &= f(a_i(t)) \\ a_i(t) &= \xi_i(t) + w_{ii}y_i(t-1) \end{aligned} \quad [6]$$

avec $f(.)$ la fonction d'activation sigmoïdale du neurone bouclé (équation [3]) et $w_{ii} > 0$.

Propriété

Un neurone bouclé possède un comportement d'oubli si pour un instant donné t_0 et une activation $a_i(t_0)$ quelconque, $\exists q > 0$, $\forall \varepsilon > 0$, $\forall t > t_0 + q$, $\forall \theta > 0$, tel que :

$$\begin{cases} \text{si } a_i(t_0) > \theta \Rightarrow \varepsilon > a_i(t) \geq 0 \\ \text{si } a_i(t_0) < -\theta \Rightarrow -\varepsilon < a_i(t) \leq 0 \end{cases} \quad [7]$$

Lors de l'étude de la stabilité des points d'équilibre nous avons constaté que le neurone bouclé ne possède qu'un seul point d'équilibre stable $a_0 = 0$ si $bw_{ii} \leq 2$ avec $w_{ii} > 0$ et en l'absence de toute excitation extérieure, $\xi_i(t) = 0 \quad \forall t > 0$. Dans ce cas, pour n'importe quelle activation $a_i(t) \neq 0$ du neurone bouclé, l'activation du neurone tend vers $a_0 = 0$. Par conséquent, nous obtenons $\lim_{t \rightarrow \infty} a_i(t) = a_0 = 0$. Le neurone bouclé possède bien un comportement d'oubli si $bw_{ii} \leq 2$ et $w_{ii} > 0$:

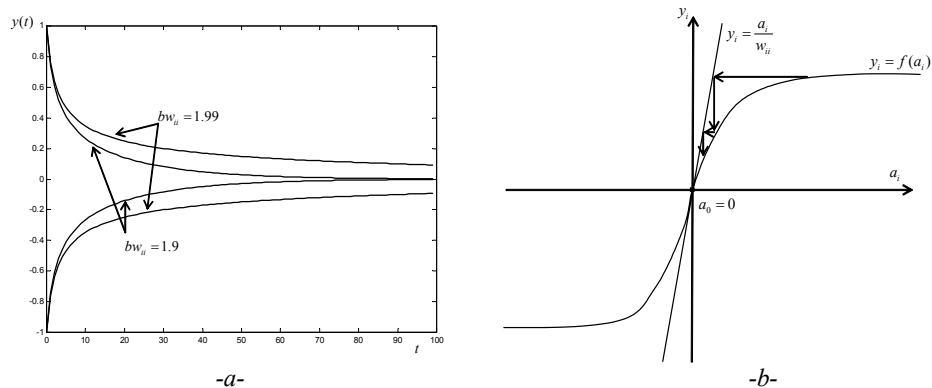


Figure 52. Comportement d'oubli : a) Convergence de la sortie du neurone bouclé vers le point d'équilibre stable $a = a_0$ ($w_{ii} > 0$), b) Etapes du comportement d'oubli

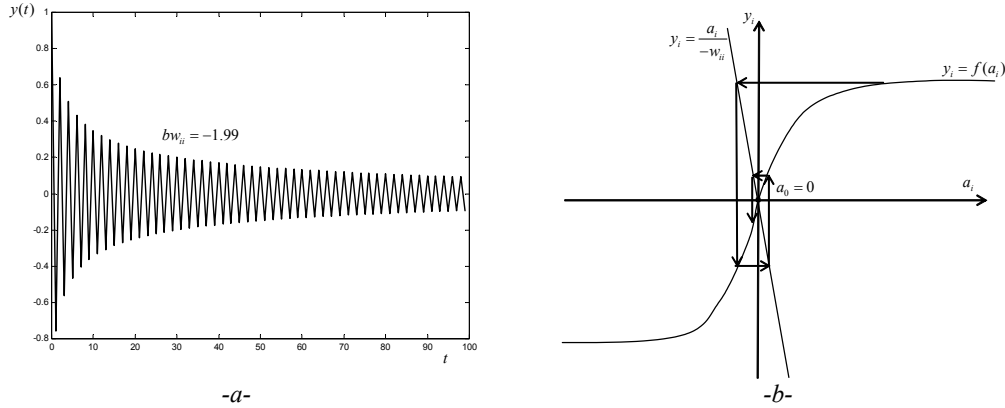


Figure 53. Comportement d'oubli : a) Oscillations dues à la valeur négative de l'auto-connexion $w_{ii} < 0$. b) Etapes du comportement d'oubli

Comportement de mémorisation

Propriété

Un neurone bouclé possède un comportement de mémorisation si pour un instant donné t_0 et une activation $a_i(t_0)$, $\exists \varepsilon > 0 : \forall t > t_0$, tel que :

$$\begin{cases} \text{si } a_i(t_0) > 0 \Rightarrow a_i(t) \geq \varepsilon > 0 \\ \text{si } a_i(t_0) < 0 \Rightarrow a_i(t) \leq -\varepsilon < 0 \end{cases} \quad [8]$$

Le neurone bouclé possède trois points d'équilibre $a_0^- < 0$, $a_0 = 0$ et $a_0^+ > 0$ si $bw_{ii} > 2$. D'après la théorie de Lyapunov, les deux points a_0^- et a_0^+ sont des points d'équilibre stables et $a_0 = 0$ est un point d'équilibre instable. Donc si la sortie du neurone bouclé s'éloigne du point $a_0 = 0$ du côté supérieur $a(t_0) > a_0$ ou inférieur $a(t_0) < a_0$ c'est pour tendre vers un point d'équilibre stable (a_0^+ pour le premier cas et a_0^- pour le deuxième) :

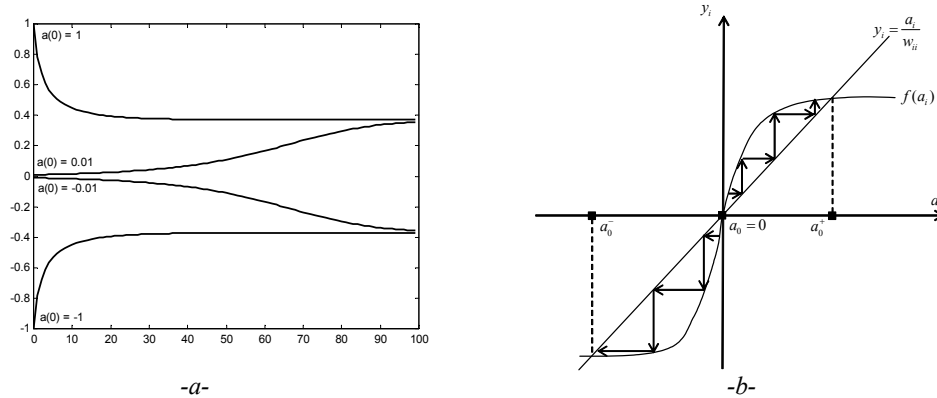


Figure 54. Mémorisation : a) Convergence de la sortie du neurone bouclé vers le point d'équilibre stable a_0^+ pour $a(0) > 0$ et vers a_0^- pour $a(0) < 0$, b) Etapes de la mémorisation.

Ce comportement avec deux points d'équilibre stables reste valable même si on ajoute à l'activation du neurone un terme extérieur constant ξ_i tel que (Frasconi *et al.*, 1995) :

$$y_i(t) = f(\xi_i + w_{ii}y_i(t-1)) \quad \text{et} \quad [9]$$

$$|\xi_i| < I^*$$

avec I^* le point d'intersection de la tangente de $f(\cdot)$ parallèle à la droite $y_i = a_i / w_{ii}$ avec la droite $y_i = 0$ (Figure 55). Dans le cas contraire, si $|\xi_i| > I^*$, il n'y aura plus qu'un seul point d'équilibre stable (positif si $\xi_i > I^*$ et négatif si $\xi_i < -I^*$), et la sortie du neurone bouclé va tendre en un nombre fini de transitions vers l'état d'équilibre proche de 1.

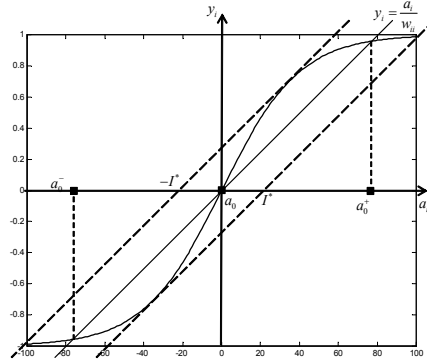


Figure 55. Limites de mémorisation à deux points d'équilibre stable du neurone bouclé.

Longueur de la mémoire dynamique du neurone bouclé

Propriété

La limite de la longueur de la mémoire dynamique d'un neurone bouclé peut être quantifiée par l'étude de l'influence d'une variation $\partial \xi_i(t)$ de l'entrée à un instant t donné sur les variations de la sortie du neurone bouclé $\partial y_i(t+q)$ à un instant $t+q$. On dira qu'un neurone bouclé a atteint sa limite maximum de mémorisation dynamique $q_{\max}$ si au bout d'un certain temps, la variation de l'entrée $\partial \xi_i(t)$ n'a aucune influence sur la variation de la sortie $\partial y_i(t+q)$. On peut exprimer ceci par l'équation suivante :

$$\lim_{q \rightarrow \infty} \frac{\partial y_i(t)}{\partial \xi_i(t-q)} = 0 \quad [10]$$

Nous avons démontré (Zemouri, 2003) que pour $bw_{ii} < 2$

$$\lim_{q \rightarrow \infty} \frac{\partial y_i(t)}{\partial \xi_i(t-q)} \leq \lim_{q \rightarrow \infty} (bw_{ii}/2)^q (b/2) = 0 \quad [11]$$

Dans ce cas, le neurone bouclé possède bien une limite de mémorisation.

Dans le cas $bw_{ii} \geq 2$, en procédant par simulation des équations obtenues, on obtient ainsi les résultats schématisés par la Figure 56.a. qui montre que le maximum $\frac{\partial y_i(t)}{\partial \xi_i(t-q)}$ est obtenu pour $bw_{ii} = 2$. Nous avons ensuite quantifié la valeur maximale de la longueur de la mémoire dynamique. Nous avons donc fixé $bw_{ii} = 2$ et calculé $\frac{\partial y_i(t)}{\partial \xi_i(t-q)}$ en fonction de q . On obtient une longueur maximale d'environ 300 unités de temps (Figure 56.b.).

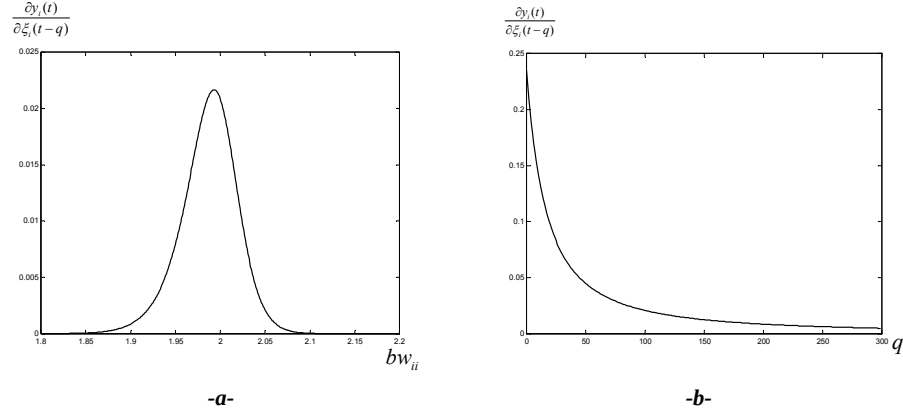


Figure 56. Longueur de la mémoire dynamique du neurone bouclé : a) en fonction du produit bw_{ii} pour un q donné, b) en fonction de q avec une configuration de longueur de mémoire maximum ($bw_{ii} = 2$) obtenue à partir du premier graphe.

a.2 Le modèle de Poddar-Unnikrishnan

La deuxième représentation locale du temps par retour de la sortie est celle du modèle de *Poddar-Unnikrishnan* illustré par la Figure 57. La sortie du neurone est régie par les relations ci-dessous :

$$y_i(t) = f(a_i(t))$$

$$a_i(t) = \sum_{j=1}^n w_{ij}e_j(t) + w_{ii}z_i(t) \quad [12]$$

$$z_i(t) = \alpha y_i(t-1) + (1-\alpha)z_i(t-1)$$

On remarque que le neurone de *Frasconi-Gori-Soda* présenté précédemment représente un cas particulier du modèle de *Poddar-Unnikrishnan* où $\alpha_i = 1$.

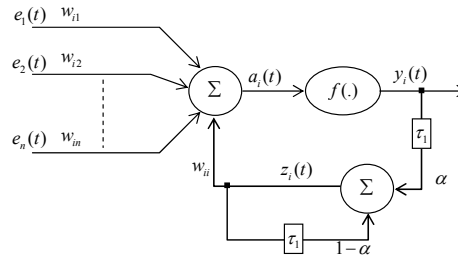


Figure 57. Architecture générale d'un réseau LRGF avec bouclage de la sortie (modèle de Poddar-Unnikrishnan). Les τ_1 représentent des retards unitaires.

Lorsqu'il s'agit d'un neurone d'entrée au réseau, son activation peut être ramenée à l'équation suivante :

$$a_i(t) = \xi_i(t) + w_{ii}z_i(t) \quad [13]$$

En procédant de façon similaire au cas précédent, nous obtenons des informations sur les points d'équilibre :

- si $bw_{ii} \leq 2$ le neurone ne possède qu'un seul point d'équilibre $a_0 = 0$,

- si $bw_{ii} > 2$ le neurone possède trois points d'équilibre a_0^- , a_0 et a_0^+ .

sur leur stabilité :

- un seul point d'équilibre stable $a_0 = 0$ si $bw_{ii} \leq 2$ avec $w_{ii} > 0$ et $0 < \alpha < 1$,
- deux points d'équilibre stables a_0^- et a_0^+ , et un point d'équilibre instable $a_0 = 0$ si $bw_{ii} > 2$ et $0 < \alpha < 1$.

sur le comportement d'oubli : comportements distincts en fonction de bw_{ii} pour $0 < \alpha < 1$:

- un comportement d'oubli si $bw_{ii} \leq 2$ avec $w_{ii} > 0$,
- un comportement de mémorisation si $bw_{ii} > 2$.

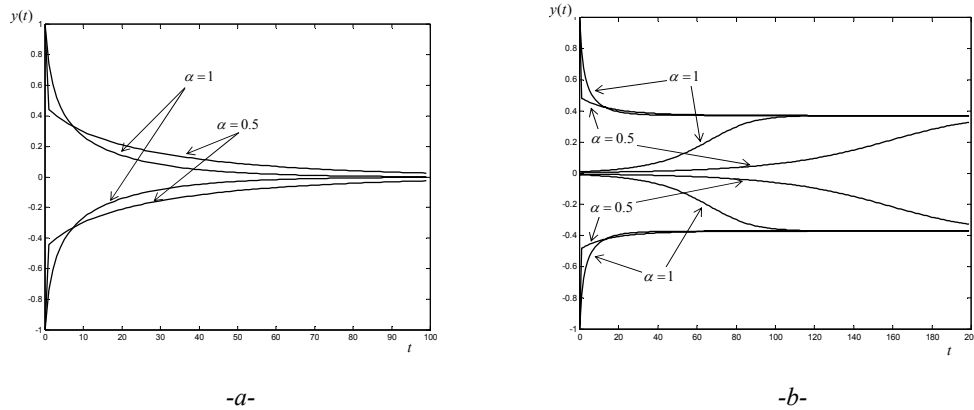


Figure 58. Influence des paramètres α et bw_{ii} sur le comportement du neurone bouclé de Poddar-Unnikrishnan : a) comportement d'oubli, b) comportement de mémorisation.

et sur la longueur maximum de la mémoire dynamique, obtenue pour $bw_{ii} = 2$ et $\alpha = 1$. Cette longueur maximale est $q = 300$. On rappelle que le cas où $\alpha = 1$ représente un cas particulier du neurone bouclé de Frasconi-Gori-Soda présenté précédemment :

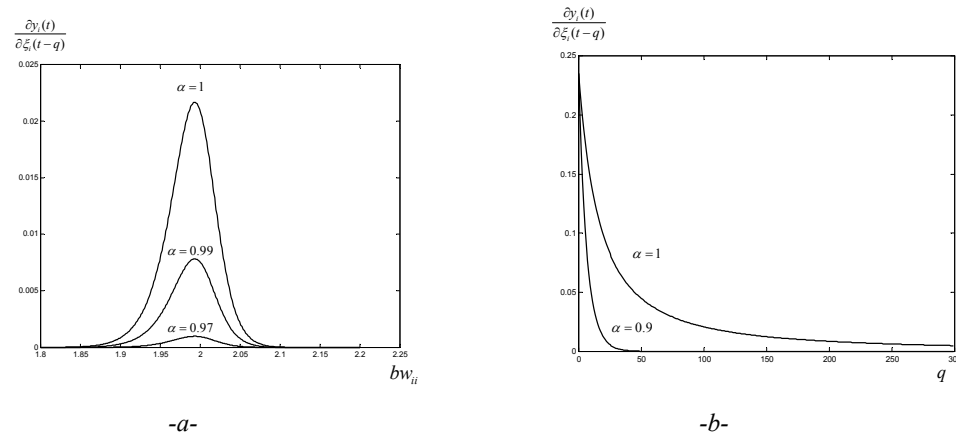


Figure 59. Longueur de la mémoire du neurone de Poddar-Unnikrishnan : a) en fonction de bw_{ii} et α pour un q donné, b) en fonction de q et α avec une configuration de longueur de mémoire maximum ($bw_{ii} = 2$) obtenue à partir du premier graphe.

b) ARCHITECTURE LRGF AVEC RETOUR DE L'ACTIVATION

La deuxième représentation locale du temps implicitement au niveau du neurone est celle présentée par la figure suivante. Cette architecture particulière de réseaux LRGF introduite par Frasconi-Gori-Soda (Gori, 1989) est appelée *local activation feedback network* ou réseau à retour local de l'activation.

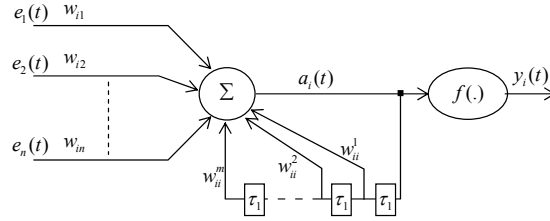


Figure 60. Architecture générale d'un réseau LRGF avec bouclage de l'activation. Les τ_1 représentent des retards unitaires.

L'évolution de la sortie de ce neurone dépend non seulement de ses entrées pondérées à l'instant t , mais également des valeurs retardées de son activation. Sa sortie $y_i(t)$ est définie par les équations :

$$y_i(t) = f(a_i(t))$$

$$a_i(t) = \sum_{j=1}^n w_{ij} e_j(t) + \sum_{q=1}^m w_{ii}^q a_i(t - q) \quad [14]$$

où $a_i(t)$ représente l'activation du neurone à l'instant t , $f(.)$ sa fonction d'activation, $e_j(t)$ son entrée, w_{ij} la valeur du poids de la connexion reliant le neurone i au neurone amont j et w_{ii}^k représente le poids de l'auto-connexion du neurone i avec un retard égal à $q\tau_1$. Cette auto-connexion permet au neurone de garder en mémoire une trace d'un certain passé de ses entrées $e_j(t)$: le neurone est donc doté d'une mémoire dynamique. Ce réseau peut être simplifié en utilisant qu'un seul retard au niveau de l'auto-connexion (Figure 61). L'activation du neurone devient alors :

$$a_i(t) = \sum_{j=1}^n (w_{ij} e_j(t)) + w_{ii} a_i(t-1) \quad [15]$$

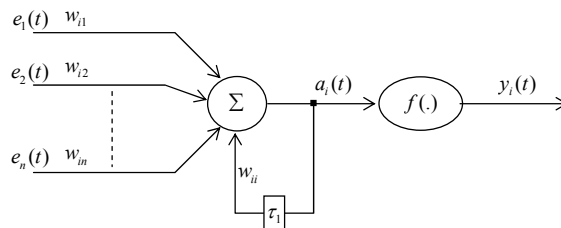


Figure 61. Architecture simplifiée d'un Réseau LRGF avec retour de l'activation

Lorsqu'il s'agit d'un neurone d'entrée au réseau, son activation peut être ramenée à l'équation suivante :

$$a_i(t) = \xi_i(t) + w_{ii} a_i(t-1) \quad [16]$$

L'étude des propriétés de ce type de neurone à retour local de l'activation, revient à étudier son comportement en l'absence de toute excitation extérieure ($\xi_i(t) = 0 \quad \forall t > 0$). En d'autres termes, ceci revient à étudier l'influence du retour local de l'activation sur la mémoire dynamique du neurone.

Concernant les points d'équilibre, deux cas sont possibles en fonction de w_{ii} :

- si $w_{ii} \neq 1$, le seul point d'équilibre dans ce cas est $a_0 = 0$,
- si $w_{ii} = 1$, tous les points $a \in \mathcal{R}$ sont des points d'équilibre.

Pour l'étude de la stabilité des points d'équilibre, nous obtenons :

- cas $w_{ii} \neq 1$, le point $a = a_0$ est un point d'équilibre instable si $w_{ii} > 1$.
- cas $-1 < w_{ii} < 1$, le point $a = a_0$ est un point d'équilibre stable.
- pour $w_{ii} < -1$, le point $a = a_0$ est un point d'équilibre instable.
- dans les cas limites : $w_{ii} = -1$ l'activation du neurone oscille entre les valeurs $\pm a$ (comportement astable), et reste stable en a pour le cas où $w_{ii} = +1$.

Comme pour les deux cas précédents, en se basant sur les propriétés des comportements d'oubli et de mémorisation, le neurone à retour local de l'activation possède deux comportements distincts en fonction de w_{ii} :

- Un comportement d'oubli si $|w_{ii}| < 1$,
- Un comportement de mémorisation si $|w_{ii}| \geq 1$.

Les figures suivantes illustrent ces deux comportements en fonction de w_{ii} :

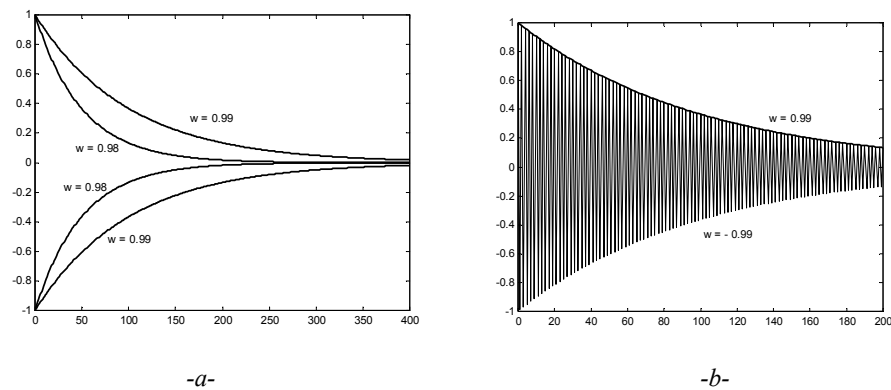


Figure 62. Comportement d'oubli. Convergence de l'activation du neurone vers le point d'équilibre stable $a = a_0$: a) sans oscillations pour des valeurs positives de l'auto-connexion, b) avec oscillations pour des valeurs négatives de l'auto-connexion.

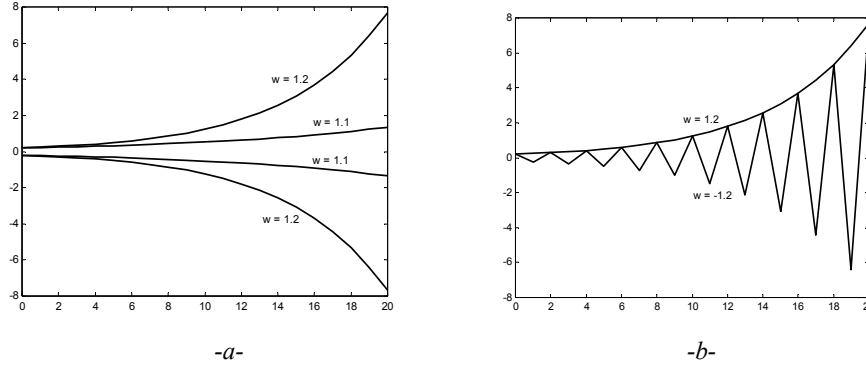


Figure 63. Comportement de mémorisation. Divergence de l'activation du neurone vers une valeur infinie a) sans oscillations, b) avec oscillations.

Enfin, l'étude de la longueur de la mémoire dynamique du neurone à retour local de l'activation se fait par l'analyse des variations $\partial y_i(t)$ de la sortie par rapport à celles de l'entrée $\partial \xi_i(t-q)$. D'après cette relation, on peut conclure que :

- le neurone à retour local de l'activation possède une mémoire de longueur finie si $|w_{ii}| < 1$:

$$\lim_{q \rightarrow \infty} \frac{\partial y_i(t)}{\partial \xi_i(t-q)} = \lim_{q \rightarrow \infty} f'(a_i(t)) w_{ii}^q = 0 \quad [17]$$

- une mémoire à longueur infinie si $|w_{ii}| > 1$

$$\lim_{q \rightarrow +\infty} \frac{\partial y_i(t)}{\partial \xi_i(t-q)} = \lim_{q \rightarrow +\infty} f'(a_i(t)) w_{ii}^q = +\infty \quad [18]$$

4.5.3 ANALYSE COMPARATIVE DES TROIS PRINCIPAUX TYPES DE MEMOIRES DYNAMIQUES DU RESEAU RRFR

Les architectures *LRGF* peuvent être divisées en trois représentations majeures : deux représentations à retour local de la sortie (le neurone *Frasconi-Gori-Soda* et le neurone *Poddar-Unnikrishnan*) et une représentation à retour local de l'activation (le neurone *Frasconi-Gori-Soda*). Ces trois neurones récurrents présentent des comportements proches.

- Les trois neurones possèdent un comportement d'oubli,
- Les trois neurones possèdent un comportement de mémorisation,
- Par contre, seul le neurone à retour local de l'activation possède dans certaines conditions une mémoire à longueur infinie. Les deux autres neurones à retour local de la sortie (neurone de *Frasconi-Gori-Soda* et celui de *Poddar-Unnikrishnan*) possèdent une mémoire à longueur limitée.

Le modèle du neurone bouclé⁴¹ de *Frasconi-Gori-Soda* représente un cas particulier du modèle de *Poddar-Unnikrishnan* ($\alpha = 1$). En utilisant les résultats de notre étude, nous pouvons conclure que le neurone bouclé *Frasconi-Gori-Soda* possède des performances meilleures que celles du neurone de *Poddar-Unnikrishnan*. En effet, la longueur de la mémoire du neurone bouclé de *Frasconi-Gori-Soda* est plus importante que celle du

⁴¹ Nous rappelons que le neurone bouclé représente le neurone à retour local de la sortie de *Frasconi-Gori-Soda*.

modèle de *Poddar-Unnikrishnan*. D'un autre côté, le neurone de *Poddar-Unnikrishnan* possède un paramètre supplémentaire à ajuster, ce qui peut dans certains cas, compliquer la procédure de réglage des paramètres optimum du neurone bouclé. On peut conclure alors que le modèle de *Frasconi-Gori-Soda* est plus performant que le modèle *Poddar-Unnikrishnan*.

Par contre, la comparaison entre les performances du neurone bouclé de *Frasconi-Gori-Soda* et le neurone à retour local de l'activation, n'est pas évidente. La Figure 64 présente une comparaison des longueurs de la mémoire des trois types de neurones localement récurrents. On peut voir que le neurone à retour local de l'activation possède la mémoire la plus longue pour des valeurs du poids de l'auto-connexion proches de 1.

Néanmoins, une différence majeure existe entre ces deux neurones localement récurrents : il s'agit de l'emplacement du point de retour de la récurrence. En effet, au niveau du neurone bouclé (retour local de la sortie) le point de retour se situe après la non-linéarité du neurone (après la fonction d'activation sigmoïde) ; au niveau du neurone à retour local de l'activation, le point de retour se situe avant la non-linéarité (avant la fonction d'activation sigmoïde). Ce détail comporte certainement des conséquences considérables aux niveaux des différences de comportements entre les deux mémoires dynamiques. Nous considérons par conséquent, le neurone bouclé de *Frasconi-Gori-Soda*, plus performant que le neurone à retour local de l'activation car la non-linéarité se situe à l'intérieur de la mémoire dynamique. Ceci le rend plus efficace dans la prise en compte de la dimension temporelle des systèmes non-linéaires.

Pour appuyer notre hypothèse, nous pouvons apporter l'argument concernant les critères de choix de la fonction d'activation des neurones afin d'obtenir un bon approximateur universel. En effet, dans (Tsoi *et al.*, 1994), les auteurs insistent sur l'avantage d'utiliser des fonctions d'activation non-linéaire pour la prise en compte de la non-linéarité des données d'entrée. L'emplacement du bouclage : avant la non-linéarité ou après le passage à la non-linéarité peut donc être un critère de choix décisif pour le type de mémoire dynamique.

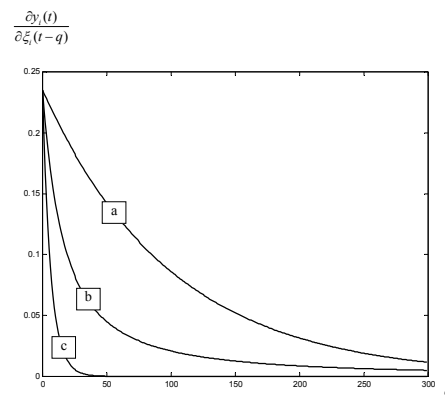


Figure 64. Comparaison de la longueur de la mémoire des trois architectures LRGF présentées précédemment : (a) neurone à retour local de l'activation, (b) neurone bouclé de *Frasconi-Gori-Soda*, (c) neurone de *Poddar-Unnikrishnan*.

4.6 CONCLUSION ET SYNTHÈSE

Dans ce chapitre, après une brève introduction des propriétés de base des réseaux de neurones et une présentation des principales contributions existantes dans le domaine de la

surveillance dynamique, nous avons introduit un état de l'art des représentations du temps dans les réseaux de neurones. Cet état de l'art met en évidence un domaine émergent de l'intelligence artificielle, dans lequel – tout d'abord pour des applications de reconnaissance de la parole – les scientifiques ont développé des outils neuronaux nouveaux, capables de prendre en compte le temps.

Dans ce contexte, les réseaux de neurones dynamiques récurrents apparaissent comme les mieux adaptés à la surveillance dynamique. Le principal désavantage de ces structures neuronales reste néanmoins la lourdeur des algorithmes d'apprentissage développés, ainsi que le manque de flexibilité des architectures neuronales temporelles proposées.

Afin de palier ces désavantages, nous avons proposé une nouvelle structure neuronale dynamique récurrente – appelée *Réseau Récurent à Fonctions de base Radiales* – faisant appel à une approche récurrente locale (LRGF) et aux fonctions gaussiennes. Ce réseau récurrent innovant présente des avantages remarquables :

En premier, l'utilisation des fonctions radiales permet de mettre en œuvre une surveillance efficace, en utilisant une approche de reconnaissance par généralisation locale, en adéquation avec les particularités de la surveillance industrielle.

En deuxième, la structure modulaire de la couche mémoire dynamique, basée sur les principes des architectures neuronales récurrentes de type LRGF, nous permet d'effectuer un paramétrage de l'acquisition de chaque élément de la signature du système surveillé, en fonction de sa dynamique et des caractéristiques de mémorisation ou d'oubli souhaitées. Ceci permet la mise en œuvre d'une surveillance dynamique flexible et facile à paramétrer.

Enfin, la structure de type RRFR permet l'apprentissage dynamique des nouveaux modes, caractéristique essentielle permettant la mise au point de systèmes de capitalisation des connaissances de surveillance, avec des débouchés intéressants dans le domaine de l'aide au diagnostic et au pronostic. Ce point constitue un atout fort, qui nous a permis de bénéficier du soutien d'une SSII bisontine Avensy Ingénierie (spécialiste du suivi de production à distance) pour le prototypage d'un système de surveillance dynamique et par la suite, de la participation de Cegelec SA (spécialistes de la maintenance) pour le prototypage d'un outil dynamique d'aide au diagnostic/pronostic.

Afin de pouvoir adapter toute la puissance et l'efficacité des techniques d'apprentissage flexibles du RFR à notre nouvel outil, le chapitre suivant est dédié à la validation et le test du RRFR, en considérant deux configurations : celle utilisant le neurone bouclé (neurone à retour local de la sortie) de Frasconi-Gori-Soda, et celle faisant appel au neurone à retour local de l'activation. Une étude comparative du RRFR et des principaux autres réseaux de neurones temporels sera présentée dans le domaine de la surveillance. Cette analyse du nouvel outil nous permet de poser de bases d'une exploitation industrielle du concept.

Chapitre 5 - Evaluation des performances, étude comparative et exploitation industrielle du RRFR

5.1 INTRODUCTION

Soucieux de l'aboutissement de nos recherches menées sur les réseaux de neurones dynamiques, une attention particulière a été accordée à l'étude de l'évaluation des performances ainsi qu'à l'étude comparative du Réseau Récurrent à Fonctions de base Radiales proposé, par rapport aux autres structures neuronales dynamiques récurrentes applicables en surveillance.

Ces étapes sont indispensables dans une optique d'exploitation industrielle de notre concept, exploitation initiée par un prototypage mis au point avec le soutien d'une SSII bisontine spécialisée dans le suivi de production à distance.

En premier, nous allons ainsi procéder à l'évaluation des performances du RRFR. Dans cette optique, nous abordons des applications-type de surveillance dynamique, en montrant la manière de mettre en œuvre l'outil et en mettant en évidence ces performances. Une technique d'apprentissage basée sur la méthode des k-moyennes est ainsi renforcée en utilisant l'algorithme fuzzy min-max.

Une étude comparative du RRFR par rapport à d'autres structures neuronales dynamiques fait l'objet d'une deuxième partie de ce chapitre. Suite à une analyse des outils neuronaux dynamique les plus efficaces en surveillance dynamique, cette étude comparative s'effectue en se basant sur le RRFR, le R2BF et le DGNN. Elle montre l'efficacité du RRFR dans les applications de type surveillance dynamique, en mettant en évidence les avantages liés au paramétrage, aux techniques d'apprentissage et à l'apprentissage dynamique propre à cet outil.

Dans le cadre du suivi à distance des unités de production, la SSII partenaire se trouve souvent confrontée à des situations de manque de données de surveillance, avec en plus, la contrainte d'une reconfiguration rapide des outils de suivi et de surveillance dynamiques mis en place chez leurs clients. Une étude de marché menée en collaboration a mis en évidence l'intérêt des industriels pour ce genre de concept, en sachant que pour l'utilisateur, l'outil neuronal dynamique peut et même doit rester parfaitement transparent. Ceci suppose une réflexion approfondie en ce qui concerne les modalités pratiques de mise en œuvre du concept, avec un rôle important accordé à la maîtrise de l'outil, sa (re)configuration, son paramétrage et son apprentissage.

L'exploitation industrielle du nouvel outil constitue par conséquent la dernière étape de ce chapitre. Elle décrit de manière très succincte le déroulement de la mise en œuvre d'une application de surveillance dynamique en temps réel d'un système, en utilisant l'apprentissage dynamique distant via le web.

5.2 EVALUATION DES PERFORMANCES DU RRFR

Les performances du *RRFR* (simplicité d'apprentissage, efficacité de la mémoire dynamique) sont évaluées sur trois applications-type de surveillance dynamique : reconnaissance de séquences temporelles, prédiction de séquences temporelles pour le pronostic et reproduction de séquences temporelles.

5.2.1 RECONNAISSANCE DE SEQUENCES TEMPORELLES

L'occurrence d'un événement externe peut être gardée dans la mémoire du *RRFR* soit temporairement pour un *comportement d'oubli* du neurone bouclé, soit indéfiniment pour un *comportement de mémorisation*. Ces deux types de comportement offrent des possibilités d'utilisation différentes, selon la problématique à résoudre. Pour une application d'apprentissage de séquences temporelles, où l'instant d'occurrence d'un événement représente une donnée importante du problème, on exploitera donc le comportement d'oubli du neurone récurrent.

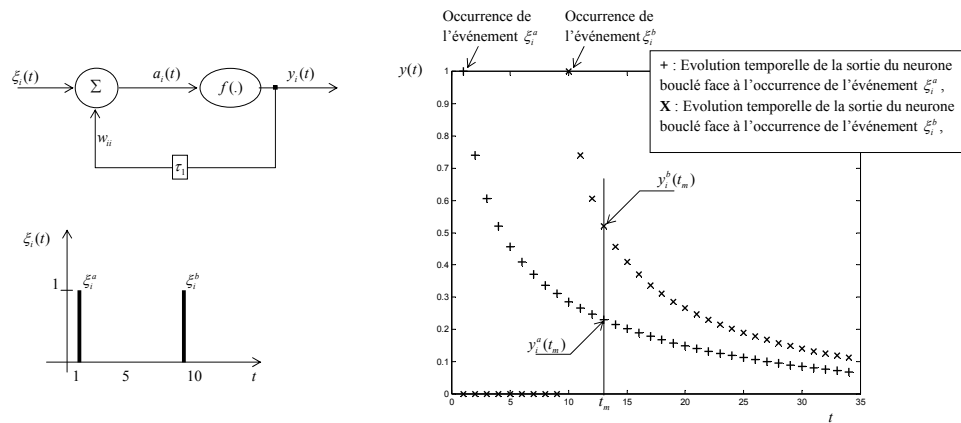


Figure 65. Evolution du neurone bouclé par rapport à deux événements distincts (se produisant à des instants différents).

En exploitant les propriétés de bijection de la fonction d'activation sigmoïde du neurone récurrent, nous avons démontré (**Zemouri, 2003**) que pour deux instants d'excitation différents, on obtient deux évolutions temporelles différentes du neurone récurrent. Par conséquent, l'évolution temporelle d'un neurone récurrent ayant un comportement d'oubli, dépend de l'instant qui caractérise l'occurrence de l'événement considéré.

Les séquences temporelles peuvent être divisées en deux catégories : séquences temporelles booléennes (surveillance des systèmes à événements discrets) et séquences temporelles réelles (surveillance d'un paramètre de type réel d'un équipement). Nous commençons donc l'évaluation du *RRFR* dans l'optique de ces deux cas d'utilisation.

a) APPRENTISSAGE DE SEQUENCES BOOLEENNES SIMPLES D'UN SYSTEME A EVENEMENTS DISCRETS

Une séquence booléenne simple est considérée comme étant une séquence dans laquelle chaque événement composant ne se produit qu'une seule fois. Chaque neurone récurrent de la mémoire dynamique ℓ_1 du réseau *RRFR* est dédié à un événement de la séquence. Le nombre de neurones récurrents de la mémoire dynamique est donc égal au nombre d'événements de la séquence (Figure 66-c).

Nous considérons le cas (Figure 66-a) d'un système de production avec deux machines M_1 , M_2 , deux zones de stockage et un convoyeur. Ce système à événements discrets est muni de capteurs C_i indiquant le passage des pièces dans chaque phase du processus. Chaque machine possède son temps de traitement nominal et chaque transfert, un temps nominal de convoyage. On peut alors prendre en compte un ensemble de perturbations sur le système, et tester les capacités du réseau *RRFR* à reconnaître ces situations de dysfonctionnement. Nous pouvons représenter ce type de système de production avec ses perturbations par un réseau de Petri temporisé à synchronisations externes (Daniel, 1995), (Racoceanu *et al.*, 2002) (Figure 66-b).

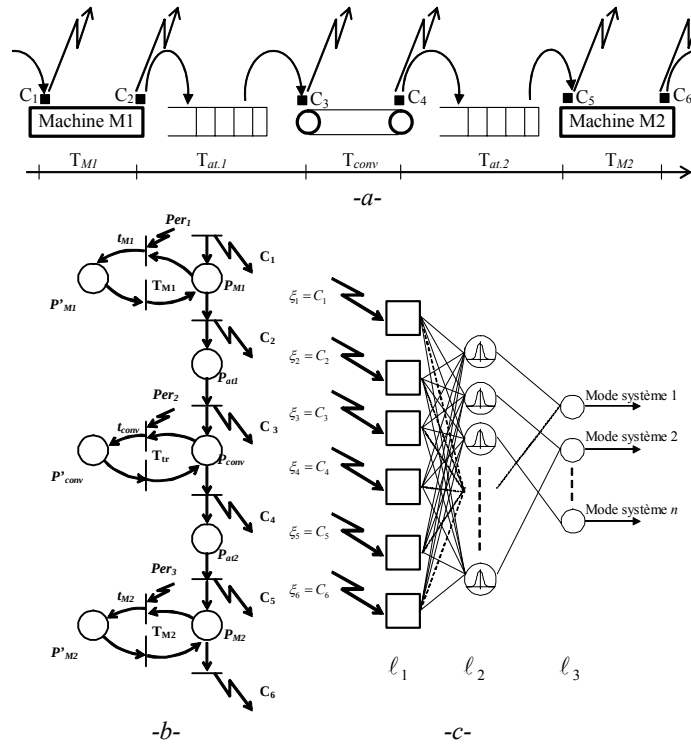


Figure 66. a) Système d'événements discrets étudié, b) Modélisation par réseau de Petri temporisés, c) Architecture du RRFR pour l'apprentissage des séquences du système.

Nous avons appliqué le réseau *RRFR* sur un ensemble de données d'apprentissage de séquences correspondant à différents modes de fonctionnement du système de la Figure 66-a (nominal, dégradés, pannes) (Zemouri, 2003). L'apprentissage se fait en utilisant l'algorithme *RCE* qui présente l'avantage majeur d'une convergence rapide.

La Figure 67 montre un exemple d'évolution de la mémoire dynamique ℓ_1 du réseau *RRFR* pour la séquence nominale. Le vecteur mémorisé par la mémoire statique (ℓ_2) est composé par les valeurs de sortie des neurones récurrents à la fin de la séquence. Les résultats obtenus nous permettent de tirer un certain nombre de conclusions :

- Le réseau *RRFR* a appris correctement les séquences de la base d'apprentissage,
- Le test sur l'ensemble des séquences proches de celles apprises de la base d'apprentissage, a été satisfaisant (degré de crédibilité moyen de 90 %). Toutes les séquences de la base de test ont été correctement reconnues par le réseau.
- Le réseau *RRFR* avec neurones à retour local de l'activation, a tendance à donner de fausses réponses, contrairement au réseau avec neurone bouclé.

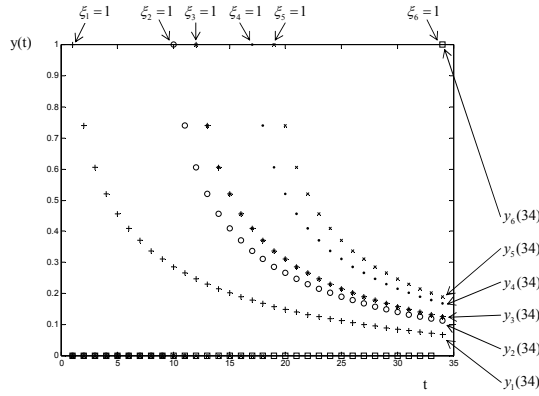


Figure 67. Evolution des sorties des neurones récurrents face aux événements associés à chaque neurone. Les valeurs de fin de séquence sont mémorisées par la mémoire statique.

b) RECONNAISSANCE DE SEQUENCES REELLES

b.1 Détection précoce d'un palier de dégradation et élimination des fausses alarmes

Contrairement aux séquences booléennes caractérisées par une succession d'événements discrets et finis, une séquence réelle est une série temporelle à paramètres réels. Dans une séquence réelle, on s'intéresse plutôt à une succession de valeurs réelles à chaque période d'échantillonnage :

$$S_k = \{u(t), u(t+1), u(t+2), \dots, u(t+n)\} \mid n \in \mathbb{N}, u(t) \in \mathbb{R} \quad [19]$$

En partant de cette forme, nous avons mis en évidence les capacités du réseau *RRFR* à apprendre à reconnaître les différents *paliers de dégradation* ainsi qu'à distinguer les *pics de fausses alarmes* des paliers de dégradation, aspects essentiels pour l'utilisation du *RRFR* en surveillance dynamique.

Considérons le point x_a appris du palier d'apprentissage de pente 0,005 correspondant à une séquence réelle S_k (Figure 68-b). Nous testons les capacités du réseau *RRFR* à distinguer ce point d'autres points faisant partie des paliers de test de pentes différentes. La longueur de la mémoire dynamique varie entre un seul neurone récurrent et un ensemble d'un neurone linéaire et une cascade de jusqu'à 6 neurones récurrents (Figure 68-a) (au-delà, l'augmentation de la qualité de la classification des paliers ne justifie pas le surcoût en terme de temps de calcul).

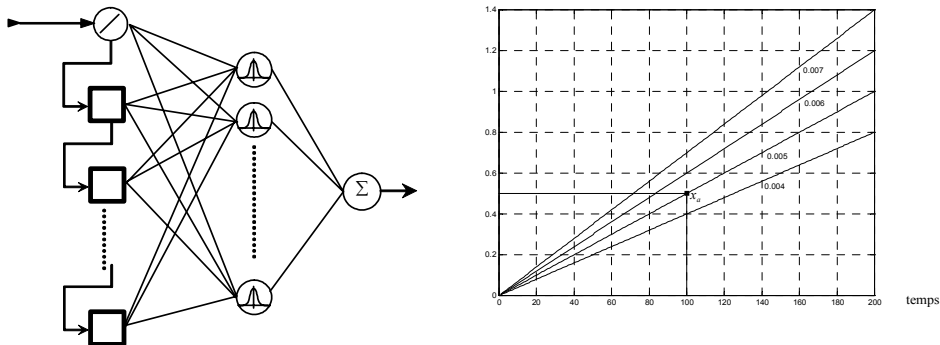


Figure 68. a – Réseau *RRFR* avec mémoire dynamique à un neurone linéaire et une cascade de neurones bouclés, b – Apprentissage d'un palier de dégradation

La Figure 69 résume les résultats des calculs de distance obtenus. La distance euclidienne est proche de zéro pour les droites proches de celle apprise, et tend à augmenter pour celles qui s'éloignent de la droite apprise. La mémoire dynamique du réseau est donc bien capable de caractériser une séquence temporelle réelle.

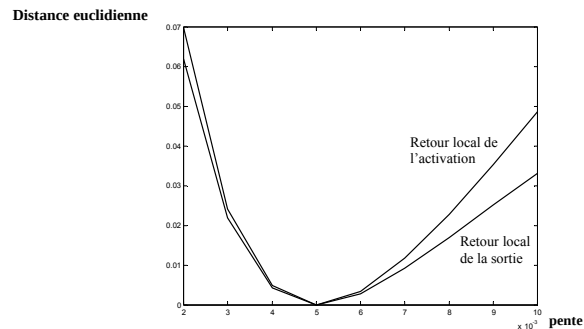


Figure 69. Comparaison entre la sortie de la mémoire dynamique avec neurones bouclés et celle avec neurone à retour local de l'activation.

Le dimensionnement de la mémoire dynamique est un paramètre important pour la caractérisation de la séquence réelle (**Zemouri et al., 2003b**). Plus on augmente la cascade de neurones bouclés, plus le vecteur de sortie de la mémoire dynamique est représentatif de la séquence réelle. Néanmoins, on commence à obtenir de résultats discriminatoires à partir d'une mémoire dynamique à un neurone linéaire et un neurone récurrent. On peut ainsi détecter une dégradation aussi minime soit elle, alors que l'équipement est toujours dans sa zone de bon fonctionnement.

Après avoir testé les capacités de la mémoire dynamique à caractériser un palier de dégradation, nous avons montré que la mémoire dynamique du réseau RRFR est capable de réagir différemment face à un pic de *fausse alarme* et à un *palier de dégradation* (**Zemouri et al., 2002c**). Comme pour le cas précédent, nous allons faire des tests de calcul de distance euclidienne entre le point x_a de la droite de pente 0.005 (palier de dégradation) et le même point, mais cette fois-ci faisant partie d'un pic de fausse alarme (voir Figure 70). Le tableau ci-dessous caractérise ce calcul de distance pour chacune des deux mémoires dynamiques (retour local de la sortie et de l'activation).

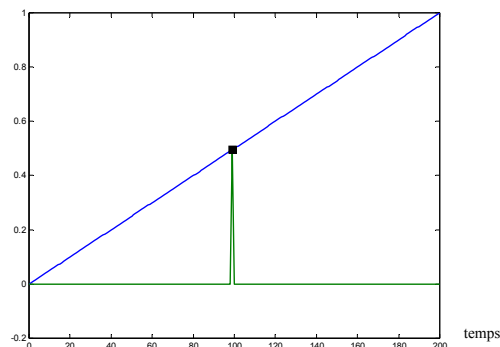


Figure 70. Dissociation entre un changement brusque, et un palier de dégradation.

La réponse du neurone récurrent est différente pour le même point physique appartenant à un palier de dégradation et à un pic de fausse alarme. Le neurone récurrent agit donc comme un filtre passe bas, en éliminant implicitement les hautes fréquences.

	Type de mémoire dynamique	
	Retour Local de la Sortie	Retour Local Activation
1 N Récurrent	0.0821	0.1780
1NL + 1NR	0.0821	0.1780
1NL + 2NR	0.1361	0.2754
1NL + 3NR	0.1650	0.3096
1NL + 4NR	0.1753	0.3179
1NL + 5NR	0.1776	0.3194
1NL + 6NR	0.1780	0.3196

Tableau 2. Calcul de distance euclidienne entre un palier de dégradation et un pic de fausse alarme (NL = Neurone Linéaire, NR = Neurone Récurrent).

Nous avons validé cette propriété du neurone récurrent sur le filtrage des bruits d'acquisition de la vitesse de rotation d'un moteur électrique (Figure 71).

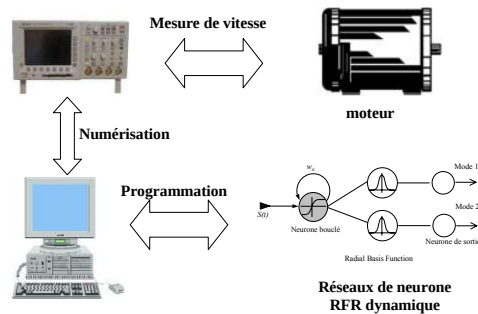
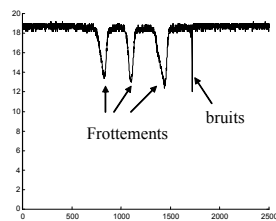
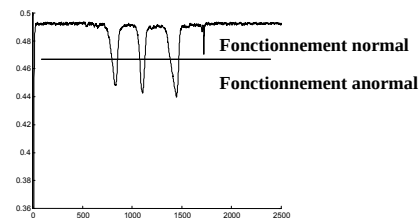


Figure 71. Schéma d'application de la surveillance d'un moteur.

Des perturbations ont été provoquées au niveau du capteur de vitesse d'une part, et des frottements continus au niveau de l'axe de rotation du moteur. La Figure 72 montre d'un côté l'acquisition de vitesse (entrée du neurone récurrent) et d'un autre côté le filtrage effectué par le neurone récurrent (sa sortie).



- a - mesure de vitesse



- b - réponse du neurone bouclé

Figure 72. Mesure de la vitesse de rotation et réponse du neurone bouclé avec les deux types de perturbations : frottements et bruits de mesures. Mise en évidence de la fausse alarme (bruit de mesure) malgré son amplitude importante.

b.2 Détection du type de collision d'un bras de robot

Dans la catégorie des test sur des séquences réelles, une validation supplémentaire a été apportée dans le domaine de la reconnaissance dynamique de signaux capteurs (**Zemouri et al., 2003a**). Cette application a été construite à partir des données réelles du benchmark « Robot Execution Failures »⁴² (Camarinha-Matos et al., 1996). Le but de cette application est de reconnaître le type de collision d'un bras de robot à partir d'une acquisition de signaux capteurs (Figure 73). Le bras du robot est muni de trois capteurs de force (F_x , F_y ,

⁴² Disponibles sur le serveur de l'Université de Californie - The UCI KDD Archive [<http://kdd.ics.uci.edu>]. Irvine, CA: University of California, Department of Information and Computer Science

F_z). Quatre types de collisions sont susceptibles de se produire (Figure 73) : *collision frontale*, *collision par derrière*, *collision à gauche* et *collision à droite*. Chaque type de collision est caractérisé par une évolution temporelle des trois signaux de mesures. La Figure 74 présente un échantillon de chaque type de collision. Les informations des trois capteurs de force constituent les trois entrées du réseau *RRFR*. Après une phase d'apprentissage, le réseau *RRFR* devra reconnaître le type de collision, caractérisé par une signature des trois mesures de force.

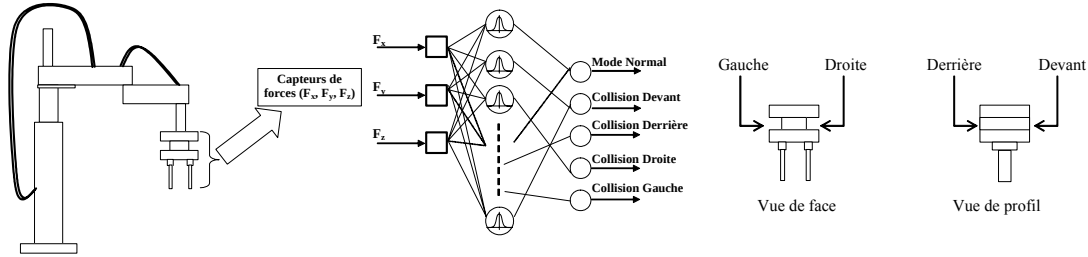


Figure 73. Application du réseau *RRFR* pour la surveillance d'un bras de robot. Différents types de collisions possibles : frontale, par derrière, à gauche ou à droite.

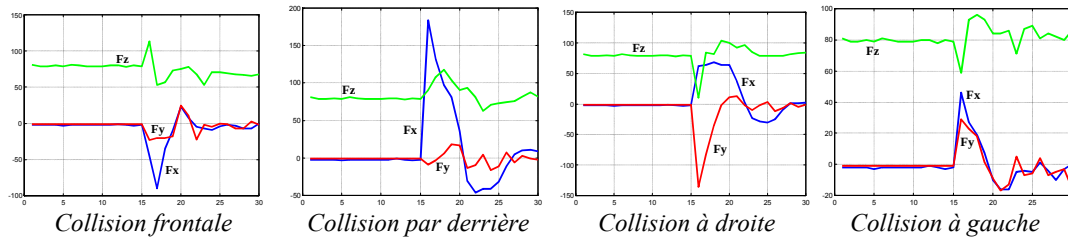


Figure 74. Réponses des capteurs de force (F_x , F_y et F_z) pour chaque type de collision.

Nous avons évalué les performances du réseau *RRFR* avec les deux types de mémoires dynamiques, en les comparant avec celles du réseau *RFR* classique. La taille ainsi que les paramètres de la mémoire statique du *RFR*, ont été déterminés par l'algorithme d'apprentissage *DDA* qui permet d'établir un traitement incrémental. Les résultats des tests obtenus par le réseau *RRFR* et *RFR* sont présentés dans le tableau ci-dessous :

Types de collisions	Taux de reconnaissance		
	Mémoire dynamique avec Neurone bouclé	Mémoire dynamique avec neurone à retour de l'activation	Absence de mémoire dynamique (<i>RFR</i> classique)
Collision Frontale	60 %	40 %	20 %
Collision par derrière	16,5 %	50 %	0 %
Collision à droite	75 %	50 %	0 %
Collision à gauche	25 %	50 %	0 %
Taux global de reconnaissance	44 %	47,5 %	5 %

Tableau 3. Résultats obtenus (*RRFR* avec les deux types de mémoire et *RFR*).

Chaque colonne du Tableau 3 représente les taux de reconnaissance obtenus par les trois réseaux testés (*RRFR* avec les deux types de mémoires dynamiques et *RFR* statique). Le pourcentage est donné par rapport aux échantillons de test pour chaque type de collision. Nous remarquons que les performances du *RRFR* sont nettement supérieures au *RFR*. L'absence de mémoire dynamique rend le *RFR* incapable de prendre en compte

l'évolution des signaux. Les temps d'apprentissage et de traitement sont sensiblement proches pour les trois réseaux (≈ 0.04 secondes⁴³).

5.2.2 LA PREDICTION TEMPORELLE POUR LE PRONOSTIC

La problématique de la prédiction temporelle représente un vrai défi dans plusieurs domaines (prévisions météorologiques, financières, ...). En surveillance, elle permet de connaître les évolutions futures d'un paramètre surveillé, afin d'anticiper sur les actions à entreprendre.

Une application de prédiction temporelle est différente de la reconnaissance de séquences temporelles. En effet, la prédiction temporelle est considérée comme une application d'approximation de fonctions et non de classification. Les variables de sortie du réseau de neurones sont donc des variables de type réel et non catégoriel.

Pour une application de type approximation de fonction, le processus d'apprentissage du *RRFR* comporte deux phases :

- Phase non supervisée - calcul des paramètres des neurones gaussiens,
- Phase supervisée - calcul du vecteur de pondération de la couche de sortie.

Plusieurs techniques existent pour la première phase du calcul. Les algorithmes d'apprentissage de type heuristique⁴⁴ *RCE* et *DDA*, peuvent présenter, sous certaines conditions, une bonne solution pour calculer les gaussiennes. Ces techniques présentent certains désavantages comme leur sensibilité aux paramètres de l'apprentissage. Les centres des gaussiennes ne sont pas déterminés par rapport à un critère à minimiser, comme pour la technique des *k-moyennes*. En effet, l'algorithme des *k-moyennes* détermine les centres des gaussiennes en minimisant les moyennes des distances quadratiques. Cette technique offre de meilleurs résultats que les techniques heuristiques, mais présente également quelques inconvénients. Nous proposons une version améliorée de l'algorithme des *k-moyennes*, qui pallie aux faiblesses de la version classique.

Tous ces points seront traités et mis en évidence à travers deux exemples-type de prédiction de séries temporelles pour des applications de pronostic industriel : la série temporelle de *Mackey-Glass* et la prédiction d'une concentration de sortie en CO_2 d'un four à gaz (base de donnée *Box-Jenkins*).

a) LA SERIE TEMPORELLE DE MACKEY-GLASS

Soit la série temporelle de Mackey-Glass définie par l'équation différentielle à délai (Mackey *et al.*, 1977):

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-\Delta)}{1+x(t-\Delta)^{10}} \quad [20]$$

L'apprentissage ainsi que le test de prédiction du réseau de neurones se fera sur un horizon de six pas : $x(t+6)$ (Figure 75). Le réseau *RRFR* testé dans cette partie comporte une cascade d'un neurone linéaire et un neurone récurrent.

⁴³ Traitement effectué sur un processeur de 1.2Ghz de fréquence d'horloge.

⁴⁴ Dites aussi techniques incrémentales.

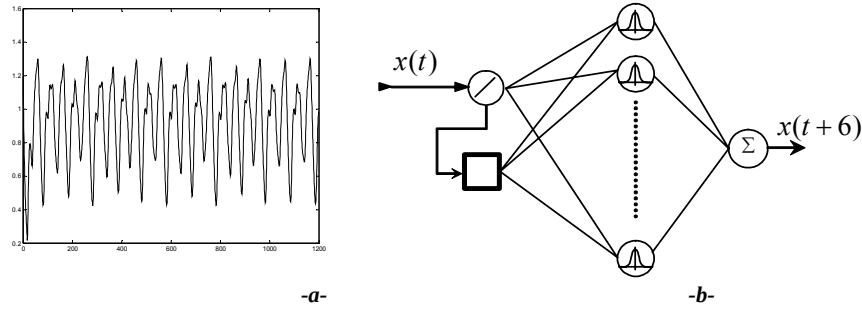


Figure 75. a) Série Mackey-Glass, b) RRFR pour la prédiction de $x(t+6)$

a.1 Techniques RCE et DDA

Ces deux techniques sont généralement dédiées à des problématiques de classification. On peut néanmoins, sous certaines hypothèses, les appliquer à l'approximation de fonction. En effet, dans ce cas, la notion de *classe d'appartenance des données* n'existe pas. Après de différents test réalisés en utilisant les algorithmes DDA et RCE (Zemouri, 2003) c'est le RCE qui se montre le plus efficace (Figure 76).

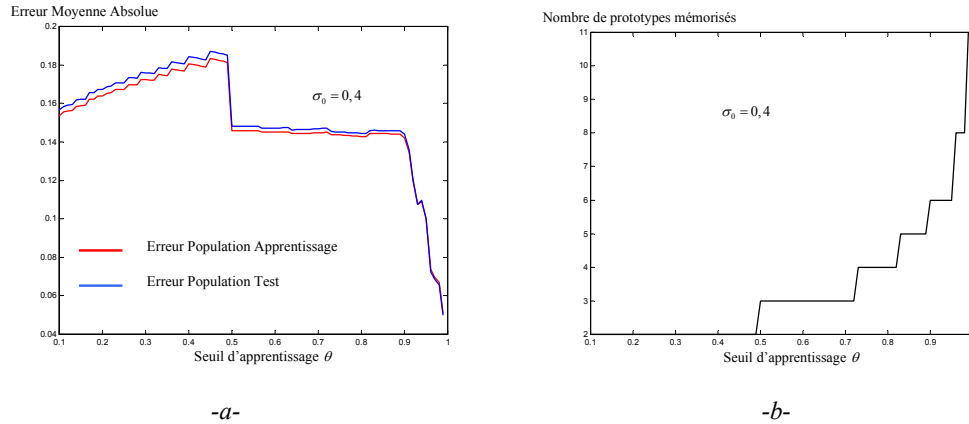


Figure 76. Résultats de l'application de l'algorithme RCE sur le problème de la série Mackey-Glass : a - erreurs d'estimation ; b - nombre de prototypes créés

L'algorithme RCE étant le plus efficace, nous avons procédé à un ensemble de tests de comparaison entre les performances du réseau RRFR (apprentissage utilisant l'algorithme RCE) avec les deux types de mémoires dynamiques et le réseau RFR statique. Le bilan de ces essais est présenté dans la figure suivante :

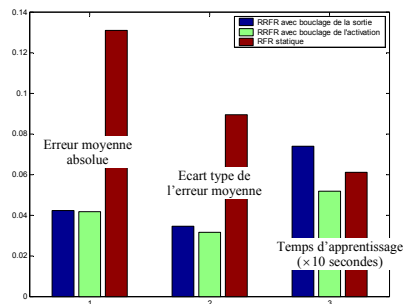


Figure 77. Comparaison des trois réseaux après apprentissage par l'algorithme RCE sur le problème de prédiction de la série MackeyGlass.

Malgré les résultats obtenus, les deux techniques d'apprentissage heuristiques présentent néanmoins quelques faiblesses. La plus importante est que leurs performances sont étroitement liées aux paramètres que l'expert doit définir. En ce qui concerne la valeur initiale (σ_0) du rayon d'influence des prototypes – par exemple - *aucune technique d'initialisation* (à part le tâtonnement) *n'est disponible actuellement dans la littérature*.

Un autre paramètre qui influe grandement sur la qualité des résultats obtenus, est le seuil θ du degré d'appartenance. Les techniques heuristiques souffrent aussi d'une certaine dépendance à l'ordre dans lequel sont présentées les données pour l'apprentissage. Nous ne pouvons donc pas garantir l'optimalité des résultats obtenus.

On peut néanmoins attribuer deux avantages non négligeables à la technique *RCE* : une bonne capacité de généralisation procurée au réseau *RRFR* (erreur sur population de test sensiblement proche de celle obtenue sur la population d'apprentissage), ainsi que la rapidité de convergence de l'algorithme.

a.2 Algorithme des k-moyennes

Les faiblesses des deux techniques abordées précédemment peuvent être contournées avec d'autres techniques plus robustes. Une des plus connues est la technique de partitionnement appelée technique des *k-moyennes*⁴⁵ qui a pour principe de déterminer le centre μ_j d'un nuage de points :

$$\mu_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \chi_j} \mathbf{x} \quad [21]$$

Le centre μ_j représente le point qui minimisera la distance quadratique par rapport à tous les autres points ($\mathbf{x}$) du nuage (χ_j). Cette distance quadratique minimum sera la variance du prototype (*carré du rayon d'influence*) :

$$\sigma_j^2 = \frac{1}{N_j} \sum_{\mathbf{x} \in \chi_j} (\mathbf{x} - \mu_j)(\mathbf{x} - \mu_j)^t \quad [22]$$

Par rapport aux techniques incrémentales, cet algorithme présente l'avantage de déterminer les prototypes les plus représentatifs d'un nuage de points. Comme pour les techniques *RCE* et *DDA*, l'algorithme des *k-moyennes* est généralement utilisé dans des problématiques de classification. Pour pouvoir l'utiliser en approximation de fonctions, il faudrait supposer que tous les points de la base d'apprentissage appartiennent au même groupe de centres (même classe – par analogie avec la classification).

Après quelques tests, nous avons confirmé le fait que l'algorithme des *k-moyennes* donne de meilleures prédictions que les deux techniques heuristiques. Par contre, cet algorithme possède quelques faiblesses :

Problème des nuages vides

Des situations peuvent apparaître où le nuage de points associé à un prototype soit vide. Ceci engendre un gros souci pour calculer le rayon d'influence du prototype. Trois solutions sont alors possibles pour résoudre ce problème :

- affecter une valeur au rayon du nuage vide : quel critère pour définir cette valeur ?
- éliminer ce point de la liste des *k* centres,

⁴⁵ Appelée également technique des centres mobiles. Son appellation anglophone est « *k-means clustering algorithm* »

- calculer le rayon d'influence en essayant d'utiliser le seuil de la technique RCE.

Suite à une série de tests de comparaisons entre la technique de la variance (équation [22]) et la technique du seuil θ de l'algorithme RCE, nous avons remarqué que le calcul des rayons d'influence par la technique du seuil θ de l'algorithme RCE donne de meilleures performances. Nous avons donc opté pour cette procédure pour calculer les rayons d'influence de chaque prototype. Nous appelons cette méthode - version simple de l'algorithme des k-moyennes.

✚ Problème du choix du nombre de prototypes

Le deuxième point de l'algorithme qui influence les performances du réseau de neurones est le choix du nombre k de centres (prototypes) que le réseau devra mémoriser. La Figure 78 montre l'influence du choix de ce paramètre sur les performances de prédiction du réseau RRFR. Trois zones sont mises en évidence : une zone de sous-apprentissage (1), une zone de bonne généralisation (2) et une zone de sur apprentissage (3). Il faudrait donc initialiser le nombre k pour que le réseau de neurones se trouve dans la zone (2). Il n'existe pas de méthodes formelles pour initialiser a priori le nombre de centres au début de l'algorithme (Chang *et al.*, 2001).

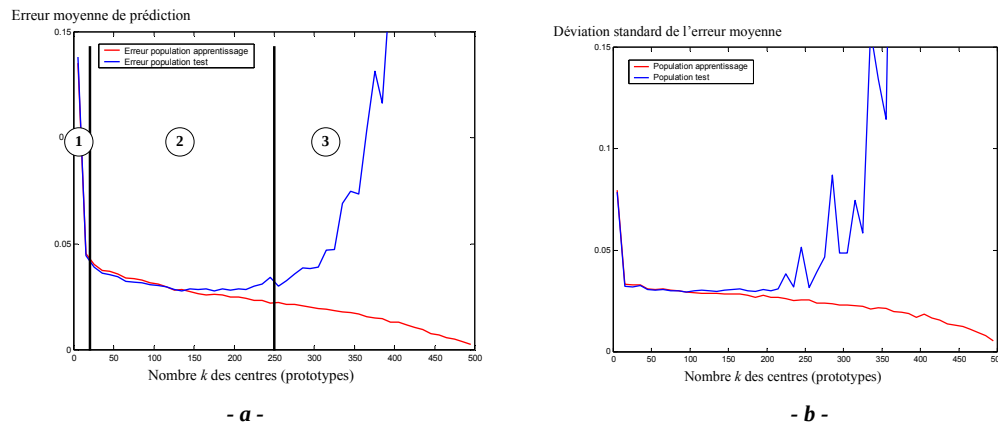


Figure 78. Influence du paramètre k sur les performances de prédiction du réseau RRFR : a) erreur moyenne de prédiction, b) déviation standard de l'erreur moyenne de prédiction.

✚ Problème d'initialisation des centres

Au tout début de l'algorithme d'apprentissage, les k centres sont initialisés généralement d'une façon aléatoire. Cette initialisation rend l'algorithme très instable. Nous proposons donc par la suite une version améliorée de l'algorithme des k-moyennes qui tente de contourner ces inconvénients de la version classique.

a.3 Proposition d'un algorithme d'apprentissage k-moyennes renforcé

Le problème d'initialisation des centres peut être résolu par une technique appelée *Fuzzy Min-Max* (Simpson, 1992, 1993), (Chang *et al.*, 2001). Cette technique permet de déterminer le nombre k des centres et leur valeur initiale d'une manière itérative. L'algorithme des *k-moyennes* peut être ainsi renforcé afin de converger vers le minimum de la somme des erreurs quadratiques entre les vecteurs d'entrée et les k centres. Durant cette phase d'initialisation, des hyper-cubes à n dimensions sont créés. Le degré d'appartenance d'un point à chaque hyper-cube est donné par la fonction d'appartenance :

$$H_j(x, v_j, u_j) = \frac{1}{n} \sum_{i=1}^n [1 - f(x_i - u_{ji}) - f(v_{ji} - x_i)] \quad [23]$$

où la fonction floue f est définie par l'expression suivante :

$$f(x) = \begin{cases} 1, & x > \eta \\ x / \eta, & \text{si } 0 \leq x \leq \eta \\ 0, & x < 0 \end{cases} \quad [24]$$

avec :

- H_j le degré d'appartenance (entre 0 et 1) d'un point x à l'hyper-cube j ;
- x_i la $i^{\text{ème}}$ dimension du vecteur d'entrée x ;
- u_{ji} la valeur de la $i^{\text{ème}}$ dimension des points maximums et minimums
- et v_{ji} celle du $j^{\text{ème}}$ hyper-cube.

Le paramètre η est appelé sensibilité de l'hyper-cube. Sa valeur détermine la pente de la décroissance du degré d'appartenance H_j (équation [23]) d'un point en fonction de son éloignement par rapport à l'hyper-cube j .

Les auteurs de cet algorithme ne donnent aucune méthode formelle pour initialiser ce paramètre de sensibilité du degré d'appartenance. Le seul critère donné est d'initialiser ce paramètre de telle sorte à minimiser les recouvrements entre hyper-cubes (c'est-à-dire donner de petites valeurs à η). Afin de respecter ce critère et de vouloir garder un certain aspect flou au degré d'appartenance, nous proposons l'initialisation suivante de η :

$$\eta = \min_i \frac{\max_{\mathbf{x}_j \in \mathcal{X}}(x_{ji}) - \min_{\mathbf{x}_j \in \mathcal{X}}(x_{ji})}{2 \times (N - 1)} \quad [25]$$

avec

- x_{ji} la $i^{\text{ème}}$ dimension du vecteur d'entrée $\mathbf{x}_j$ du nuage de points $\mathcal{X}$;
- N le nombre total de points du nuage $\mathcal{X}$.

L'algorithme *Fuzzy Min-Max* possède trois phases :

- 1. Initialisation des valeurs maximales et minimales du premier hyper-cube par le premier point présenté au réseau.
- 2. Calcul du degré d'appartenance de chaque point d'entrée par l'équation [23].
- 3. L'extension de l'hyper-cube ayant la plus grande fonction d'appartenance selon la condition suivante :

$$\sum_{i=1}^n (\max(u_{ji}, x_i) - \min(v_{ji}, x_i)) \leq n\theta \quad [26]$$

où θ représente un paramètre de contrôle de la création des nouveaux hyper-cubes.

- 4. Si aucun hyper-cube ne peut être élargi (condition de l'équation [26] n'est pas respectée), un nouvel hyper-cube contenant le nouveau point est créé.

Après avoir présenté au réseau l'ensemble des données d'apprentissage, un certain nombre d'hyper-cubes est créé en fonction de la valeur du paramètre θ . Ce dernier est le

seul paramètre que l'expert doit ajuster. On calcule alors les centres de chaque hyper-cube. Les k centres de l'algorithme des k -moyennes sont ainsi initialisés.

A partir de ces résultats, nous proposons la version renforcée de l'algorithme d'apprentissage des k -moyennes suivant :

1. Initialisation du paramètre θ de l'algorithme Fuzzy Min-Max [26],
2. Initialisation des k centres avec une itération de l'algorithme Fuzzy Min-Max,
3. Faire jusqu'à pas de changement :
 - 3.1. Faire jusqu'à pas de changement :
 - 3.1.1. affectation de chaque point $\mathbf{x} \in \mathcal{X}$ au centre μ_i le plus proche,
 - 3.1.2. calcul des nouveaux centres $\{\mu_1, \mu_2, \dots, \mu_k\}$ de chaque nuage de points avec l'équation [21],
 - 3.2. Eliminer les centres des nuages vides
4. Calcul des rayons de chaque prototype avec la technique RCE.

Tableau 4. Etapes de l'algorithme proposé.

Le seul paramètre qu'il faut donc ajuster avec cette version renforcée est le paramètre θ de l'algorithme Fuzzy Min-Max (équation [26]). Pour étudier la sensibilité de cette version par rapport aux variations de ce paramètre, nous avons testé son influence sur la détermination du nombre de prototypes créés et mémorisés par le réseau de neurones :

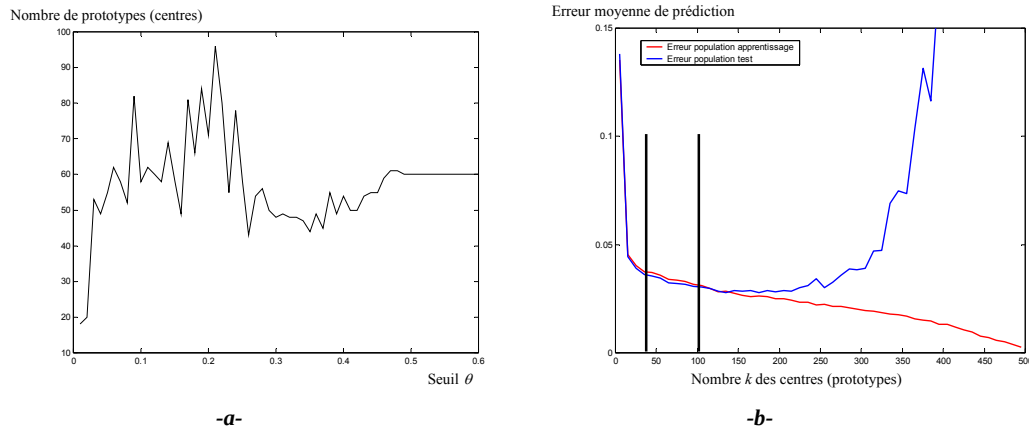


Figure 79. Influence du paramètre θ sur les performances de l'algorithme : a) Nombre de prototypes créés en fonction des variations du seuil θ , b) limites de la zone des prototypes créés après convergence de l'algorithme des k -moyennes renforcé.

La Figure 79.a montre que quelles que soient les variations du seuil θ , le nombre de prototypes créés se situe bien dans la zone de bonne généralisation (Figure 79-b). L'algorithme force le réseau à converger vers une zone de bonne généralisation, en évitant ainsi les zones de sur-apprentissage et de sous-apprentissage. Cet algorithme est donc plus robuste dans la phase de paramétrage.

Nous avons par ailleurs établi une série de tests comparatifs entre les performances obtenues avec l'algorithme proposé et la version simple des k -moyennes. Les résultats de ces tests montrent dans tous les cas une erreur moyenne de prédiction (jusqu'à 5 %) inférieure dans le cas de l'utilisation de l'algorithme des k -moyennes renforcé.

Pour mieux expliquer les résultats obtenus et ainsi mettre davantage en évidence la différence entre la version simple de l'algorithme des k -moyennes et la version renforcée proposée, nous avons effectué des tests de comparaison sur un exemple illustratif présenté

sur la Figure 80. La figure (a) montre une initialisation des centres avec une itération de l'algorithme Fuzzy Min-Max. Les centres de nuages vides seront éliminés par notre algorithme. En fin de calcul, les centres obtenus sont ceux représentés par la figure (b). L'algorithme ne garde ainsi que les centres les plus représentatifs d'une population importante de points. Notons que le résultat obtenu est le même à chaque exécution de l'algorithme (stabilité de l'algorithme), contrairement à la version simple de l'algorithme des k-moyennes qui est assez instable. En effet, la figure (c) représente un exemple de résultat obtenu avec la version simple des k-moyennes.

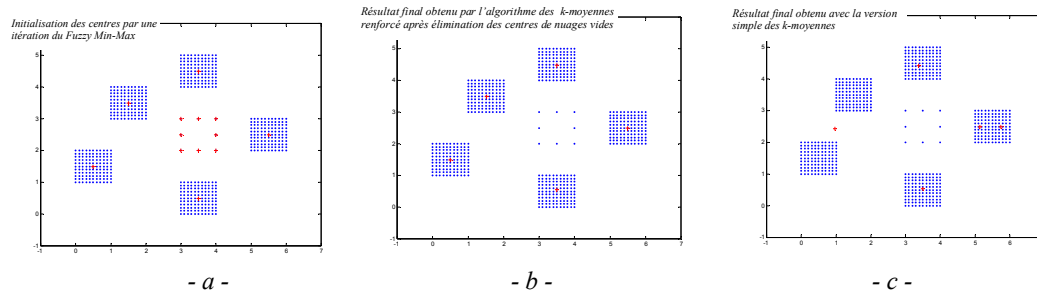


Figure 80. Stabilité du calcul des centres avec la version renforcée des k-moyennes (a) et (b), par rapport à version simple de l'algorithme (c).

Ces tests nous permettent de conclure sur trois remarques :

- Stabilité des résultats obtenus avec l'algorithme des k-moyennes renforcé.
- Le nombre k calculé après la convergence de l'algorithme des k-moyennes renforcé permet au RRFR de se trouver dans la zone de bonne généralisation (Figure 79).
- Le résultat vers lequel converge le RRFR se trouve parmi les meilleurs résultats pouvant être obtenus par la version simple des k-moyennes.

A partir de ces constats, on peut conclure que l'algorithme des k-moyennes renforcé converge vers les meilleurs résultats, avec d'avantage de robustesse.

b) APPLICATION ET VALIDATION DU RRFR SUR LA PREDICTION DES PARAMETRES D'UN FOUR A GAZ

Nous avons procédé à une validation de ces résultats sur une application de prédiction industrielle avec des données réelles : le benchmark d'un four à gaz – *Box-Jenkins Furnace Data*⁴⁶. Le but de cette application est de prédire la concentration de sortie en CO_2 $y(t+1)$ à partir de la sortie $y(t)$ et du débit de gaz en entrée $u(t)$ (Figure 81).

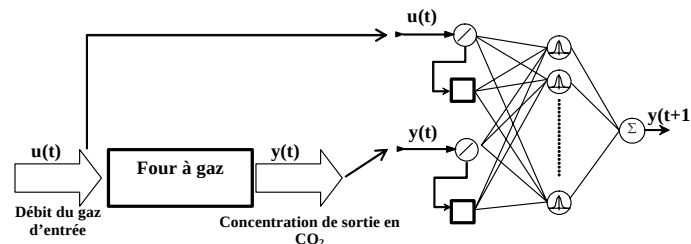


Figure 81. Surveillance d'un four à gaz par le réseau de neurones RRFR.

⁴⁶ Benchmark disponible sur le serveur du groupe de travail IEEE Working Group on Data Modeling Benchmarks : <http://neural.cs.nthu.edu.tw/jang/benchmark>

Le réseau *RRFR* utilisé comporte un neurone linéaire et un neurone récurrent pour chacune des deux variables d'entrée. La Figure 82 montre l'évolution des deux signaux avec les deux groupes $y(t)$ et $u(t)$ choisis pour l'apprentissage et le test.

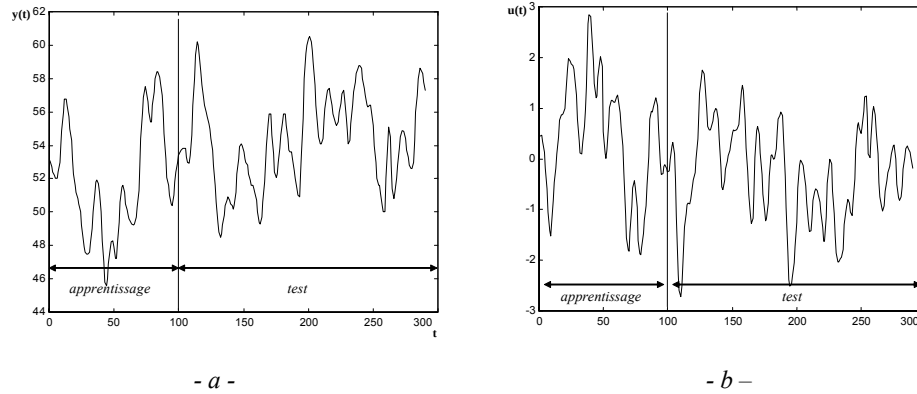


Figure 82. a) Concentration du CO_2 en sortie du four à gaz, b) Débit du gaz en entrée.

L'initialisation par la technique *Fuzzy Min-Max* permet de garder uniquement les centres les plus représentatifs d'un nuage de données. En appliquant toutes les étapes de l'algorithme des *k-moyennes* renforcé, on obtient un nombre de centres k tel que le réseau de neurones se trouve dans la zone de bonne généralisation, et ceci quelles que soient les variations du paramètre d'apprentissage (Figure 83). Comme pour l'exemple de la série temporelle Mackey-Glass, le réseau *RRFR* converge vers la zone de bonne généralisation. On évite ainsi les deux zones de sur-apprentissage et de sous-apprentissage.

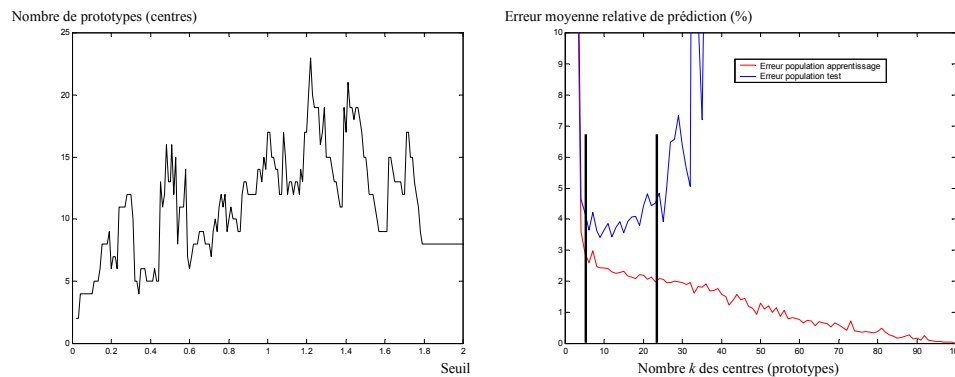


Figure 83. Nombre de prototypes obtenus en fonction des variations du seuil θ de l'algorithme Fuzzy Min Max.

Des comparaisons ont été effectuées sur une centaine de tests entre la version renforcée de l'algorithme des *k-moyennes* et la version classique. On remarque que, non seulement le résultat obtenu par la version améliorée est stable, mais est aussi meilleur que la moyenne des 100 itérations de la version simple de l'algorithme des *k-moyennes*.

Des tests de comparaison entre les performances de deux types de mémoires dynamiques du *RRFR* et le *RFR* statique ont démontré la supériorité du *RRFR* (Figure 84). Par ailleurs, les performances obtenues par le *RRFR* avec mémoire dynamique à retour local de la sortie sont légèrement supérieures au retour local de l'activation.

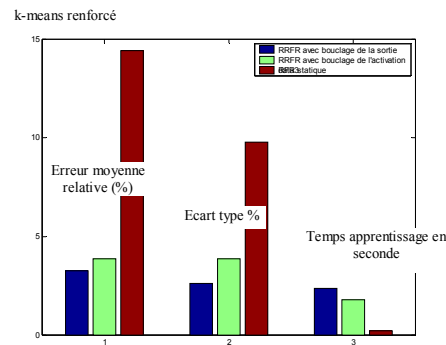


Figure 84. Comparaison des performances entre les deux types de mémoires dynamiques et le réseau RFR statique sur une application de prédiction

5.2.3 LA REPRODUCTION DE SEQUENCES

Le troisième type de test que nous avons appliqué au *RRFR*, concerne la reproduction de séquences temporelles. En d'autres termes, après avoir appris au réseau de neurones une trajectoire temporelle bien définie, ce dernier devra pouvoir la reproduire librement, et ceci sans l'aide d'aucun mécanisme externe. Le moyen qui lui permet d'apprendre et de reproduire de telles séquences temporelles, est bien l'existence de la mémoire dynamique. Les réseaux récurrents sont les seuls à pouvoir apprendre à reproduire une telle séquence.

Les techniques utilisées pour déterminer les paramètres du réseau de neurones lui permettant de reproduire des sorties désirées à des instants désirés sont appelées « *Trajectory Learning* » (Atiya *et al.*, 2000), (Aussem, 2002). Ces algorithmes permettent de faire apprendre au réseau une certaine trajectoire spatio-temporelle. Les deux techniques les plus utilisées sont la technique de rétropropagation du gradient de l'erreur dans le temps ou *Back-Propagation Through Time : BPTT* (Werbos, 1990), (Rumelhart *et al.*, 1986) et la Propagation Avant ou *Forward Propagation* appelé aussi *Real Time Recurrent Learning : RTRL* (Williams *et al.*, 1989). D'autres techniques sont mentionnées dans le travail de thèse de Ryad Zemouri (Zemouri, 2003).

La complexité temporelle du *RTRL* peut le rendre extrêmement lourd. Quant à l'algorithme *BPTT*, le réseau récurrent est transformé en un réseau feedforward par dépliement. Cette phase de dépliement peut rendre la technique de la rétropropagation du gradient très lourde et gourmande en ressource informatique. Les deux techniques ne garantissent cependant pas le suivi du gradient total de l'erreur de toute une séquence d'apprentissage, puisque la trajectoire suivie par le réseau dans l'espace d'état dépend des modifications apportées aux poids à chaque instant.

Un autre aspect, qui nous semble très important, est directement lié à la différence structurelle qui existe entre les architectures Globalement Récurrente Globalement Feedforward et Localement Récurrente Globalement Feedforward (*GRGF* et *LRGF*). Dans une architecture *GRGF*, une erreur de sortie du réseau de neurones peut être réinjectée à l'entrée du réseau, et donc avoir des répercussions sur les nouvelles réponses du réseau (par exemple les architectures de Jordan, Elman, Moakes, Miyoshi). Par contre, dans un réseau *LRGF*, la récurrence n'est tolérée qu'au sein du neurone lui même. Le réseau est donc vu globalement comme un réseau *feedforward*. Les erreurs de sortie du réseau ont donc de moindres répercussions que pour un réseau *GRGF*. Grâce à la récurrence locale, la mémoire dynamique du *RRFR* permet par conséquent la transformation du problème de reproduction de séquences en un problème d'interpolation linéaire.

Pour faire apprendre au réseau *RRFR* à reproduire des séquences temporelles réelles, nous avons exploité son comportement dynamique. Grâce à la récurrence locale des neurones d'entrée, le réseau *RRFR* est capable de garder une trace d'une excitation externe, soit indéfiniment dans le temps (pour un comportement de mémorisation) soit provisoirement (pour un comportement d'oubli). Dans notre cas, ce comportement d'oubli est utilisé de manière à transformer le problème de reproduction de séquences temporelles en un simple problème d'interpolation linéaire (**Zemouri, 2003**).

Nous avons établi deux tests du réseau *RRFR* sur l'apprentissage de la séquence chaotique Mackey-Glass et la séquence de commande du four à gaz.

a) REPRODUCTION DE LA SERIE CHAOTIQUE MACKEY-GLASS

Le premier test du réseau *RRFR* est celui d'apprendre à reproduire une séquence finie de la série chaotique Mackey-Glass. Le but rechercher ici est d'apprendre parfaitement reproduire une association *entrée-sortie* du réseau de neurones. En d'autres termes, on force délibérément le réseau à se trouver en situation de sur-apprentissage. Tous les points d'entrée au réseau seront donc mémorisés en tant que prototypes. Le calcul des rayons d'influence est réalisé par la technique du *RCE*.

La figure ci-dessous montre les résultats de l'apprentissage d'une séquence d'une longueur de 300 unités de temps. La figure de gauche présente les données réelles de la série temporelle, tandis que la figure de droite présente la sortie du réseau *RRFR* après apprentissage. Cette sortie correspond à l'évolution de la réponse du réseau *RRFR* face à l'évolution temporelle du neurone bouclé (en rouge). Ainsi, à chaque excitation externe, le réseau évoluera librement en reproduisant les 300 valeurs de la série temporelle apprise. L'erreur de reproduction de la série temporelle est quasiment nulle.

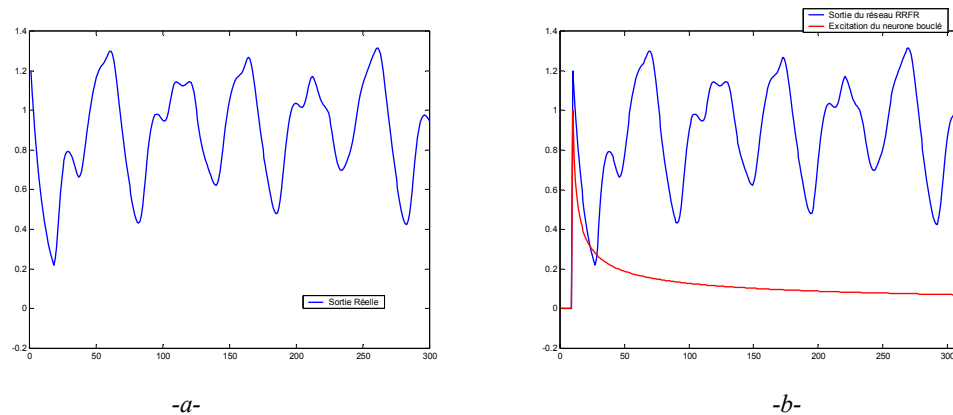


Figure 85. Apprentissage du réseau *RRFR* à reproduire les 300 premières valeurs de la série Mackey-Glass, a) données réelles de la série, b) sortie du réseau *RRFR* avec l'évolution de la sortie du neurone bouclé après excitation.

b) APPRENTISSAGE D'UNE TRAJECTOIRE DE COMMANDE

Nous avons effectué le même test du réseau *RRFR* sur la reproduction d'une séquence de commande $u(t)$ de l'application du four à gaz, décrite précédemment. Nous avons pour cette fois-ci testé les deux types de mémoires dynamiques : neurone à retour local de la sortie et neurone à retour local de l'activation. Les résultats du test sont présentés sur la Figure 86 : celle de gauche présente les résultats du *RRFR* avec neurone bouclé, et celle de droite, le réseau *RRFR* avec neurone à bouclage de l'activation. Les deux types de

mémoires arrivent à reproduire remarquablement la séquence de commande à chaque excitation de la mémoire dynamique (à chaque cycle de commande par exemple).

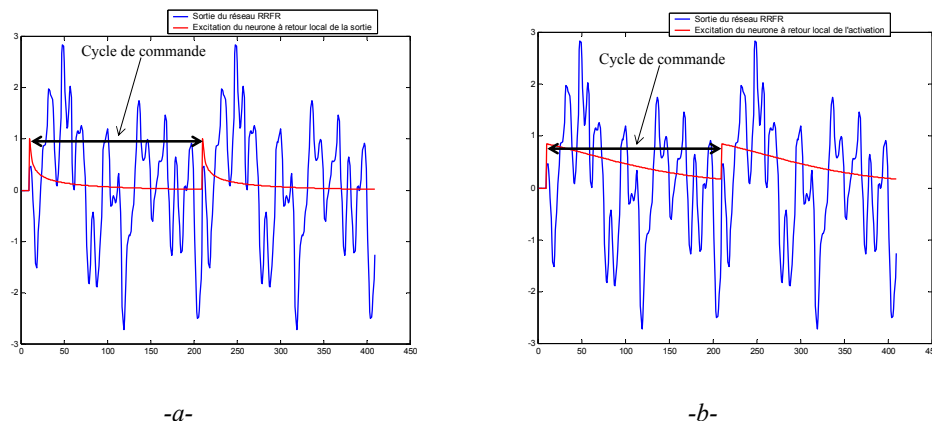


Figure 86. Apprentissage du réseau RRFR de la commande $u(t)$ du problème du four à gaz, a) RRFR avec neurone bouclé, b) RRFR avec neurone à retour local de l'activation.

5.2.4 BILAN DE L'ÉVALUATION DES PERFORMANCES DU RRFR

Suite à l'évaluation des performances du RRFR avec ses différentes configurations, ses différents algorithmes d'apprentissage et sur trois types d'applications des réseaux de neurones en surveillance industrielle dynamique, nous avons constaté que les performances des neurones bouclés (retour local de la sortie) et des neurones à retour local de l'activation sont comparables.

Dans la majeure partie des applications, nous conseillons néanmoins l'adoption du neurone bouclé. La raison essentielle de ce choix est en grande partie due à la différence structurelle entre les deux neurones récurrents. En effet, la récurrence du signal se situe avant la non-linéarité de la fonction d'activation pour le neurone à retour local de l'activation, et se situe après cette non-linéarité pour le neurone à retour local de la sortie. Cette différence de structure permet au neurone à retour local de la sortie d'intégrer la non-linéarité des données d'entrée dans sa prise en compte de la dynamique de ces données, puisque la fonction d'activation sigmoïde se trouve à l'intérieur du cycle de récurrence du neurone.

Afin de situer le RRFR ainsi évalué par rapport aux autres réseaux de neurones dynamiques, le paragraphe suivant traite une étude comparative menée dans la perspective d'une application en surveillance dynamique.

5.3 ÉTUDE COMPARATIVE DU RRBF PAR RAPPORT AUX STRUCTURES DE RESEAUX DE NEURONES TEMPORELS LES PLUS SIGNIFICATIFS, DANS L'OPTIQUE DE L'UTILISATION EN SURVEILLANCE DYNAMIQUE

Afin de sélectionner les réseaux de neurones dynamiques les plus pertinents pour la surveillance dynamique et effectuer ainsi une étude comparative efficace des principaux réseaux de neurones dynamiques, nous procédons à une classification plus précise des structures neuronales temporelles selon deux critères essentiels : l'architecture et

l'approche de généralisation utilisée. Il est à noter que ces travaux ont fait l'objet d'un article publié dans la Revue d'Intelligence Artificielle (**Palluat et al., 2005a**).

5.3.1 CHOIX DES STRUCTURES NEURONALES TEMPORELLES ETUDIEES

Au cours de nos recherches bibliographiques, nous avons pu établir que l'on pouvait classer la plupart des réseaux de neurones suivant la réponse de leurs neurones : réponses globales et réponses locales. Le choix d'une réponse locale ou globale déterminera les caractéristiques principales du réseau. En effet, un réseau temporel à représentation externe du temps donnant une réponse globale, comme le NETtalk (Sejnowski *et al.*, 1986), possède une très bonne capacité de généralisation. Toutefois, lors d'un nouvel apprentissage, tous les paramètres du réseau doivent être recalculés. On utilisera de préférence ce type de réseaux pour les problèmes d'interpolation. Par contre, pour un réseau donnant une réponse locale, comme le TDRBF (Berthold, 1994b), la sortie du réseau est non nulle si le vecteur d'entrée est proche d'un vecteur appris, et nulle sinon. Lors d'un nouvel apprentissage, seule une partie des paramètres du réseau sont générés, car seuls quelques neurones ont une réponse non nulle. Ces réseaux pourront être utilisés pour résoudre les problèmes d'extrapolation.

En prenant en compte un double critère architectural et temporel, les structures neuronales temporelles peuvent être classées selon le schéma suivant :

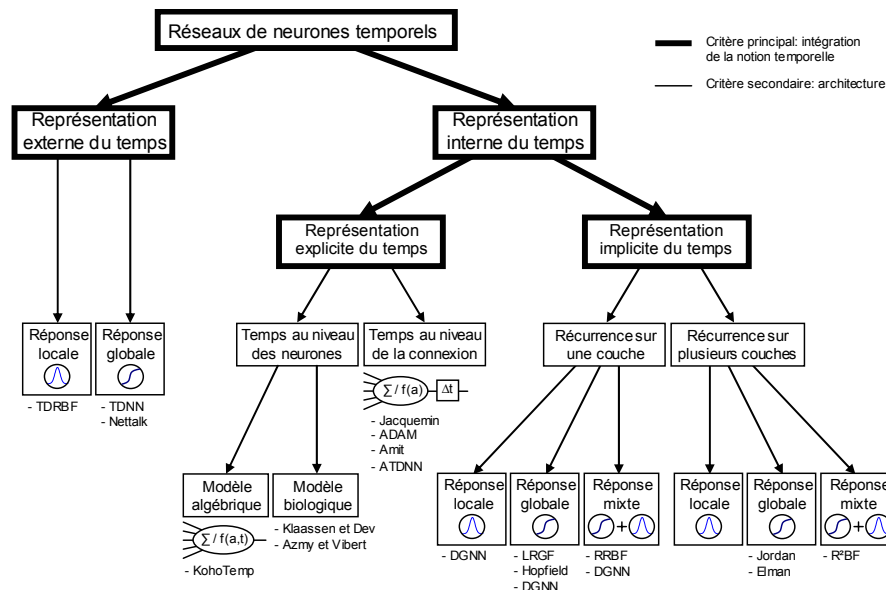


Figure 87. Classification des réseaux de neurones temporels à l'aide de deux critères : temporel et architectural.

Dans le cadre des applications de surveillance d'un système de production, la représentation externe du temps dans les réseaux de neurones temporels présente quelques inconvénients du fait de l'existence de la fenêtre temporelle. En effet, il est, d'une part, difficile de définir la taille de la fenêtre utile, et d'autre part, la taille de cette fenêtre modifie directement la taille du réseau.

Du point de vue des applications étudiées, la représentation interne du temps semble donc plus intéressante. Elle n'inclue pas d'éléments dynamiques externes. Elle ne nécessite donc pas de connaissances a priori du système. De plus, la dimension du vecteur d'entrée

est réduite car il n'est plus nécessaire de présenter les valeurs précédentes des entrées et des sorties à l'entrée du réseau.

Parmi les représentations internes du temps, la représentation implicite du temps semble plus adéquate grâce à sa facilité d'intégration et ses algorithmes d'apprentissage plus faciles à mettre en oeuvre. Il est toutefois bon de noter que les réseaux de neurones associés à cette représentation, appelés réseaux de neurones récurrent, possèdent aussi une faiblesse en ce qui concerne la définition de la taille de la mémoire comme cela a été montré dans (Bengio *et al.*, 93), mais cette faiblesse est insignifiante par rapport aux qualités de ces réseaux ainsi que par rapport au contexte d'application proposé.

Pour l'étude de la surveillance en ligne, nous effectuerons des opérations de classification et de prédiction. Dans le cas d'une classification, il est important que le réseau de neurones puisse donner un degré de pertinence à sa réponse et ainsi, lors d'un degré faible pour chaque réponse, ne pas donner de fausses réponses. L'aspect local répondant à cette condition nous éliminons ainsi les réseaux utilisant une réponse globale tels que les réseaux de Jordan, Elman et de Hopfield.

Il nous reste donc le choix entre la famille des réponses locales et celle des réponses mixtes. A part le RRFR que nous avons proposé, notre choix se porte ainsi sur le R^2BF et sur le DGNN.

Le RRFR est intéressant de part la simplicité du réseau, de la prise en compte du temps et des algorithmes d'apprentissages relativement simples. Pour le DGNN, et particulièrement pour le développement de Ferariu (Ferariu *et al.*, 2002), l'intérêt de l'étude provient du choix d'utiliser les algorithmes génétiques pour déterminer l'architecture optimale en fonction du problème posé. Cette faculté est particulièrement intéressante pour surveiller un système et pour pronostiquer une défaillance. Enfin pour le R^2BF , et notamment pour le développement de Frasconi (Frasconi *et al.*, 1996), son intérêt réside dans un comportement semblable à celui d'un automate, du à l'existence d'un récurrence entre plusieurs couches.

a) LE R^2BF (RECURRENT RADIAL BASIS FUNCTION)

Cette architecture (Figure 44), initialement prévue pour l'inférence inductive des grammaires régulières devra être modifiée pour l'appliquer au pronostic et à la surveillance dynamique. Les auteurs proposent des solutions pour l'algorithme d'apprentissage à l'aide de différentes méthodes qui pourront être appliquées telle que la méthode hybride définie par Moody et Darken (Moody *et al.*, 1989).

b) LE DGNN (DYNAMIC GENERAL NEURAL NETWORK)

Les réseaux DGNN (Figure 45) sont basés à la fois sur la structure des PMC et sur celle des RBF permettant ainsi d'utiliser les avantages de chacune d'entre elles : les capacités de généralisation des PMC et l'efficacité calculatoire des RBF. Le but principal de ces réseaux est de permettre la résolution des problèmes d'identification dans les systèmes non-linéaires. Devant la difficulté à sélectionner des modèles convenant à la résolution du problème complexe, les auteurs proposent une méthode basée sur les algorithmes génétiques permettant de déterminer la topologie et les paramètres adéquats.

Le principal avantage dans cette méthode se situe au niveau de la procédure de génération du DGNN qui ne nécessite aucune information sur le gradient des critères. Les modèles neuronaux obtenus se caractérisent à la fois par une bonne précision de leurs résultats et la simplicité de leurs architectures, l'utilisateur doit introduire un nombre réduit

de paramètres par rapport à une création manuelle d'un réseau DGNN. Par contre, la méthode nécessite de grande capacité calculatoire.

Afin de comparer les différentes architectures et algorithmes d'apprentissage de ces outils, nous proposons deux séries de tests en pronostic et surveillance dynamique.

5.3.2 ETUDE COMPARATIVE DANS LE DOMAINE DU PRONOSTIC

L'étude comparative dans le domaine du pronostic se fait sur les données du benchmark du four à gaz.

a) PRONOSTIC PAR RRFR

Le traitement s'effectue par une prise en compte directe de l'entrée plus la mémorisation des entrées précédentes. L'apprentissage s'effectue en deux étapes :

- un apprentissage non supervisé
- un apprentissage supervisé

L'apprentissage non supervisé permet de déterminer les centres et les rayons des neurones gaussiens. Pour déterminer les centres, nous avons choisi d'utiliser l'algorithme Fuzzy C-mean Clustering développé par Bezdek (Bezdek, 1974). Cet algorithme est analogue à l'algorithme k-mean clustering. Le but est de construire une pseudo partition floue constituée de c sous-ensembles flous. Les points sont alors assignés (de manière floue) à un sous-ensemble en déterminant la distance au centre des sous-ensembles.

La méthode Fuzzy C-mean Clustering permet de déterminer les centres des neurones gaussiens. Pour déterminer les rayons, nous utilisons la méthode tirée de l'algorithme RCE. La validité du résultat dépend du nombre de centres que nous choisissons.

Cette architecture s'adapte parfaitement pour la prédiction de concentration de CO₂. Même si l'algorithme peut demander de grandes ressources matérielles pour un nombre important de centres, il reste parfaitement acceptable lorsque ce nombre est restreint.

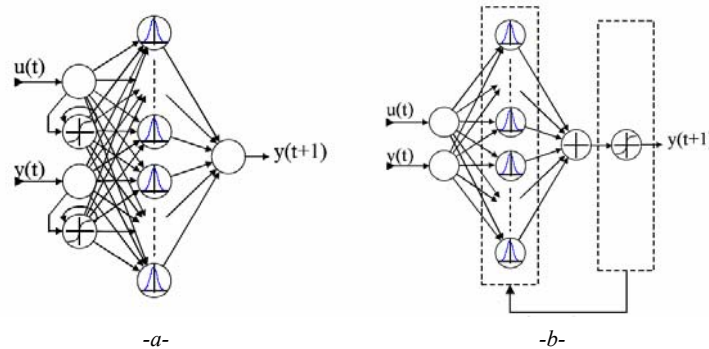


Figure 88. Architectures RRFR (a) et R^2BF (b) pour le pronostic.

b) PRONOSTIC PAR R^2BF

Le traitement s'effectue par une prise en compte directe de l'entrée. La mémorisation des états précédents s'effectue par le bouclage de la sortie sur les neurones gaussiens. L'apprentissage s'effectue en deux étapes : une partie non supervisée pour la détermination des centres à l'aide de la méthode Fuzzy C-mean clustering et des rayons à l'aide de la méthode tirée de l'algorithme RCE, ainsi qu'une seconde partie supervisée à l'aide d'une méthode de régression linéaire.

Cette architecture s'adapte bien pour la prédiction de concentration de CO₂. Même si l'algorithme peut demander de grandes ressources matérielles pour un nombre important de centres, il reste acceptable lorsque ce nombre est restreint (ici égal à 5).

c) PRONOSTIC PAR DGNN

A des fins de simplification de représentation, nous introduisons les notations de la Figure 89-a. Le réseau DGNN utilisé pour le pronostic indiqué dans la Figure 89-b :

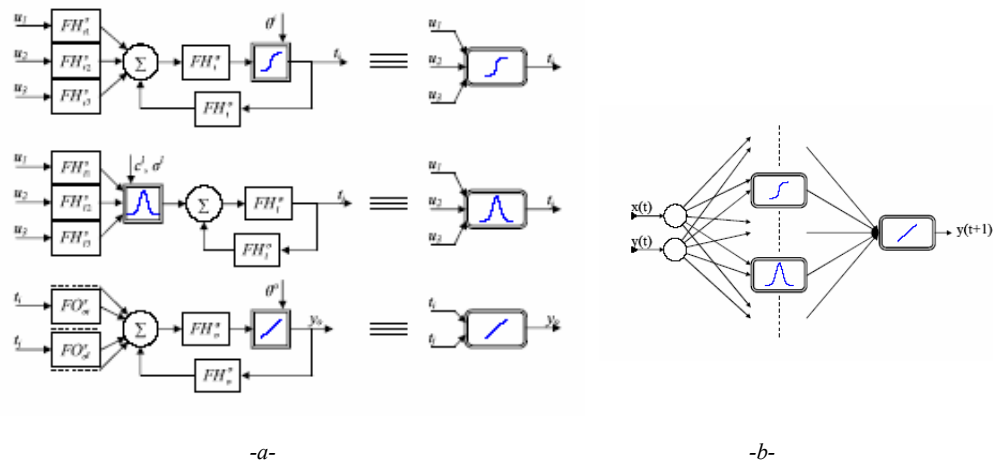


Figure 89. Simplification du réseau DGNN (a), structure DGNN utilisée pour le pronostic(b).

Le traitement s'effectue par une prise en compte directe de l'entrée. L'apprentissage s'effectue avec l'algorithme basé sur les algorithmes génétiques. Il permet à partir d'une population de DGNN admissibles de déterminer le réseau optimal pour l'application désirée. Pour cela, une optimisation multicritères est appliquée et résolue dans le sens de Pareto (la solution est représentée par une famille de points, nommée ensemble Pareto optimal).

Cette architecture ne s'adapte que moyennement pour la prédiction de concentration de CO₂. Même si l'algorithme peut sembler prometteur avec l'utilisation d'algorithme génétique, il reste cependant décevant quand au temps de calcul et au résultat obtenu. Un résultat plus intéressant pourrait être obtenu avec un plus grand nombre de neurones dans la couche cachée mais cela augmenterait de façon considérable le temps de calcul.

d) SYNTHESE SUR L'UTILISATION DE RESEAUX DE NEURONES RECURRENTS POUR LE CAS DU PRONOSTIC

Le tableau suivant reprend les résultats des tests de pronostic par chacun des 3 réseaux :

	Erreur moyenne	Temps d'apprentissage	Temps de reconnaissance
RRFR (RRBF)	0.86 %	0.26 secondes	0.2 millisecondes
R^2BF	1.25 %	0.81 secondes	0.3 millisecondes
DGNN	2.37 %	260 000 secondes (3 jours)	2.1 millisecondes

Tableau 5. Synthèse des essais comparatifs sur le pronostic

Le réseau RRFR apparaît comme celui dont l'erreur de prédiction est la plus faible tout au long de la séquence. Pourtant, le réseau R^2BF est aussi particulièrement intéressant dans ces résultats. Le réseau DGNN termine en dernière position du fait de ses résultats moins satisfaisant. Avec un plus grand nombre de neurones dans la couche cachée, ce réseau donnerait sûrement de meilleurs résultats, mais cela augmenterait considérablement le temps de calcul. En ce qui concerne ce temps d'apprentissage, le réseau RRFR est une nouvelle fois le plus intéressant. Le réseau R^2BF , avec un temps trois fois plus long est parfaitement acceptable puisqu'il reste inférieur à la seconde. Ce temps est important car dans les premiers temps d'utilisation du réseau, il est possible qu'il y ait de nombreuses séquences d'apprentissage pour mettre à jour la base de donnée d'apprentissage afin d'effectuer une prédiction plus précise, tout en évitant bien entendu le surapprentissage. Le DGNN avec ses trois jours de calculs est bien entendu en dernière position.

Il est important de noter que cette séquence de test ne comportait pas de phase d'apprentissage en ligne. Si un tel cas devait être traité, les algorithmes choisis nécessitant l'utilisation de l'ensemble de la base, donneraient des résultats équivalents à un apprentissage hors ligne. Donc pour un système dont la réactivité est plutôt lente comme le four à gaz, les réseaux RRFR et R^2BF seraient applicable contrairement au DGNN.

5.3.3 ETUDE COMPARATIVE DANS LE DOMAINE DE LA SURVEILLANCE DYNAMIQUE

L'étude comparative en surveillance dynamique se fait à l'aide du benchmark de surveillance d'un bras de robot (*Robot Execution Failures*⁴⁷).

a) SURVEILLANCE DYNAMIQUE PAR RRFR

Le RRFR utilisé pour la surveillance dynamique est le suivant :

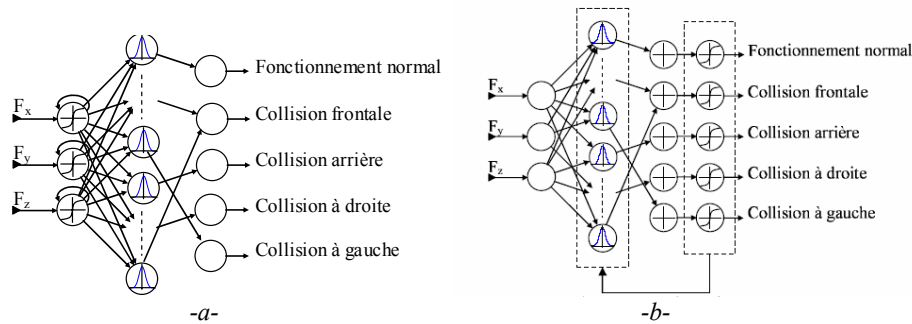


Figure 90. Structure RRFR (a) et R^2BF (b) pour la surveillance dynamique

Le RRFR fonctionne parfaitement pour la détection d'erreur. En ce qui concerne les erreurs de classification, elles peuvent venir de la pauvreté de la base d'apprentissage, ainsi que des trop faibles différences entre les séquences de catégories différentes. L'avantage de ce neurone réside dans l'utilisation de la mémoire statique et de son algorithme d'apprentissage simple et stable. En effet, avec la même base, l'apprentissage peut être fait un grand nombre de fois. On trouvera toujours le même nombre de centres, les mêmes rayons, et donc les mêmes résultats en classification.

⁴⁷ Benchmark disponible sur le site de l'Université de Californie à Irvine : <http://kdd.ics.uci.edu/databases/robotfailure/robotfailure.html>

b) SURVEILLANCE DYNAMIQUE PAR R^2BF

Le R^2BF , ou tout du moins son algorithme d'apprentissage, est impropre en détection d'erreur. Ceci est dû principalement à son manque de stabilité. Un autre de ses défauts réside dans la nécessité d'une grande capacité calculatoire. Une solution consisterait à déterminer au préalable le bon nombre de centres ainsi que la partition pseudo floue initiale.

c) SURVEILLANCE DYNAMIQUE PAR DGNN

La figure suivante présente la structure DGNN utilisée pour la surveillance dynamique :

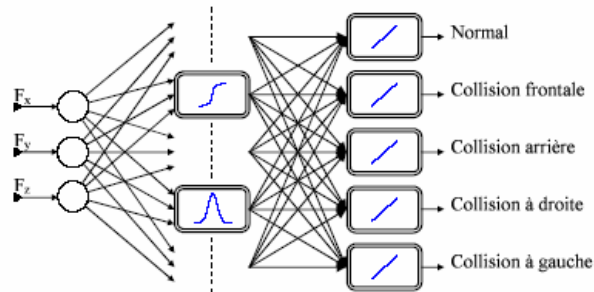


Figure 91. Schématisation du réseau DGNN pour la surveillance dynamique.

Ce réseau présente plus d'intérêt dans la théorie que dans la pratique. Son grand défaut se situe dans les capacités calculatoires gigantesques qui lui sont nécessaires pour effectuer le moindre apprentissage. Il n'est donc pas utilisable pour une application industrielle où nous pourrions avoir à ajouter de nouvelles données à apprendre assez souvent dans les premiers mois d'utilisation.

d) SYNTHESE SUR L'UTILISATION DE RESEAUX DE NEURONES RECURRENTS POUR LA SURVEILLANCE DYNAMIQUE

Reprenons les résultats de chacun des réseaux :

	Temps d'apprentissage	Temps de reconnaissance
RRFR (RRBF)	1.03 secondes	2 millisecondes
R^2BF	14 secondes	1 millisecondes
DGNN	13 423 secondes	20 millisecondes

Tableau 6. Temps d'apprentissage et de reconnaissance pour la surveillance dynamique

	Détection (%)			Classification (%)		
	RRBF	R2BF	DGNN ²	RRBF	R2BF	DGNN ²
Fonctionnement normal				73.68	54.32	0.00
Collision avant	100.00	62.33	80.00	50.00	7.00	80.00
Collision arrière	100.00	50.40	66.67	60.00	8.80	0.00
Collision à droite	100.00	78.00	50.00	50.00	12.00	0.00
Collision à gauche	100.00	68.50	100.00	75.00	21.00	0.00

Tableau 7. Erreurs en surveillance dynamique

Il est important de noter que les résultats du DGNN ont été intégrés de manière indicative afin de démontrer ses faibles performances en plus d'un temps de calcul déjà conséquent. Ce réseau ou tout du moins son algorithme d'apprentissage est donc incompatible avec la surveillance dynamique. En effet, comme pour le pronostic, il est parfois nécessaire d'effectuer plusieurs apprentissages avec plusieurs bases d'apprentissage différentes avant d'obtenir un réseau qui nous permettrait d'obtenir des résultats satisfaisants. Avec un temps d'apprentissage estimé à 2700 h, il apparaît clairement que nous ne pouvons appliquer cette méthode.

En ce qui concerne le réseau R^2BF , le problème provient une nouvelle fois de l'apprentissage. Si cette fois le temps de calcul est correct, le problème se situe dans la non régularité des résultats. Ce défaut peut éventuellement être atténué en déterminant une pseudo partition floue initiale.

Le seul réseau qui a donné des résultats réguliers avec un temps d'apprentissage court est le réseau RRFR. Si ces résultats sont excellents en détection d'erreurs, il reste moyen dans la classification de celles-ci. Ce défaut est dû en partie à la faiblesse de la base d'apprentissage. Une base plus fournie devrait donc donner de meilleurs résultats.

Il est, une nouvelle fois, important de noter que cette séquence de test ne comportait pas de phase d'apprentissage en ligne. Si un tel cas devait être traité, le réseau RRFR aurait un avantage encore plus prononcé. En effet, si les deux autres réseaux, R^2BF et DGNN, nécessitent de reprendre l'ensemble de la base pour un nouvel apprentissage, le RRFR n'utilise l'ensemble de la base que pour certains cas. Admettons que nous souhaitions apprendre une donnée non reconnue par la configuration actuelle du réseau. Suivant l'algorithme d'apprentissage du RRFR, il est alors nécessaire de créer un nouveau prototype centré sur la donnée et de relier ce prototype à la classe de la donnée. Il n'est pas nécessaire de modifier le reste du réseau et donc de reprendre la base.

Nous avons ainsi un temps d'apprentissage en ligne extrêmement court. Pour le cas où l'ensemble de la base est utilisé, le temps d'apprentissage en ligne est équivalent au temps d'un apprentissage hors ligne. Il est alors nécessaire que le temps d'échantillonnage soit supérieur au temps d'apprentissage. Bien entendu, plus le temps d'apprentissage est court, plus les possibilités d'applications sont grandes.

5.3.4 BILAN DE L'ETUDE COMPARATIVE DES DIFFERENTES STRUCTURES NEURONALES TEMPORELLES

En prenant en considération des critères de représentation temporelle et de structure, trois types de réseaux de neurones temporels ont été choisis pour leur applicabilité dans le domaine de la surveillance : le RRFR, le R^2BF et le DGNN.

Ces réseaux ont été adaptés et testés sur des applications de pronostic et de surveillance dynamique. Une étude comparative est ainsi menée autour de deux benchmarks spécifiques aux domaines concernés.

Les résultats des deux séquences de test ont montrés que le réseau RRFR était le plus efficace des trois réseaux, à la fois par ses résultats que par ses temps d'apprentissage. Pour le réseau R^2BF , ses résultats moyens dans la surveillance dynamique ne permettent pas de l'employer dans cette situation. Un algorithme d'apprentissage plus performant permettrait sûrement une révision de ce jugement. Enfin, pour le DGNN, son algorithme

d'apprentissage extrêmement demandeur de capacité calculatoire ne lui permet pas d'être implanté aisément.

Par ailleurs, nous avons constaté que le RRFR présente un intérêt certain du point de vue de l'apprentissage en ligne, un aspect extrêmement important. En effet, le RRFR possède d'une part une architecture simple (séparation de la mémoire dynamique et de la mémoire statique) et d'autre part, les temps d'exécutions des algorithmes d'apprentissage sont relativement courts. Dans l'optique d'un transfert industriel de ce concept, cette propriété apparaît comme fondamentale.

5.4 EXPLOITATION INDUSTRIELLE DE LA SOLUTION DE SURVEILLANCE DYNAMIQUE PAR RRFR

5.4.1 INTRODUCTION

L'exploitation industrielle du *RRFR* proposé met en évidence l'intérêt de ce nouveau réseau de neurones dynamique pour la mise au point de systèmes de surveillance dynamiquement en temps réel, paramétrables à distance via le Web. En effet, nous avons développé une solution originale concernant l'implémentation du réseau *RRFR* dans un Automate Programmable Industriel (API). Cette solution de surveillance neuronale embarquée permet de profiter de la rapidité du traitement des données et de la sûreté de fonctionnement de l'automate industriel, ainsi que de l'ouverture donnée par l'interface coupleur web de ce type de système.

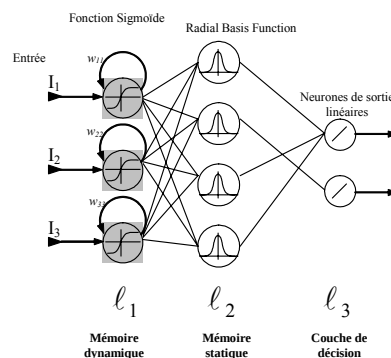


Figure 92. Réseau Récurrent à base de Fonctions Radiales avec ses deux mémoire (dynamique et statique) embarqué dans l'automate programmable.

Cette application s'encadre dans le domaine de la *e-maintenance*, en faisant appel aux supports modernes de communication ainsi qu'en donnant plus d'autonomie et d'intelligence aux équipements de surveillance temps réel. L'étude a été développée en collaboration avec une entreprise régionale - *AVENSY Ingénierie* -, spécialiste du suivi de production et de la gestion de la maintenance. Son originalité réside dans l'exploitation du concept *coupleur Ethernet - TCP/IP* présent dans les automates programmables modernes.

Dans ce contexte de collaboration, nous avons apporté notre contribution au développement d'une solution réactive de surveillance industrielle en temps réel. Nous avons ainsi opté pour un accès à l'information à distance avec une surveillance en ligne temps réel effectuée par un automate. Pour cela, nous avons dû « décomposer » le réseau *RRFR* en fonctions élémentaires pour qu'il soit interprétable en langage automate. En effet,

le langage de programmation automate (step7 pour les API Siemens) représente un langage de bas niveau et nécessite une certaine restructuration du réseau *RRFR*.

La solution que nous avons développée comporte deux parties :

- *Surveillance temps réel en ligne : le RRFR chargé dans l'unité centrale de l'automate traite les variables de surveillance et donne sa réponse en temps réel,*
- *Apprentissage et affichage des résultats de surveillance à distance : le réseau de neurones chargé dans l'automate devra être entièrement mis à jour par un expert distant via le coupleur web.*

Un premier prototype de cette maquette a été présenté lors d'un salon sur les microtechniques⁴⁸. L'intérêt de la société *AVENSY Ingénierie* étant de commercialiser cette solution de surveillance neuronale temps réel, les deux premières étapes étaient bien évidemment le dépôt d'un brevet et le lancement d'une étude de marché.

Après plusieurs concertations avec notre partenaire, nous avons abouti à une solution qui semble s'adapter aux besoins actuels des entreprises industrielles. La solution retenue est détaillée par la suite, en y associant une série de tests de performance.

5.4.2 CHOIX DE LA SOLUTION DE SURVEILLANCE DYNAMIQUE EN TEMPS REEL

Les exigences actuelles du marché dans le domaine de la surveillance, présentent quelques caractéristiques récurrentes :

- *Capacité d'assurer une surveillance en temps réel.*
- *Capacité d'avoir une architecture flexible et évolutive.*
- *Simplicité d'utilisation et de paramétrage.*
- *Entièrement paramétrable à distance (apprentissage à distance)*

La configuration finale de la solution de surveillance temps réel est l'aboutissement d'un processus sélectif de conception. Pour plus de détails, toutes les étapes franchies vers la solution finale peuvent être consultées dans la thèse de R. Zemouri (2003).

Afin d'éviter des problèmes de lenteur du traitement et d'instabilité de l'application, la solution choisie concerne la surveillance embarquée au niveau d'un automate programmable. Le programme de surveillance neuronal est ainsi chargé dans l'unité centrale (CPU) de l'automate. Une surveillance en temps réel est donc possible. Le système pourra même agir sur l'équipement en cas de défaillance critique.

L'outil que nous proposons représente un outil de surveillance et d'aide à la décision. Le coupleur WEB permet dans ce cas non seulement d'avoir une Interface Homme Machine distante, mais également de mettre à jour le réseau de neurones depuis une connexion TCP/IP : *apprentissage à distance*. Les programmes (IHM + Apprentissage) sont mis au point en langage *JAVA – langage distribué, multi plateforme* – offrant une généricité parfaitement adaptée à notre application.

La Figure 93 montre l'architecture générale d'une telle configuration avec les différents postes distants. Une base de données mémorisant toute l'évolution du système à surveiller est créée et mémorisée dans le coupleur WEB de l'API. Cette base de données contient les variables de surveillance horodatées ainsi que les différentes réponses du réseau de neurones. Un expert distant peut ainsi se baser sur cet historique pour mettre à jour les

⁴⁸ 14ème SALON INTERNATIONAL DES MICROTECHNIQUES, MICRONORA, Besançon 2002.

paramètres du réseau de neurones embarqué. Cet expert peut très bien être situé à l'intérieur de l'entreprise (Intranet) ou bien à l'extérieur de l'entreprise (Internet).

Pour aboutir à cette configuration, plusieurs choix étaient possibles (**Zemouri, 2003**). Notre choix s'est porté sur une solution dans laquelle la mise à jour du réseau de neurones (apprentissage) est gérée par le programme JAVA localisé au niveau du coupleur WEB. L'expert responsable du paramétrage du réseau de neurones ne sera plus obligé de le reprogrammer en step7 ni même de se déplacer au niveau de l'automate pour le recharger dans la CPU (par réseau local industriel). Toute la phase de mise à jour s'effectue par connexion TCP/IP via le coupleur WEB. La réussite d'une telle configuration nécessite une structure du programme neuronal évolutive et flexible.

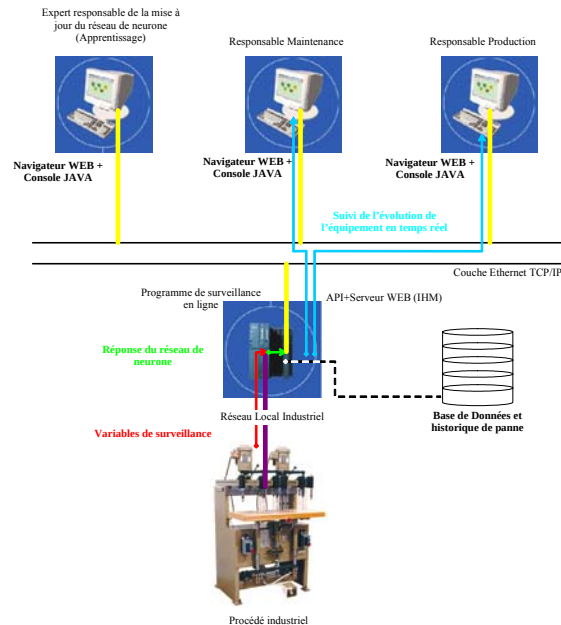


Figure 93. Solution de surveillance en ligne avec apprentissage distant via le protocole de communication Internet TCP/IP.

5.4.3 DESCRIPTION DE LA SOLUTION PROPOSEE

a) INTRODUCTION

L'architecture globale de la solution proposée est illustrée par la Figure 94. Une fois chargée dans la CPU de l'automate, la structure du programme neuronal est capable d'évoluer en étant supervisée par un expert distant. Toute la phase d'apprentissage est gérée via le coupleur WEB par la connexion TCP/IP. Le programme JAVA chargé dans le coupleur permet de faire la liaison entre l'automate et l'expert. Pour ce faire, deux fonctions de base sont nécessaires : une fonction de *lecture* et une fonction d'*écriture*. Ces deux fonctions sont fournies par le constructeur⁴⁹ et permettent donc de communiquer avec l'automate soit par réseau interne à l'entreprise (*Intranet*) soit par réseau externe (*Internet*). Les conditions imposées par notre partenaire industriel à la structure du programme neuronal en step7 sont les suivantes :

⁴⁹ Dans notre application, les deux fonctions ont été fournies par SIEMENS.

- *Aspect évolutif :*

La structure doit être entièrement évolutive via le coupleur WEB. L'utilisateur n'aura pas à reprogrammer et recharger le réseau de neurones dans la CPU de l'automate. Cette facilité d'utilisation est un critère très important et très apprécié par les industriels qui ne possèdent pas forcément de connaissances sur l'outil neuronal. Le paramétrage du réseau de neurones doit être entièrement transparent à l'utilisateur. Cette solution ouvre des perspectives vers une externalisation de la maintenance via le web. En effet, l'expert distant pourra suivre les différentes évolutions des variables de surveillances ainsi que les réponses du réseau de neurones chargé de la surveillance de l'équipement. Il pourra également le paramétrer par connexion TCP/IP.

- *Flexibilité :*

La structure du programme neuronal doit être standard pour un équipement quelconque. Le programme doit être capable de s'adapter à un nombre quelconque de variables de surveillance (nombre de sortie capteur) et à plusieurs types de variables de surveillance (entier, double entier, booléen, etc.). Cette contrainte a pour exigence une plus grande ouverture au maximum de clients potentiels. Le développement d'un produit générique coûtera moins cher qu'un produit fait sur mesure à un client précis.

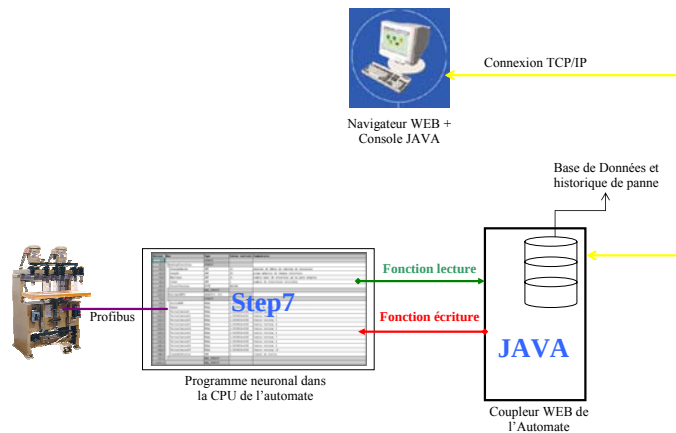


Figure 94. Différentes interactions entre chaque composant de l'application de surveillance.

b) STRUCTURE DU PROGRAMME NEURONAL EN LANGAGE AUTOMATE

L'implémentation du réseau RRFR dans un Automate Programmable Industriel, est basée essentiellement sur deux composants⁵⁰ de programmation :

- Un composant (**FC** en step7) qui permet de définir des fonctions de calcul type comme le calcul de la sortie du neurone bouclé ou gaussien en fonction de ses entrées.
- Un composant (**DB** en step7) qui permet de stocker tous les paramètres du réseau RRFR. Ces paramètres sont exploités par les composants FC pour les différents calculs.

La Figure 95 présente la structure globale du programme neuronal en step7. Chaque couche du réseau de neurones est représentée par une FC et une DB. La communication avec le programme step7 via le coupleur web ne peut se faire que sur des variables se trouvant dans un composant DB. Un numéro est associé à chaque DB et chaque variable possède une adresse absolue fixe dans une DB. En localisant ainsi l'emplacement de

⁵⁰ Nous ne nous attarderons pas sur le détail de ces deux composants. Pour plus d'explication, se référer à la documentation technique du constructeur SIEMENS.

chaque variable du programme, on peut soit afficher cette variable sur un poste distant, soit changer sa valeur par un expert distant grâce aux fonctions de lecture écriture.

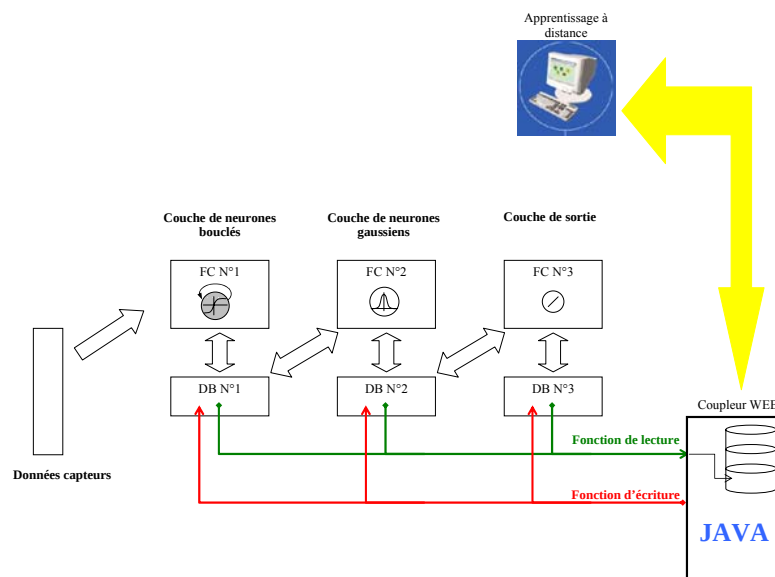


Figure 95. Organigramme du réseau RRFR chargé dans la CPU de l'automate. La communication avec le coupleur Web se fait à travers des variables situées dans le composant DB.

Les deux critères de *flexibilité* et d'*évolutivité* sont obtenus grâce à l'utilisation de variables de type « *structure* » pour chaque DB des différentes couches du réseau de neurones. Chaque variable de type structure contient une entête où figurent certaines informations exploitées par des *pointeurs* et qui servent à identifier l'adresse de chaque paramètre du réseau de neurones.

c) APPRENTISSAGE ET VISUALISATION DES DONNEES

La phase de mise à jour du réseau de neurones (phase d'apprentissage) consiste à re-paramétrer le réseau de neurones en fonction de l'évolution de l'équipement. Cette étape est entièrement gérée par le langage de programmation JAVA localisé dans le coupleur Web de l'automate. Un expert distant responsable de cette mise à jour peut donc modifier les paramètres du réseau à travers une interface de visualisation. La version renforcée de l'algorithme des k-moyennes est alors facilement programmable en langage JAVA. Une fois les paramètres du réseau *RRFR* extraits, ils sont chargés dans le programme automate par les fonctions d'écriture.

Comme le montre la figure suivante, une base de données est créée par le programme JAVA. Cette base de données contient des informations qui seront exploitées pour la phase d'apprentissage. Ces informations sont les valeurs des différentes variables de surveillance horodatées (se trouvant dans la DB N°1) ainsi que les différentes réponses du réseau de neurones (dans la DB N°3). Un expert distant peut alors consulter cet historique, analyser les différentes données de cette base et exécuter l'algorithme d'apprentissage. Les paramètres du nouveau réseau de neurones sont alors obtenus. L'expert peut ainsi recharger ces nouveaux paramètres dans la CPU de l'automate par la fonction d'écriture. Les DB concernées par cette phase d'écriture sont les DB N°1 et N°2 où figurent respectivement les paramètres des neurones bouclés et des neurones gaussiens. L'ajout de nouveaux neurones gaussiens devient alors aisé. Il suffit de mettre le nouveau nombre de neurones gaussiens au niveau du nombre de sous structures contenues dans la structure

principale de la DB N°2 ainsi que les valeurs de leurs paramètres (prototypes et rayons d'influence) aux adresses correspondantes. Le programme neuronal chargé dans l'automate est ainsi entièrement *évolutif* par connexion TCP/IP via le coupleur web.

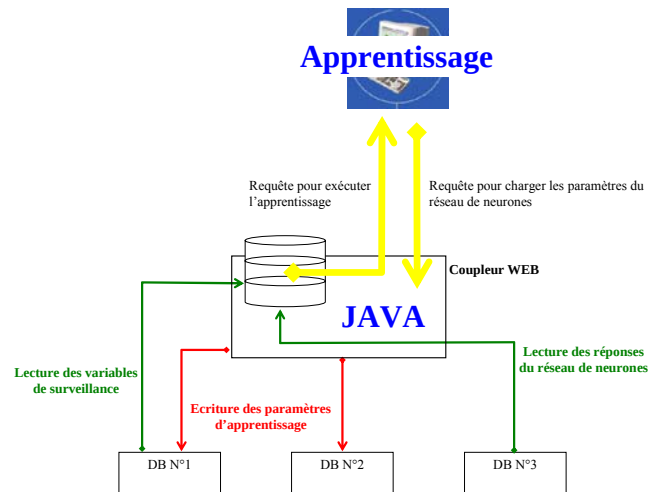


Figure 96. Organigramme du processus d'apprentissage du réseau de neurones. Le programme JAVA chargé dans le coupleur peut être exécuté à distance (TCP/IP). Une base de données est également disponible au niveau du coupleur et consultable à distance.

La Figure 97 montre l'interface de visualisation pour un utilisateur passif c'est-à-dire sans pouvoir d'écriture dans l'automate.

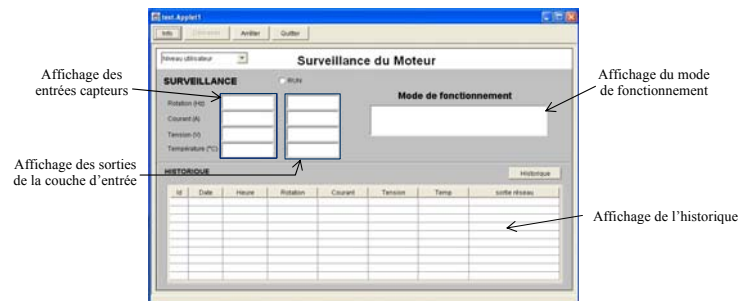


Figure 97. Interface (IHM) pour un utilisateur passif - visualisation des valeurs des paramètres de surveillance, du mode de fonctionnement et de l'historique de l'équipement.

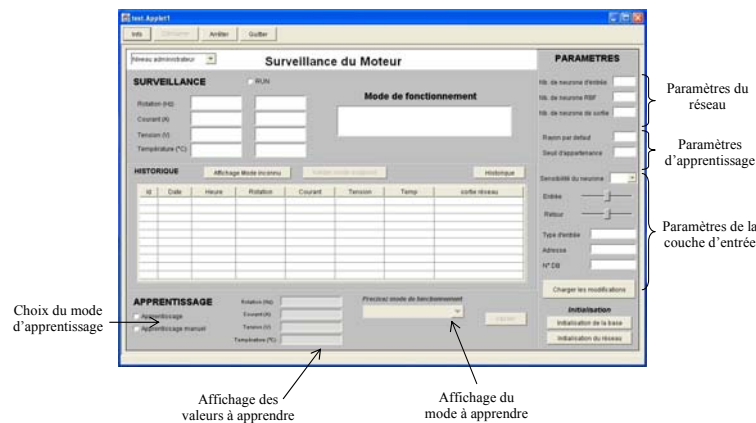


Figure 98. Interface (IHM) pour un utilisateur avec pouvoir (expert actif) - seule personne à pouvoir configurer le programme neuronal chargé dans l'unité central de l'automate.

Cette interface ne sert qu'à lire les données de surveillance. Un tel degré de sécurité est nécessaire si l'on veut ouvrir l'accès au programme de surveillance à des personnes qui ne sont pas forcément qualifiées pour la phase de paramétrage du programme neuronal (phase d'apprentissage). Cette phase d'apprentissage est gérée par une interface dédiée à un expert responsable de la mise à jour du programme de surveillance (Figure 98). Cet expert est donc autorisé à écrire dans le programme neuronal de l'automate et ainsi paramétrer le réseau *RRFR* à distance.

5.4.4 EVALUATION DES PERFORMANCES DU PROTOTYPE

a) DESCRIPTION DE LA MAQUETTE DE TEST

Nous avons évalué et testé les performances du programme neuronal sur une maquette comprenant un automate SIEMENS équipé d'une CPU 414-2 et d'un coupleur Web CP 443-1, un variateur de type MICROMASTER 4 et un moteur à courant continu.

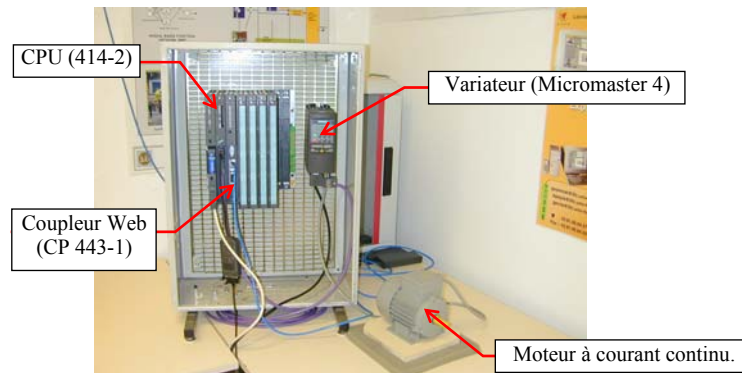


Figure 99. Maquette de test.

b) EVALUATION DES TEMPS DE CYCLE

Plusieurs temps de cycles pour plusieurs dimensions de réseaux de neurones ont été évalués sur cette maquette. En fonction des différents temps d'exécution de chaque module du programme neuronal, nous avons pu formaliser (une précision de l'ordre de 5 %) ce temps de cycle (T_c) par la relation suivante :

$$T_c = 0.241 \times N_{ent} + 0.0152 \times N_{ent} \times N_{rbf} + 0.08 \times N_{rbf} \quad [27]$$

où N_{ent} représente le nombre de neurones bouclés en entrée, N_{rbf} le nombre de neurones gaussiens. Le temps de cycle (T_c) est estimé en millisecondes (ms). Le tableau suivant présente les différents temps de cycles en fonction de plusieurs topologies du réseau *RRFR*.

N_{ent}	N_{rbf}	T_c (ms)	N_{ent}	N_{rbf}	T_c (ms)
2	10	1,586	8	50	12,008
2	50	6,002	8	100	22,088
2	100	11,522	8	200	42,248
2	150	17,042	8	300	62,408
2	200	22,562	8	400	82,568
5	100	16,805	10	100	25,61
5	200	32,405	10	500	118,41
5	500	79,205	10	1000	234,41

Tableau 8. Différents temps de cycle en fonction de plusieurs dimensions du réseau *RRFR*.

On peut penser qu'une dimension moyenne d'un réseau *RRFR*, avec laquelle on peut obtenir de bons résultats⁵¹, est constituée d'environ 5 neurones bouclés et 500 neurones gaussiens. Une telle dimension possède un temps de cycle de 79 ms. Sauf pour des processus rapides, un temps de cycle de cet ordre (env. 80 ms) représente une perspective intéressante pour des applications de surveillance temps réel de processus industriel.

En prenant en compte la relation des temps de cycle – temps d'acquisition, on peut avoir deux cas de figures : une situation où le temps de cycle du programme neuronal est inférieur au temps d'acquisition. Dans ce cas, une synchronisation entre les deux cycles est nécessaire. Par contre, si le temps d'acquisition est plus rapide que le temps de cycle du programme neuronal, on perd dans ce cas une partie des données de surveillance. Une étude détaillée de ces deux situations est présentée dans (**Zemouri, 2003**).

5.4.5 BILAN DE L'EXPLOITATION INDUSTRIELLE

Nous avons présenté une solution d'exploitation du réseau *RRFR* pour des applications de surveillance industrielle en temps réel. Le principe de cette solution est d'avoir un programme neuronal en langage automate, chargé dans l'unité centrale de celui-ci pour traiter les données capteurs en temps réel. La phase d'apprentissage et de mise à jour du réseau *RRFR* est entièrement gérée à distance par un expert responsable de la mise à jour du réseau *RRFR* embarqué. Cette communication avec un expert distant est obtenue grâce au coupleur Web de l'automate. Ce coupleur joue le rôle d'une jonction entre la couche de communication TCP/IP et l'unité centrale de l'automate (CPU). Nous avons donc structuré le programme automate pour qu'il puisse être entièrement évolutif à distance, c'est-à-dire permettre à un expert distant d'effectuer un apprentissage par connexion TCP/IP.

Le réseau *RRFR* s'est montré performant pour ce type d'exploitation. Ceci est dû à la simplicité de son architecture et de son algorithme d'apprentissage. Les choix effectués, en occurrence le choix du réseau *RFR* (pour la partie statique du traitement) ainsi que celui des architectures localement récurrentes (*LRGF* pour la partie dynamique), ont été très avantageux. Une autre architecture récurrente, comme un *PMC* globalement récurrent aurait été plus compliquée à embarquer et aurait certainement pris beaucoup plus de mémoire dans la *CPU*. La gestion des connexions d'un tel réseau récurrent est beaucoup plus compliquée qu'une architecture *LRGF*.

Des tests de performance ont été effectués par rapport à plusieurs dimensions du réseau *RRFR* embarqué. A partir des quantifications des différents temps de cycles du programme neuronal, nous considérons que cette solution ouvre des perspectives très intéressantes en traitement temps réel des variables de surveillance à condition de bien veiller à ce que le nombre de neurones gaussiens soit bien choisi pour ne pas avoir un cycle d'acquisition plus rapide que le cycle du programme neuronal dans l'automate. Ce nombre maximum a été quantifié en fonction du nombre des variables à surveiller et en fonction du cycle d'acquisition.

Un autre avantage de cette solution est donné par la robustesse industrielle des architectures API. En effet, contrairement aux PC qui ne sont pas à l'abri d'un bug imprévu, un automate est conçu pour fonctionner dans un environnement industriel hostile avec plus de fiabilité et de robustesse.

⁵¹ Par exemple la carte IBM/ZISC (Zero Instruction Set Computer), qui représente une implémentation Hard d'un réseau *RFR* possède 576 neurones gaussiens. (Zemouri et al., 2001)

En ce qui concerne la phase de commercialisation du système neuronal de surveillance avec notre partenaire *AVENSY Ingénierie*, il faudra continuer le prototypage par un travail de développement et de test de détection, de diagnostic de défaillances ainsi que des tests de prédiction en environnement industriel. Ces tests sont absolument nécessaires pour convaincre une clientèle industrielle (constructeur de machines outils, professionnels de la surveillance et de la maintenance, intégrateurs, etc.) devenue de plus en plus exigeante. Des perturbations peuvent alors être rajoutées à la maquette pour des simulations de défaillances, car provoquer de vraies défaillances avec comme risque la destruction du matériel serait relativement dangereux et potentiellement coûteux. Ces perturbations peuvent être obtenues en agissant sur les valeurs de surveillance après l'acquisition, c'est-à-dire en générant des perturbations au niveau du programme automate (soft) et non pas sur le matériel (hard). Plusieurs combinaisons de défaillances simulées peuvent alors être obtenues en agissant sur l'ensemble des données acquises.

5.5 CONCLUSION ET PERSPECTIVES

5.5.1 PRINCIPALES CONTRIBUTIONS

Le réseau *RRFR* se démarque des autres architectures de réseaux récurrents essentiellement par sa simplicité de paramétrage. Nous avons ainsi mis en évidence ces propriétés en le testant sur trois problématiques distinctes :

- *reconnaissance de séquences temporelles*

Nous avons montré la simplicité avec laquelle le réseau RRFR est capable d'apprendre plusieurs séquences booléennes simples. Sa capacité de généralisation locale lui permet de reconnaître des séquences proches de celles apprises et de détecter des séquences inconnues. Ce type d'application est très utile pour la surveillance d'un système à événements discrets. Le réseau apprend à reconnaître des séquences de bon fonctionnement et à détecter des séquences de dysfonctionnement connues. En surveillance de paramètres réels, l'apprentissage de séquences temporelles peut servir à la détection précoce d'un palier de dégradation et à éliminer les pics de fausse alarme.

- *prédiction temporelle*

En ce qui concerne la prédiction temporelle, nous avons choisi la série temporelle chaotique Mackey-Glass et la prédiction de la sortie en concentration de CO₂ d'un four à gaz. Nous avons amélioré les performances de prédiction du réseau RRFR grâce à une version évoluée de l'algorithme d'apprentissage des k-moyennes. En exploitant la technique Fuzzy Min-Max pour l'initialisation de l'algorithme des k-moyennes, nous pouvons garantir une meilleure stabilité du résultat avec une erreur de prédiction proche du minimum global. Cette initialisation permet également à l'algorithme de converger vers un nombre des centres k garantissant une bonne généralisation en évitant les deux zones de sous-apprentissage et de sur-apprentissage. L'apprentissage du RRFR est ainsi moins dépendant de la phase de paramétrage de l'algorithme d'apprentissage.

- *reproduction de séquences temporelles*

Grâce à la mémoire dynamique du RRFR, l'apprentissage d'une séquence temporelle n'est autre qu'un problème d'interpolation linéaire. Une fois les paramètres des gaussiennes déterminés, les poids de sortie du réseau RRFR sont calculés de telle sorte à résoudre le problème d'interpolation dont le vecteur d'entrée représente l'évolution de la mémoire dynamique, et la sortie représente la séquence temporelle à reproduire. La seule condition est que la séquence temporelle ne soit pas plus longue que la longueur de la mémoire

dynamique du réseau RRFR. L'apprentissage d'une séquence temporelle en utilisant les algorithmes des réseaux globalement récurrents est extrêmement lourd en temps de convergence et en ressources informatique, alors qu'avec le RRFR le calcul des poids de sortie s'effectue par simple inversion matricielle.

Une étude comparative du RRFR par rapport aux structures neuronales dynamiques les plus appropriés pour la surveillance dynamique (R^2BF et $DGNN$) a mis en évidence la supériorité de l'outil proposé, dans tous les secteurs d'application liés à ce domaine.

Enfin, du point de vue de l'exploitation industrielle du concept, un des objectifs de notre travail étant de développer un outil neuronal dynamique, facilement paramétrable via la couche de communication TCP/IP pour des applications de surveillance industrielle temps réel, nous avons alors décomposé le réseau RRFR en fonctions élémentaires. Le réseau RRFR est chargé dans un Automate Programmable Industriel (*API*) et assure une surveillance temps réel. La partie apprentissage et visualisation des variables de surveillance est entièrement gérée à distance grâce au coupleur serveur Web de l'automate. Un expert distant (à l'intérieur ou à l'extérieur de l'entreprise) peut ainsi suivre l'évolution de l'équipement à surveiller et également paramétrer le réseau RRFR à distance. Cette solution profite du traitement temps réel avec la fiabilité du fonctionnement des architectures à automates industriels. L'application de ce concept ouvre des perspectives nouvelles, très intéressantes pour l'externalisation de la maintenance. En effet, beaucoup d'entreprises optent pour ce genre de solution qui leur permet à la fois de mieux maîtriser leur budget maintenance, mais surtout de se recentrer sur leur véritable métier de production.

Une des retombées de ce travail a été marquée par l'exploitation industrielle en collaboration avec la société Avensy Ingénierie, qui a menée au dépôt d'un brevet d'invention⁵².

5.5.2 PERSPECTIVES

a) PERSPECTIVES SCIENTIFIQUES

Malgré les résultats grandement surprenants et prometteurs obtenus par les réseaux de neurones artificiels, ces derniers restent tout de même assez loin d'égaler les capacités sensorielles et surtout de raisonnement d'un expert humain. Nous avons vu que les réseaux de neurones en général, et le RRFR en particulier sont très efficaces dans la détection d'une défaillance, détection d'une dégradation (palier de dégradation), modélisation et prédiction d'une évolution temporelle d'un signal non linéaire ; par contre, la fonction de diagnostic est à notre avis une tâche très complexe, et ne peut être qu'en partie résolue par la technique de reconnaissance des formes. La raison principale est que l'expert humain dans sa mission de tenter de diagnostiquer la cause d'une défaillance de toute une machine ou d'un sous ensemble de cette machine, fait souvent appel à d'autres informations que les valeurs quantitatives (les données capteurs).

La question qu'on peut se poser et qui peut ouvrir deux perspectives complètement antagonistes est de savoir *est-ce que l'on veut à tout prix remplacer l'expert humain afin d'automatiser à 100% cette tâche de diagnostic ?*

⁵² Brevet intitulé : « Système de surveillance temps-réel basé sur les réseaux de neurones dynamiques, accessible à distance par un serveur web », auteurs : D. Racocceanu, R. Zemouri, N. Zerhouni, R. Labourier, dépositaires du brevet : ENSMM (50%) et Avensy (50%), date de dépôt du brevet : 30 septembre 2002, numéro d'enregistrement : 021 2205. Le brevet est actuellement en cours de réactualisation.

Dans l'affirmatif, les recherches s'orienteront dans ce cas plutôt vers les neurosciences et le développement propre des réseaux de neurones artificiels afin de développer des architectures neuronales qui tendent à se rapprocher davantage vers les réseaux de neurones biologiques.

Par contre la deuxième option, qui nous semble la plus intéressante, est de savoir comment faire pour extraire le bon vecteur caractéristique associé à une cause bien précise. Comment fait l'expert humain pour rassembler toutes les informations lui permettant de prendre sa décision ? Nous pensons que des systèmes hybrides de type *Neuro-Flous Temporels* (association des techniques de la logique floue avec les techniques neuronales temporelles) peuvent offrir une piste intéressante, permettant l'extraction et l'exploitation des connaissances supplémentaires empiriques de l'expert humain.

b) PERSPECTIVES D'EXPLOITATION INDUSTRIELLE

Plusieurs perspectives restent à entreprendre concernant l'exploitation industrielle. La première est de tester les capacités dynamiques du réseau *RRFR* chargé dans l'automate. En effet, les capacités du réseau *RRFR* à détecter un palier de dégradation, un pic de fausse alarme ainsi que ses capacités de prédiction n'ont pas été testées sur le prototype de l'automate programmable bien qu'elles soient validées par la simulation. Le problème récurrent à tout test d'un système de détection et de diagnostic de défaillances réside justement dans l'existence de ces défaillances. L'idée de provoquer une défaillance réelle ne serait évidemment pas envisageable de peur de détruire l'équipement. Le plus judicieux serait donc de concevoir un simulateur de pannes. On pourrait ainsi perturber les variables de surveillance, en générant une situation similaire à une défaillance réelle. Par ailleurs, toute une combinaison de défaillances et de dégradations peut être générée pour tester le réseau *RRFR*.

La seconde démarche est bien évidemment d'adapter l'interface homme machine en concordance avec le type d'équipement industriel à surveiller. En effet, l'interface actuelle présente un caractère générique qui nous a servi pour effectuer la validation des fonctions de communication de base et de la structure neuronale. Cette adaptation fait partie de la suite du projet de collaboration entre le *LAB* et la société *AVENSY*.

Chapitre 6 - Surveillance dynamique par réseaux de Petri flous

6.1 INTRODUCTION

Technique importante présentant un intérêt marqué pour la surveillance, le système expert flou fait appel à une base de connaissances incertaines modélisées par l'enchaînement des règles logiques floues. L'utilisation de la logique floue dans la surveillance permet d'affiner les informations acquises sur le terrain. Ceci se fait en introduisant un élément de type « degré de vérité » (réel compris entre 0 et 1) associé à chaque affirmation, ainsi qu'en considérant un ensemble de règles spécifiques (souvent évolutives) propres au système surveillé et à son environnement. Cette technique semble très prometteuse grâce à son potentiel d'utilisation en surveillance dynamique et pronostic, avec la possibilité de rester proche de l'opérateur humain, en prenant en compte sa façon de raisonner et en offrant une traçabilité intéressante pour le retour d'expérience, aspect essentiel de toute fonction de maintenance industrielle.

Dans ce sens, selon les critères définis dans le paragraphe 3.3.3 d), les réseaux de Petri flous font partie des outils de surveillance génériques, faciles à modéliser, capables de prendre en compte l'incertain, et dont la validation est envisageable grâce à l'outil mathématique associé.

Nous proposons ainsi des outils intégrés de type réseaux de Petri flous (RdPF) dédiés à la surveillance et la reprise des systèmes à événements discrets et plus particulièrement des systèmes de production. Un mécanisme d'interfaçage approprié est intégré à chaque outil, afin de rendre possible la mise en place de la boucle système - surveillance - commande. Dans la maintenance industrielle, cette boucle est utilisée en tant que système d'aide à la décision et comprend deux étapes supplémentaires fondamentales : la validation par l'ingénieur de maintenance et la mise en œuvre par un opérateur spécialisé. L'intégration de la logique floue dans un tel système d'aide à la décision permet une formalisation proche du savoir-faire et de la culture d'entreprise ainsi qu'une flexibilité et une dimension prédictive très intéressante.

Il est à noter que les éléments de base présentés dans ce chapitre correspondent aux travaux développés dans la thèse de Mme. Eugenia Minca (**Minca, 2004**), travaux qui ont fait l'objet d'un certain nombre de publications dans des revues et congrès internationaux (**Minca et al., 2002a,b, 2003a,b,c**).

6.2 ETAT DE L'ART DES RESEAUX DE PETRI FLOUS. UTILISATION DES RESEAUX DE PETRI FLOUS POUR LA SURVEILLANCE DES SYSTEMES A EVENEMENTS DISCRETS

L'intégration de l'approche floue dans la surveillance des systèmes à événements discrets permet des descriptions qualitatives des procédés par l'intermédiaire des variables

logiques floues et par leurs valeurs pondérées de vérité, appartenant à l'intervalle $[0,1]$. Un outil capable de modéliser la surveillance floue des systèmes à événements discrets doit intégrer les concepts de la modélisation générique d'un système à événements discrets et les concepts généraux de la logique floue.

Les premières contributions de modélisation des bases de règles logiques correspondent aux approches génériques de modélisation des *règles de production* par des outils de type RdP (Zisman, 1978), (Valette *et al.*, 1987). D'autre part, les travaux de (Murata *et al.*, 1988), montrent la relation entre la dynamique d'un réseau de Petri et la programmation logique (les réseaux de Petri à Objets) (Sibertin-Banc, 1985). Le jeton devient un objet : il est défini par des attributs et des méthodes spécifiques. Ces deux directions ont permis la conception d'un outil capable d'intégrer la logique floue dans la structure du réseau de Petri. Le réseau de Petri flou (RdPF) est un outil très approprié pour l'étude des systèmes à événements discrets décrits par des connaissances incertaines. Un état de l'art très complet des différentes approches existantes a été publié par Cardoso, Valette et Dubois (Cardoso *et al.*, 1996a). La première contribution spécifique appartient à Looney. Même si l'outil proposé (Looney, 1988) est plus proche d'un réseau de neurone, il respecte, dans sa dynamique, les concepts d'un réseau de Petri dédié aux raisonnements logiques flous. Par la suite, quelques approches de réseaux de Petri flous ont été développées à partir des réseaux de Petri ordinaires (Chen *et al.*, 1990), (Pedriycz *et al.*, 1994), (Lipp *et al.*, 1994) ou des réseaux de Petri de haut niveau (Scarpelli *et al.*, 1994), (Gomes *et al.*, 1998), (Camargo, 1998).

		Looney	Chen & al.	Bugarin & al.	Konar & al.	Scarpelli & al.	Cardoso
Raisonnement à base de règle	Imprécision	Non	Ensemble flou	Ensemble flou	Ensemble flou	Ensemble flou	Non
	Incertitude	Degré de crédibilité $[0,1]$	Facteur de certitude	Non	Facteur de certitude	Non	Degré de crédibilité $[0,1]$
Réseau de Petri flou	Emplacement de l'imprécision ?	Non	Places, jetons	Places, jetons	Places, jetons	Places, jetons	Non
	Emplacement de l'incertitude ?	Transitions, jetons	Transitions, jetons	Non	Transitions, jetons	Non	Transitions, jetons
	Les règles d'évolution de la théorie des RdP demeurent-elles sans changement ?	Non (Looney a modifié la règle de franchissement)	Non (La définition de la règle d'évolution pour les places intermédiaires n'est pas conforme)	Oui	Non (Ils ont adopté les modifications apportées par Looney)	Oui	Non
Algorithme d'implémentation	Est-il conforme au raisonnement à base de règles ?	Oui	Oui (Sauf pour les places intermédiaires où ils considèrent que plusieurs règles ne peuvent pas aboutir à une même conclusion)	Oui (La structure du réseau et de l'algorithme dépend du marquage initial)	Oui	Oui	Oui
	Est-il conforme avec l'exécution proposée par le réseau de Petri ?	Non (En raison de la modification de la règle de franchissement)	Oui (Hormis le cas d'une place intermédiaire qui ne peut avoir qu'un seul arc à l'entrée)	Oui (La structure du réseau et de l'algorithme dépend du marquage initial)	Non (En raison de la modification de la règle de franchissement)	Oui	Oui

Tableau 9. Différentes approches des réseaux de Petri flous

Les différents types de logiques (classique, linéaire et floue) utilisées dans la description des systèmes, génèrent deux catégories de modèles RdPF :

- La première catégorie est représentée par les réseaux qui modélisent des raisonnements flous exprimés par des règles floues décrivant le comportement générique d'un système à événements discrets. Les règles floues qui deviennent pour un système de production des règles de production (Valette *et al.*, 1987, 1994), sont mises en correspondance avec les transitions. Le modèle global, correspondant à l'ensemble des règles, modélise un système expert flou. Dans ce cas, le RdPF est interfacé avec le système réel surveillé par des informations provenant des capteurs. Les transitions, qui modélisent l'ensemble des règles simultanément vraies, seront franchies en parallèle et la conclusion

sera le résultat obtenu par défuzzyfication (Hanna, 1994), (Gomes *et al.*, 1995). On reconnaît le mécanisme spécifique à un contrôleur flou, qui, par rapport à un système expert flou, présente une typologie simplifiée.

- L'autre catégorie de RdPF modélise un système dynamique physique. On obtient des classes d'application qui expriment l'incertitude ou le flou des connaissances au niveau des états des ressources du système (Cardoso *et al.*, 1990), (Valette *et al.*, 1989). Ce type de RdPF modélise un système physique par application de la logique linéaire aux transformations de l'état des ressources, en conservant l'incertitude des connaissances concernant leur présence ou leur disponibilité.

Les composantes du RdPF sont adaptées pour la représentation des connaissances incertaines par des variables floues (ensembles flous ou fonctions d'appartenance) associées aux places (Cao *et al.*, 1993), (Bugarin *et al.*, 1994) et/ou aux transitions et/ou aux jetons (Chen *et al.*, 1990), (Pedrycz *et al.*, 1994), (Lipp *et al.*, 1994), (Scarpelli *et al.*, 1994), (Gomes *et al.*, 1995) et/ou au marquage (Cardoso *et al.*, 1990), (Cardoso, 1995), (Valette *et al.*, 1989) et/ou aux arcs (Camargo, 1998), (Lipp *et al.*, 1994). Dans toutes ces approches, comme dans notre étude, la présence du jeton est binaire, même si celui-ci se charge de valeurs floues au moment de son apparition dans la place. Par ailleurs, pour conserver la cohérence des modèles RdPF avec des règles logiques équivalentes (monotonie de la logique classique), il est nécessaire de faire la remarque suivante :

Remarque (Looney, 1988) : Dans un RdPF qui modélise des bases de règles logiques, le franchissement des transitions ne consomme pas les jetons des places antérieures.

Pratiquement, cela peut être modélisé par des arcs qui remettent les jetons dans les places amont après le franchissement d'une transition ou en supposant un nombre infini de jetons dans chaque place. Ce nombre infini de jetons ne sera donc pas modifié ($\infty - 1 = \infty$) après le franchissement de cette transition (Murata *et al.*, 1988), (Cardoso *et al.*, 1996a,b).

6.3 OUTIL DE SURVEILLANCE PAR APPROCHE FLOUE. APPLICATION A LA SURVEILLANCE DYNAMIQUE

6.3.1 INTRODUCTION

L'intérêt principal de la théorie de la logique floue est de permettre la représentation du raisonnement approximatif, qui fait intervenir des connaissances incertaines⁵³, imprécises⁵⁴, ainsi que des connaissances expérimentales et pratiques, habituellement difficiles à modéliser par les techniques de la logique classique.

L'apport de l'utilisation des réseaux de Petri pour représenter la synchronisation des activités des systèmes à événements discrets, n'est plus à démontrer. La prise en compte du temps dans les réseaux est calquée sur le modèle du *réseau de Petri temporel* qui permet une intégration facile des mécanismes conventionnels pour la modélisation de la détection directe et indirecte (chiens de garde, comptes rendus).

⁵³ La notion d'incertain est relative à la vérité d'une information. Elle traite du cas où les éléments de connaissance ont une valeur de vérité connue avec plus ou moins de précision

⁵⁴ La notion d'imprécision porte sur un défaut quantitatif de connaissance. Elle traite le cas où les éléments de connaissance ont un contenu imprécis

La simplicité d'utilisation des RdP pour la détection des défauts dans les systèmes à événement discrets (SED) et la modélisation de la propagation, et l'adaptation de la logique floue pour la représentation des systèmes de surveillance complexes ou mal définis, nous amène à proposer un réseau de Petri flou pour la surveillance.

Les réseaux de Petri flous sont orientés vers la modélisation des raisonnements flous, sur la base des règles logiques statiques (Looney, 1988), (Chen, 1990). C'est pour donner un aspect dynamique à la surveillance des systèmes que nous proposons un *Réseau de Petri Flou pour la Surveillance (RdPFS)*, outil qui utilise une approche *temporelle floue*.

6.3.2 DESCRIPTION ET DEFINITION DE L'OUTIL RdPFS

L'outil de surveillance proposé suppose l'existence d'un modèle du système surveillé (Figure 100). Ce dernier doit intégrer la fonction de *détection directe* (Lefevre, 2000) qui consiste à observer l'apparition des symptômes de défaut signalés par les capteurs de surveillance (Combacau, 1991). La détection directe est mise en correspondance avec les capteurs ou les actionneurs placés dans les points de référence de début/fin des activités.

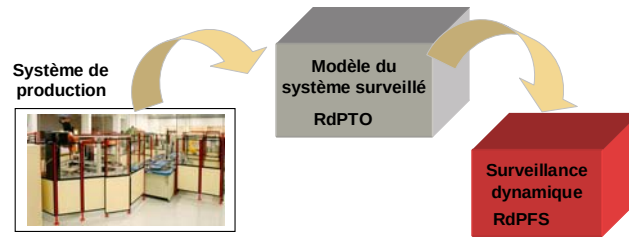


Figure 100. Place des modèles de détection (RdPTO) et de surveillance dynamique (RdPFS)

Dans le cadre de notre étude, la modélisation du système surveillé fait appel à deux *Réseaux de Petri Temporels à Objets (RdPTO)*, sur le modèle de (Combacau, 1991). Un des réseaux modélise les états de fonctionnement correct du système et représente le *réseau de référence*. Ce réseau sert de référence pour le système de commande modélisé par un autre RdPTO appelé *réseau de commande*. Les deux réseaux se trouvent dans une relation de collaboration (Figure 101). L'aspect temporel orienté objet consiste en l'affectation de *l'instant de franchissement* d'une transition en tant qu'attribut fondamental du jeton.

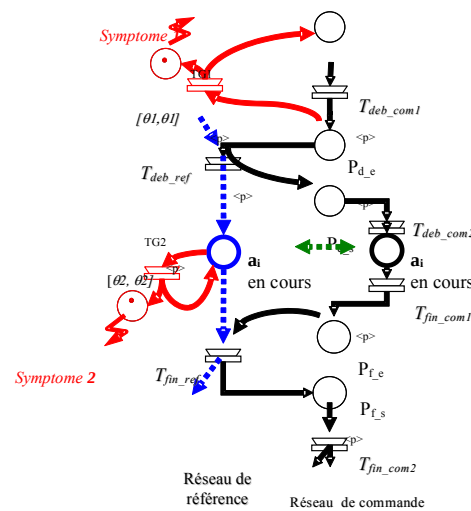


Figure 101. Modélisation du système surveillé par RdPTO. Intégration de la détection directe.

Par rapport aux travaux de Combacau, le modèle RdPTO du système surveillé que nous considérons n'intègre pas de fonctions de surveillance avancées. Cette fonction fait l'objet d'un module RdPF spécifique à la Surveillance dynamique - le RdPFS.

La communication et la synchronisation entre les modèles du système (RdPTO) et le modèle de surveillance (RdPFS) se fait par l'intermédiaire de *symptômes* de type signaux de synchronisation (*notés par !?s*), inspirés des réseaux de Petri à synchronisations internes (Daniel, 1995), (**Racoceanu et al., 2002**). A ces signaux, nous avons rajouté des informations floues concernant l'instant d'apparition du défaut. Au moment de l'arrivée du jeton de type *symptôme* dans la place du RdPTO qui modélise le symptôme de défaut (*chien de garde - CG*), un signal de synchronisation sera émis vers la transition correspondante du RdPFS.

Le RdPFS modélise l'évolution temporelle des défauts, exprimée par un outil descriptif dynamique inspiré de l'arbre de défaillance. La propagation et l'évolution des défauts sont évaluées par des sections temporelles du marquage du RdPFS. Ce modèle est basé sur des informations prédictives qui déclarent a priori certains défauts de l'arbre de défaillance, comme étant critiques dans l'évolution du système. Ces états critiques – identifiés par des places correspondantes du RdPFS –, définissent le *chemin critique des défauts*. Les places critiques sont caractérisées par des seuils de criticité des défauts correspondants. Elles matérialisent l'interface qui permet de déclencher l'action de reprise/maintenance.

Le RdPFS est défini comme étant le n-uplet $\langle P, T, D, I, O, F, ?S, !R, \alpha, \beta, \lambda \rangle$, avec :

$P = \{p_1, p_2, \dots, p_n\}$	ensemble fini des places qui modélisent les défauts possibles du système. Cet ensemble de défauts est structuré en deux catégories : défauts de base et défauts dérivés. Les défauts pris en considération peuvent être aussi bien transitoires que persistants ;
$T = \{t_1, t_2, \dots, t_n\}$	ensemble fini des transitions qui modélisent l'évolution des défauts, en conformité avec un ensemble des règles logiques floues R . Chaque transition est associée à une règle ;
$D = \{d_1, d_2, \dots, d_n\}$	ensemble fini de propositions logiques qui définissent la base des règles R ;
$I : T \rightarrow P$	fonction d'entrée dans les places ;
$O : P \rightarrow T$	fonction de sortie des places ;
$f_i : T \rightarrow F$	fonction qui associe à chaque règle modélisée par une transition, une fonction d'appartenance F qui décrit le degré de crédibilité $\mu_i = F(t)$ de la règle. L'instant t correspond à la détection d'un symptôme de défaut ;
$\alpha_j : P \rightarrow [0,1]$	fonction associative qui établit une valeur de crédibilité α_j (nombre flou) pour les places correspondant aux propositions logiques $d_j \in D$. Ce paramètre représente la possibilité d'apparition du défaut associé à la place p_j ;
$\beta_j : P \rightarrow D$	fonction bijective qui associe les propositions logiques d_j aux places $p_j \in P$;
$\lambda_k : P \rightarrow [0,1]$	fonction qui associe à une place $p_k \in P$, une valeur de seuil λ_k d'acceptabilité/permisivité pour le défaut correspondant, du point de vue de l'action de <i>reprise</i> . λ_k correspond au seuil d'avertissement du paramètre α . Le paramètre λ_k est associé uniquement à certaines places symbolisant les défauts dérivés critiques. Le dépassement de ce seuil nécessite une action de reprise ;
$!S = \{s_1, s_2, \dots, s_l\}$	ensemble des signaux (flous) des symptômes des défauts envoyés (!) ou reçus (?) par le système surveillé respectivement le système de surveillance. Ils arrivent sur les transitions chargées de l'information temporelle « instant t de franchissement » ;
$!R = \{r_1, r_2, \dots, r_l\}$	ensemble des signaux (flous) de reprise émis par le système de surveillance ;
M_0	marquage initial qui correspond aux jetons assignés aux places correspondant aux défauts de base. Par convention, les places associées aux défauts de base ne sont pas représentées dans le modèle global du RdPFS.

Par l'intermédiaire des transitions temporelles, une modélisation de *l'évolution dynamique* de la propagation des défauts est mise en oeuvre. Les signaux de synchronisation avec le modèle (RdPTO) de détection directe, valident le franchissement des transitions, durant les fenêtres temporelles allouées. Chaque implication élémentaire dispose des paramètres suivants :

- La valeur floue α_j représente la modélisation de la gravité du défaut, en fonction de l'instant de son apparition. Cette valeur est élaborée dans le modèle RdPTO.
- La valeur floue μ_i représente le degré de vérité de l'implication logique $d_i \rightarrow d_k$ en fonction de l'instant d'arrivée du signal de défaut. Cette valeur est élaborée dans le modèle RdPFS et elle représente le degré d'appartenance de la variable floue $t - \text{instant d'émission d'un signal de défaut} - \text{à la variable linguistique "apparition des défauts } d_i, d_k \text{"}$.
- L'instant t d'injection du signal flou représente l'instant de franchissement de la transition du RdPFS si elle ne se trouve pas à la concurrence des plusieurs places. Dans ce cas, t représente l'instant de franchissement possible de la transition temporelle ou le début de sa sensibilisation.

Caractéristiques du RdPFS :

- Le RdPFS modélise des raisonnements logiques et pas des transformations de ressources. C'est la raison pour laquelle la conservation des ressources produites/consommées n'est pas respectée et la logique linéaire ne s'applique pas.
- Une place peut être marquée par un seul jeton. Vu qu'une place modélise l'état d'une variable logique, si la place est marquée, la variable logique est vraie et si la place n'est pas marquée, la variable logique associée est fausse. Par ailleurs, vu que le réseau modélise des raisonnements logiques flous, le degré de vérité de chaque variable logique est variable. Ceci est caractérisé par le nombre flou α associé à la place.
- Le franchissement d'une transition ne "consomme" pas les jetons des places précédentes. Ceci est la conséquence du fait que le statut des variables logiques ne change pas si on continue à faire des raisonnements logiques avec ces variables. On affirme dans ce cas que, dans chaque place aval d'une transition franchissable, apparaît un jeton qui consomme des copies des jetons des places situés en amont de la transition. Cette remarque nous permet de modifier l'équation fondamentale d'un réseau de Petri qui ne contiendra pas la matrice Pré.
- Le partage des ressources, concept spécifique aux RdP, perd sa signification dans le cas du RdPFS. Une place marquée peut être "partagée" sans conflit, par plusieurs transitions. En réalité, le sens de partage est incorrect, parce que chaque place peut envoyer une infinité de copies des jetons vers les transitions suivantes.

Dans le système à événements discrets surveillé (modèle RdPTO), l'instant de franchissement t de la transition associée à une activité a_i surveillée (faisant l'objet d'une détection directe) peut être modélisé par la fonction d'appartenance FN correspondant à une activité normale ou par la fonction d'appartenance F (ou F') correspondant à l'apparition d'un défaut d'exécution (Figure 102). La détection directe implantée dans le modèle RdPTO du système, surveille ainsi l'apparition des défauts par un mécanisme de type *chien de garde* (CG). Ce mécanisme envoie un signal $!s_i$, en utilisant une modélisation floue de l'instant de franchissement de la *transition du chien de garde* (TG). Le signal s_i est chargé d'une variable floue $\mu_i = F(t)$ associée au degré de crédibilité de la règle de propagation du défaut d_j vers le défaut dérivé d_k . Dans le réseau RdPFS, le signal s_i est réceptionné ($?s_i$) et une modélisation floue de l'implication $d_j \rightarrow d_k$ permet de calculer la valeur de vérité de la variable logique d_k induite.

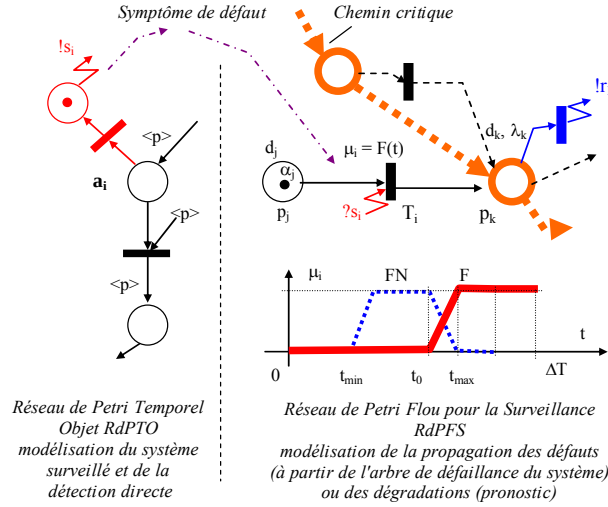


Figure 102. Principe de modélisation de la propagation des défauts

Le point de départ de la propagation des défauts dans le modèle de surveillance RdPFS, est constitué par les *défauts de base* (de type d_j). Leur apparition est signalée par l'ensemble des signaux de synchronisation $!s_j$ émis par le modèle RdPTO du système surveillé. Les *défauts dérivés* (de type d_k) sont des combinaisons logiques de défauts de base et/ou de défauts dérivés antécédents.

Chaque transition T_i du RdPFS correspond à une règle floue qui décrit la propagation des défauts. Elle est associée à la fonction $\mu_i = F(t)$ qui exprime le degré de crédibilité de la règle de propagation des défauts modélisée par la transition concernée. Cette fonction associe à chaque règle une *crédibilité dynamique*.

Chaque place du RdPFS correspond à un défaut (ou une dégradation) du système surveillé. Le marquage d'une place p_j correspondant à un défaut de base, est associé à une valeur floue $\alpha_j \in [0, 1]$ de crédibilité (gravité) du défaut de base observé. La valeur floue α_k , associée au marquage de la place p_k correspondant à un défaut dérivé, est calculée en appliquant le modus ponens généralisé (Chen, 1990). Dans notre approche, nous considérons les opérateurs : $T(u,v) = \min(u,v)$ - norme triangulaire (t-norme) et $\perp(u,v) = \max(u,v)$ - conorme triangulaire (t-conorme), ainsi que l'opérateur modus ponens généralisé $T_{\text{probabiliste}}(u, v) = u \cdot v$. La valeur de crédibilité α_k de la conclusion sera ainsi :

$$\alpha_k = \sup_{t \in T} T(\alpha_j, f_R(\alpha_j, \alpha_k)) = T(\alpha_j, \mu_i) \quad [9]$$

$$\alpha_k = \alpha_j \cdot \mu_i \quad [10]$$

Quand une place p_k appartenant au chemin critique est marquée, un signal $!r_k$ est instantanément envoyé vers le système de reprise (maintenance). Mis à part sa mission de synchronisation avec le sous-système cible, le signal sera chargé de deux informations : le nombre flou α_k associé au marquage de la place et la valeur du seuil λ_k .

Une analyse approfondie des problématiques de modélisation floue de la détection est présentée dans (Minca, 2004).

6.3.3 MODELISATION DES REGLES LOGIQUES DE L'ARBRE DE DEFAILLANCE PAR RDPFS

Comme nous l'avons mentionné, le RdPFS est dédié à la modélisation des règles logiques qui découlent de l'arbre de défaillances (ADD) dans l'objectif d'effectuer une surveillance dynamique.

Une expression logique de type ADD est composée par l'enchaînement logique des variables logiques, par l'intermédiaire des opérateurs de *conjonction* et de *disjonction logique*.

L'outil RdPFS peut modéliser des règles logiques d'ADD en respectant un certain nombre de contraintes évoquées dans (Minca, 2004).

Dans la figure suivante, nous présentons la modélisation de l'expression logique $d_i \wedge d_j \rightarrow d_k$. Cette expression est équivalente à une disjonction de deux expressions logiques élémentaires :

$$d_i \wedge d_j \rightarrow d_k \Leftrightarrow (d_i \rightarrow d_k) \vee (d_j \rightarrow d_k)$$

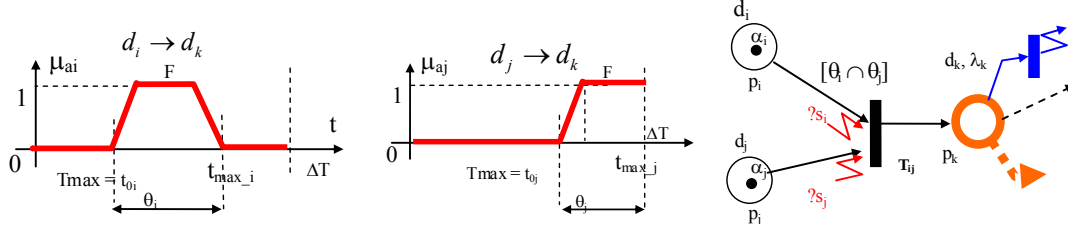


Figure 103. Modélisation de l'implication logique par une transition temporelle

La transition T_{ij} qui modélise l'implication $(d_i \rightarrow d_k)$ peut être franchie dans l'intervalle :

$$\theta_i = [t_{0i}, t_{\max_i}] : \text{le signal de synchronisation avec le RdPTO peut arriver durant } \theta_i.$$

La transition qui modélise l'implication $(d_j \rightarrow d_k)$ peut être franchie dans l'intervalle :

$$\theta_j = [t_{0j}, t_{\max_j}] : \text{le signal de synchronisation avec le RdPTO peut arriver durant } \theta_j.$$

Les signaux $!s_i$, $!s_j$ peuvent arriver dans l'intervalle temporel :

$$[t_{0i}, t_{\max_i}] \cap [t_{0j}, t_{\max_j}] = \theta_i \cap \theta_j \quad [28]$$

6.3.4 EXEMPLE DE MODELISATION EN UTILISANT LE RDPFS

On considère l'expression logique F associée à un arbre de défaillance. L'ensemble des signaux $\{!s_1, !s_2, !s_3, !s_4\}$ sont émis par la détection directe du système surveillé (RdPTO). L'ensemble $\{?s_1, ?s_2, ?s_3, ?s_4\}$ représente les signaux reçus par le système de surveillance modélisé par RdPFS (Figure 104).

$$F = (d1+d2)*d3 + d4*d1 + d3*(d1+d4) \quad [29]$$

Nous pouvons écrire l'expression F en utilisant les opérateurs logiques réunion et intersection :

$$F = \underbrace{\left(\underbrace{(d1 \cup d2)}_{ds1} \cap d3 \right)}_{ds5} \cup \underbrace{(d4 \cap d1)}_{ds4} \cup \underbrace{d3 \cap (d1 \cup d4)}_{ds3} \underbrace{\hspace{10em}}_{ds6}$$

L'hypothèse de disposer d'une description floue, pour un contexte quelconque de défauts simultanés, semble utopique. Dans ce sens, le RdPFS modélise l'expression F en mettant en forme chaque implication élémentaire de type $d_i \rightarrow d_j$. Ceci nous permet de valoriser l'expérience de l'expert qui peut faire des évaluations pour des activités singulières.

En utilisant l'outil RdPFS, nous modélisons l'expression logique F [29] en respectant la correspondance des symboles : « + » pour l'union et « * » pour l'intersection des variables logiques. Pour chaque transition qui modélise une règle correspondant à une opération logique de l'arbre de défaillance, nous avons conçu les fonctions F_i d'appartenance de l'instant t aux classes « apparition des défauts d_i, d_k ». A chaque instant t , on identifie des différents degrés de crédibilité.

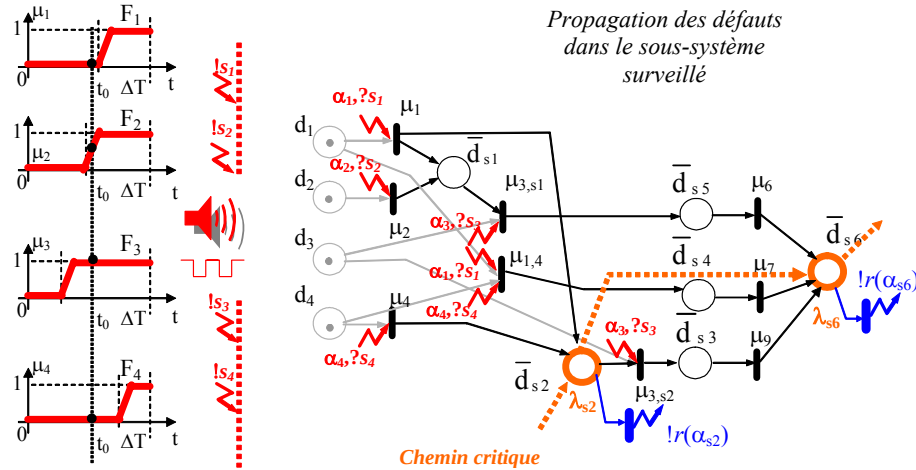


Figure 104. Représentation de la fonction F par le RdPFS

Le franchissement de chaque transition (la règle) consomme les *copies des jetons* des places amont (l'antécédence de la règle) et induit des jetons dans les places aval (la conclusion de la règle). A chaque instant t , une section dans le marquage global du RdPFS indique un front temporel dans le sens de la propagation des défauts.

6.4 ALGORITHME DYNAMIQUE DE SURVEILLANCE PAR APPROCHE FLOUE

Dans l'étude de la dynamique du RdPFS, nous présentons, dans un premier temps, une analyse du comportement des transitions. Par la suite, une analyse temporelle floue de la propagation des défauts sera introduite. Elle donne lieu à un algorithme dynamique d'évaluation des dégradations/défauts du système. Enfin, un exemple d'application met en évidence l'intérêt de cet algorithme.

6.4.1 APPROCHE TEMPORELLE FLOUE POUR LA SURVEILLANCE DES SYSTEMES A EVENEMENTS DISCRETS

L'outil RdPFS est dédié à la modélisation des implications logiques qui enchaînent des variables logiques par l'intermédiaire des opérateurs logiques ET ou OU. Nous avons montré qu'il n'est pas adapté à la modélisation complète des raisonnements logiques. Il est par contre spécialisé dans la surveillance des systèmes de production et, plus particulièrement, la modélisation des règles logiques qui décrivent la propagation des défauts. En utilisant cet outil, on peut donc représenter les typologies suivantes de règles logiques :

- propositions logiques élémentaires de type implication floue : $p \rightarrow q$
- propositions logiques de type implication floue avec conjonction en antécédence :
 $p_1 \wedge p_2 \dots \wedge p_n \rightarrow q$
- propositions logiques de type implication floue avec disjonction en antécédence :
 $p_1 \vee p_2 \dots \vee p_n \rightarrow q$
- propositions logiques de type implication floue avec conjonction en conséquence :
 $p \rightarrow q_1 \wedge q_2 \dots \wedge q_m$
- propositions logiques de type implication floue avec disjonction en conséquence :
 $p \rightarrow q_1 \vee q_2 \dots \vee q_m$

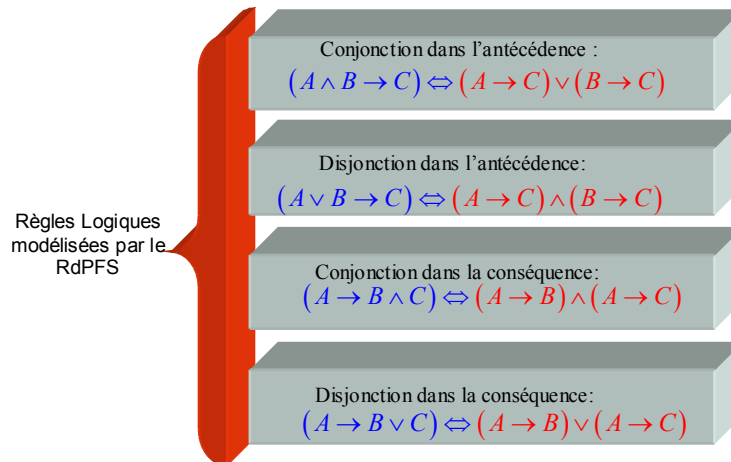


Figure 105. Propositions logiques élémentaires modélisées par le RdPFS

Remarque : L'opérateur négation n'est pas modélisé, car les règles logiques de propagation des défauts prennent en considération la possibilité d'apparition des défauts et pas la possibilité de leur non- apparition.

Pour les propositions logiques complexes de type implication logique, des transformations sont effectuées pour mettre en évidence ces propositions logiques élémentaires.

L'algorithme proposé pour l'élaboration du RdPFS, outil spécialisé dans l'analyse temporelle floue de la propagation des défauts, fait appel aux étapes suivantes :

1. Modélisation du système surveillé par RdPTO
2. Détection directe modélisée par des mécanismes de chien de garde (CG), extensions du modèle RdPTO du système.

3. *Fuzzyfication* de l'instant de franchissement $\tau_{Symptome}$ de la *transition du chien de garde* TG. Les jetons du CG du RdPTO deviennent des objets ayant comme attribut le nombre flou $\mu_d(\tau_{Symptome})$
4. L'apparition des *symptômes* dans les places des mécanismes CG, implique l'*émission des signaux* de synchronisation $!s_i$. Ces signaux, émis par les places du RdPTO, sont réceptionnés comme signaux duaux $?s_i$ par les transitions du RdPFS. Les signaux $!s_i$ sont chargés des informations $\alpha_i = \mu_d(\tau_{Symptome})$ représentant la gravité du défaut (dégradation) enregistré.
5. *Elaboration du modèle du système de surveillance* par RdPFS, eu utilisant l'expression associée à l'ADD. L'expression de l'ADD sera décomposée en règles logiques. La structure du réseau RdPFS traduit l'enchaînement logique des règles de la base des règles (**R**) obtenue.
6. En partant du réseau RdPFS obtenu, on identifie les *implications logiques élémentaires*, modélisées par des transitions. L'implication logique élémentaire met en évidence la "contribution" de chaque variable logique de l'antécédence dans la valeur de vérité de la conséquence.
7. Les implications élémentaires mettent en évidence la relation de *causalité temporelle* entre les deux défauts, relation qui permet d'associer à chaque transition qui modélise la règle, un intervalle de sensibilisation correspondant à la *fenêtre temporelle* $[ti_{min}, ti_{max}]$ pendant laquelle, après l'occurrence d'un défaut, un défaut dérivé est susceptible d'apparaître.
8. Pour chaque implication logique élémentaire, on propose une modélisation floue de la force de liaison entre les variables impliquées. Cette description floue est en fonction de l'instant d'arrivée du $\tau_{Symptome}$. La règle logique modélisée reçoit la valeur de crédibilité :

$$\mu_i = \mu_d(\tau_{Symptome}) \quad [30]$$

9. En appliquant le modus ponens généralisé, on peut calculer la valeur de crédibilité de la conséquence de cette règle :

$$\alpha_k = (\alpha_j \cdot \alpha_i) \cdot \mu_i \quad [31]$$

10. Pour les transitions qui modélisent une implication avec plusieurs variables en antécédence / conséquence, on applique la formule :

$$\alpha_k = [\max/\min(\alpha_j \cdot \alpha_i)] \cdot [\min/\max(\mu_i)] \quad [32]$$

6.4.2 DYNAMIQUE DES TRANSITIONS DANS LE RESEAU DE PETRI FLOU POUR LA SURVEILLANCE

En fonction de l'influence des défauts du système surveillé et de la permissivité d'une action de maintenance (reprise), nous considérons la classification suivante des défauts :

- *Défauts permis* :

Peuvent être corrigés sans arrêter le système et sans interventions sur le système de commande. Ils sont pris en compte par la politique économique comme des défauts admissibles ;

- *Défauts autorisés :*

Caractérisés par un certain degré de risque qui demande une intervention obligatoire au niveau de la commande, pour rendre le système dans un des états dégradés autorisés (Combacau, 1991) ;

- *Défauts interdits :*

Provoquent des états interdits du système. Dans ce cas, une action de reprise est impossible. Pour cette catégorie des défauts, il est souhaitable de mettre en place une approche préventive qui combine la prédiction avec un système de correction.

Si un défaut autorisé ou interdit est en train de se produire, on doit intervenir au niveau de la commande, essentiellement dans les activités qui coopèrent à son apparition. Les places correspondant à ces défauts définissent le *chemin critique des défauts*. Elles sont associées aux seuils d'intervention ou d'avertissement :

$$\lambda : P \rightarrow [0,1] \quad [33]$$

On peut donc définir un *vecteur critique des défauts* Λ , utilisé pour évaluer l'instant d'intervention corrective ou préventive.

En prenant en considération les opérateurs ET et OU – présents en grande majorité dans la plupart des arbres de défaillance – les propositions logiques peuvent être regroupées en quatre catégories : conjonction et/ou disjonction en antécédence et en conséquence. L'outil RdPFS est adapté à ces catégories des problèmes. Dans notre approche temporelle, le seul cas qui pose des problèmes particuliers par rapport à l'approche de Chen (90) est le cas de la conjonction en antécédence. Nous focalisons notre étude sur ce type de transition.

Soit la proposition logique élémentaire : $d_i \rightarrow d_k$ définissant une propagation de défaut. Nous pouvons énoncer la proposition suivante :

Proposition : Si le signal $!s_i$ de type Symptôme, associé à un défaut de base d_i n'est pas envoyé vers le RdPFS, la transition temporelle qui modélise l'implication élémentaire $d_i \rightarrow d_k$ n'est pas franchissable.

La transition qui modélise l'implication logique $d_i \rightarrow d_k$ est une transition temporelle. Si le signal $!s_i$ est envoyé à l'instant τ , il sert de signal de sensibilisation pour la transition temporelle. Une transition qui ne reçoit pas le signal $?s_i$ est une *transition inactive*.

La transition correspondant à un ensemble de places (défauts) concurrentes présente les propriétés suivantes :

Proposition : Les formules qui généralisent le calcul de la valeur floue pour la place de sortie et pour la transition équivalente, s'appliquent uniquement pour les couples places/transitions qui reçoivent les signaux de synchronisation.

Proposition : Si une transition n'a pas au moins un défaut de base en amont, elle ne reçoit pas de signal de type symptôme et le degré de vérité associé est la fonction constante $\mu=1$. Cette transition a le comportement d'une transition ordinaire correspondant à un raisonnement logique classique.

Proposition : Si une transition $t_i \in T$ reçoit en entrée au moins deux signaux s_i et s_j , un coefficient de vérité équivalent ($\mu = \mu_{equiv}$) est associé à la transition des places concurrentes (Figure 106). Cette transition matérialise la conjonction : $d_i \wedge d_j \rightarrow d_k$

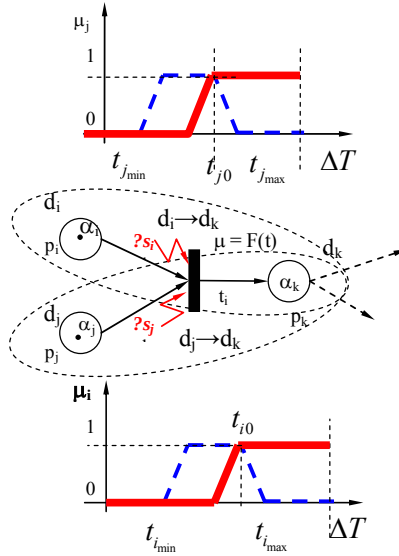


Figure 106. Modélisation des raisonnements logiques concurrents $d_i \wedge d_j \rightarrow d_k$ par RdPFS.

Cette transition temporelle reçoit pendant l'intervalle de sensibilisation associé, les signaux de défaut $?s_i$ et $?s_j$. L'apparition des signaux $?s_i$ et $?s_j$ (associés chacun d'eux à un défaut de base) induit des valeurs différentes μ_i , μ_j à la transition :

$$d_i \rightarrow d_k (CV = \mu_i) \Rightarrow \alpha_k = \alpha_i \cdot \mu_i$$

$$d_j \rightarrow d_k (CV = \mu_j) \Rightarrow \alpha_k = \alpha_j \cdot \mu_j$$

Un degré de vérité global doit ainsi être calculé pour la transition qui modélise la concurrence de ces places. La formalisation de ce degré de vérité global passe par l'analyse du réseau de Petri flou qui matérialise l'expression logique équivalente :

$$d_i \wedge d_j \rightarrow d_k \Leftrightarrow \left(d_i \xrightarrow{\mu_i} d_k \right) \vee \left(d_j \xrightarrow{\mu_j} d_k \right); \quad (CV = \mu)$$

Pour la transition équivalente, le degré de crédibilité global μ sera par conséquent :

$$\mu = \mu_i \vee \mu_j = \max(\mu_i, \mu_j) \quad [34]$$

ce qui permet d'obtenir le degré de crédibilité de la conséquence :

$$\alpha_k = \mu \cdot \min(\alpha_i, \alpha_j) = \max(\mu_i, \mu_j) \cdot \min(\alpha_i, \alpha_j) \quad [35]$$

Remarque : Nous proposons une approche basée sur l'hypothèse de la seule connaissance des propositions élémentaires de type implication logique : $d_i \rightarrow d_k$. Le RdPFS permet ainsi de modéliser des expressions logiques complexes, en traitant des implications logiques élémentaires, beaucoup plus simples à appréhender.

On généralise ce raisonnement pour une transition qui a plusieurs (m) places amont qui modélisent des défauts de base $\bar{d}$ et plusieurs (n) places aval qui modélisent des défauts dérivés $\bar{d}$. En conformité avec une base de règles R, la structure générale des enchaînements logiques des défauts concurrents est la suivante :

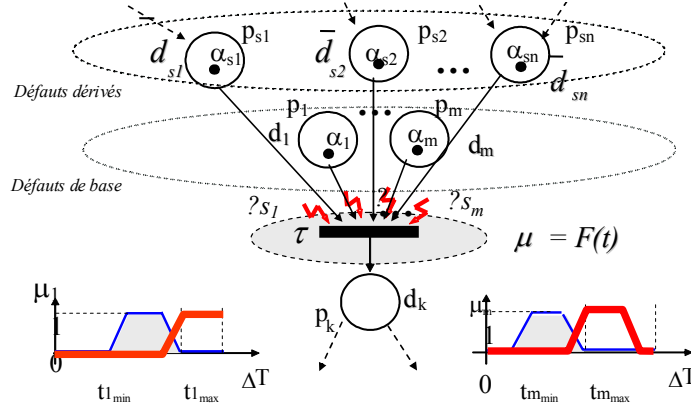


Figure 107. Modélisation des défauts concurrents en antécédence par RdPFS

Proposition : Le coefficient de vérité μ correspondant à la transition associée à l'ensemble des défauts concurrents $\mathbf{D}_i \subset \mathbf{D}$ prend la valeur :

$$\mu = \max(\mu_1, \mu_2, \dots, \mu_m, \mu_{s1}, \dots, \mu_{sn}) \quad [36]$$

si la transition a des places d'entrée qui modélisent des défauts de base, et la valeur :

$$\mu = 1 \quad [37]$$

si la transition est une transition intermédiaire et se déroule sans restrictions temporelles. La valeur de crédibilité de la conclusion (Chen, 90) est donnée par l'expression suivante :

$$\alpha_k = \mu \cdot \min(\alpha_1, \alpha_2, \dots, \alpha_m, \bar{\alpha}_{s1}, \bar{\alpha}_{s2}, \dots, \bar{\alpha}_{sn}) \quad [38]$$

D'une manière similaire, nous renvoyons le lecteur vers (Minca, 2004) pour des propositions concernant la propagation des degrés de crédibilité des défauts disjonctifs en antécédence, ainsi que des défauts disjonctifs et conjonctifs en conséquence.

Nous supposons qu'il existe au moins un défaut qui peut induire un défaut interdit du système surveillé. Ceci implique d'intervenir au niveau de la commande, essentiellement dans les activités qui coopèrent à l'apparition de ce défaut. Il faut donc faire le pronostic des états critiques qui exigent l'action de *maintenance* du système. C'est le rôle des paramètres λ représentant des seuils d'acceptabilité pour certains défauts $\lambda : P \rightarrow [0,1]$

Les défauts sont différents du point de vue de leur gravité : l'importance d'un défaut comme « le chariot en panne » n'est pas comparable avec celle du défaut « outil cassé ». Cette caractéristique est reflétée par les valeurs floues α_i associées à chaque défaut et donc implicitement à chaque place du RdPFS. On a montré que ces valeurs sont variables dans le temps et en fonctions de l'occurrence des différents défauts.

Pour un défaut qui fait partie de la catégorie des défauts interdits, on attribue un seuil plus petit que pour un défaut qui fait partie de la classe des défauts permis ou autorisés. Si

le seuil est dépassé, l'action de *maintenance* sera déclenchée. Sinon, le système de surveillance évolue sans intervention corrective.

6.4.3 ALGORITHME D'EVALUATION DYNAMIQUE DES DEFAUTS

En partant de l'algorithme d'élaboration du RdPFS, présenté dans le paragraphe précédent, nous proposons un outil algébrique capable de représenter la propagation temporelle des défauts dans le système surveillé.

La dynamique du marquage du réseau RdPFS, ne peut pas être évaluée en appliquant l'équation fondamentale du RdP générique. Ceci est dû au fait que le réseau RdPFS comporte un aspect atypique du point de vue de la propagation des valeurs de crédibilité. Nous nous intéressons donc à l'algorithme d'évolution du marquage du RdPFS, en fonction des contraintes temporelles liées à chaque transition, ainsi que des signaux de synchronisation du RdPFS avec le réseau dual RdPTO.

Exécution du RdPFS

Pour montrer l'évolution du RdPFS, nous proposons deux directions de calcul :

- évolution du marquage – le vecteur de marquage M
- détermination des degrés de crédibilité des places correspondantes au nouveau marquage – le vecteur MU

On utilise les deux opérateurs de l'algèbre Max :

- $\oplus$: l'opérateur qui, pour deux variables matricielles $m \times n$, choisit la valeur maximum entre les éléments de chaque position :

$$A \oplus B = D \quad [39]$$

$$d_{ij} = \max \{a_{ij}, b_{ij}\}$$

- $\otimes$: l'opérateur qui, pour deux variables matricielles A et B de dimensions $(m \times p)$ et $(p \times n)$, choisit la valeur maximum entre les produits des éléments deux, sur la ligne/colonne p (la matrice D aura les dimensions $(m \times n)$) :

$$A \otimes B = D \quad [40]$$

$$d_{ij} = \max_{1 \leq k \leq p} (a_{ik} \cdot b_{kj})$$

On considère l'équation fondamentale du réseau de Petri :

$$M = M_0 + (Post - Pre) \cdot \bar{s} \quad [41]$$

qui permet de déterminer l'évolution du marquage pour un réseau avec :

M_0	→	Marquage initial
$Post$	→	Matrice qui décrit les relations entre les transitions et les places aval
Pre	→	Matrice qui décrit les relations entre les transitions et les places amont
s	→	Séquence des transitions franchies.

En considérant [41] et l'hypothèse que les jetons, après le franchissement d'une transition du RdPFS, restent dans leurs places, on supprime la matrice *Pre*. La matrice *Post* a les dimensions $(n \times t)$, avec :

$$n = \text{card}(\cup d_i) + \text{card}(\cup \overline{d_j}) \quad [42]$$

avec $\cup d_i$ et $\cup \overline{d_j}$ les défauts de base respectivement dérivés et t le nombre des transitions.

Pour préciser la séquence des transitions qui seront franchies à l'instant t , on doit construire le vecteur S qui décrit la dépendance de la validation de chaque transition, des signaux $?s_i$ qui sont attendus dans sa précédence. Le vecteur S reflète la base de règles logiques déduites de l'expression logique de l'ADD. Ce vecteur a les dimensions $(t \times 1)$. Il exprime l'idée que l'occurrence d'un défaut est possible si certains défauts, temporellement précédents, se sont produits.

En respectant l'hypothèse que les jetons restent dans leur place après le franchissement d'une transition, l'équation fondamentale du RdP devient :

$$M = M_0 \oplus (Post \otimes S) \quad [43]$$

Les valeurs de crédibilité pour chaque défaut se trouvent en dépendance directe des valeurs (α_i, μ_i) injectées par les signaux $?s_i$. En fonction des considérations faites dans le paragraphe 3.1., on calcule pour chaque place du RdPFS, la valeur analytique de α_i . Par la suite, on définit la matrice A qui contient sur la diagonale principale, les valeurs α_i :

$$A' = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \vdots \\ \alpha_i \end{bmatrix}; \quad A = \text{diag}(A') \quad [44]$$

Par ailleurs, on propose la construction de la matrice MD qui fait apparaître le vecteur du marquage M sur la diagonale principale. On peut donc déterminer les valeurs floues, correspondant à l'état de crédibilité de chaque variable logique modélisée par une place, par l'intermédiaire de l'équation :

$$MU = A \cdot \text{diag}(M_0 \oplus (Post \otimes S)) \quad [45]$$

$$MU = A \cdot MD \quad [46]$$

L'algorithme pour l'évaluation dynamique des défauts produits est le suivant :

Construction du RdPFS associé à l'arbre de défaillance du système surveillé ;

Identification du vecteur des défauts de base et dérivés ;

Identification du vecteur critique des défauts

$$\Lambda = [\lambda_1; \lambda_2; \dots; \lambda_i], \quad p_i \in \{\text{chemin critique}\}$$

Par hypothèse, le *marquage initial* M_0 du RdPFS est nul dans les places correspondant aux défauts dérivés, en considérant qu'au début de son évolution, le système est dans un état de fonctionnement normal (sans défaillance). Les places marquées sont celles correspondant aux défauts de base, seules causes potentielles de déclenchement de l'état défaillant du système surveillé. Les événements discrets responsables de l'évolution du marquage du système, sont les signaux de synchronisation avec la détection directe.

Détermination du vecteur S des signaux conditionnels pour les transitions;

Détermination du vecteur A avec les expressions générales des valeurs floues correspondant à la chaque place;

Calcul du marquage M du RdPFS en utilisant la formule [43];

- Calcul du vecteur MU des valeurs floues pour le RdPFS avec [46] :
- Comparaison de v avec le *vecteur critique des défauts* A afin d'évaluer l'opportunité de la maintenance.

6.5 EXEMPLE DE SURVEILLANCE DYNAMIQUE PAR RDPFS.

Pour illustrer l'application de l'algorithme l'évaluation dynamique des défauts dans le RdPFS, nous reprenons l'exemple de l'ADD (paragraphe 6.3.4) :

$$F = ((d1 + d2) \times d3) + (d1 \times d4) + (d3 \times (d1 + d4)) \quad [47]$$

$$D = \bigcup d_i = \{d1, d2, d3, d4\}$$

$$\overline{D} = \bigcup \overline{d_{si}} = \{\overline{ds1}, \overline{ds2}, \overline{ds3}, \overline{ds4}, \overline{ds5}, \overline{ds6}\}$$

En appliquant l'algorithme d'évaluation dynamique des défauts, nous obtenons :

Construction du RdPFS associé à l'expression [47] :

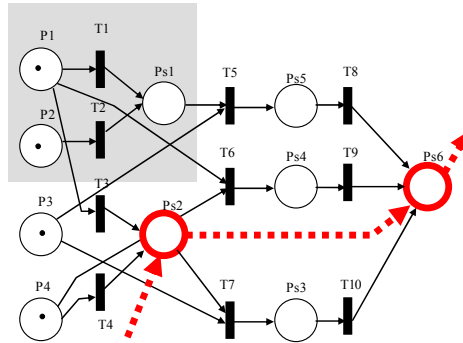


Figure 108. Modélisation de la surveillance séparée par le RdPFS

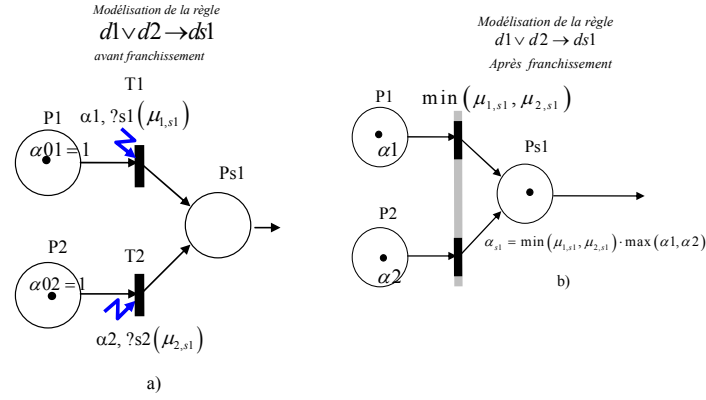


Figure 109. Modélisation de la règle $d1 \vee d2 \rightarrow ds1$ par RdPFS avant a) et après b) franchissement

Identification du vecteur des défauts de base et dérivés :

$$D \cup \overline{D} = \{d1, d2, d3, d4, \overline{ds1}, \overline{ds2}, \overline{ds3}, \overline{ds4}, \overline{ds5}, \overline{ds6}\}$$

Identification du vecteur critique des défauts

$$\Lambda = [0; 0; 0; 0; 0; \lambda_{s2}; 0; 0; 0; \lambda_{s6}], \{p_{s2}, p_{s6}\} \in \{\text{chemin critique}\}$$

Marquage initial.

$$m0 = [1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad [48]$$

Détermination du vecteur **A** des valeurs floues correspondant à la chaque place ;

$$A = \begin{bmatrix} \alpha1(t) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha2(t) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha3(t) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha4(t) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha1s(t) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha2s(t) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha3s(t) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha4s(t) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha5s(t) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha6s(t) \end{bmatrix}$$

Nous proposons la base de règles suivante, décrivant l'enchaînement des défauts associés à l'expression logique F :

$$\left\{ \begin{array}{ll} R1(\mu_{R1}): & d1 \vee d2 \rightarrow \overline{ds1} \\ R2(\mu_{R2} = \mu_{3,s1}): & \overline{ds1} \wedge d3 \rightarrow \overline{ds5} \\ R3(\mu_{R3} = \mu_{1,4}): & d4 \wedge d1 \rightarrow \overline{ds4} \\ R4(\mu_{R4}): & d4 \vee d1 \rightarrow \overline{ds2} \\ R5(\mu_{R5} = \mu_{3,s2}): & ds2 \wedge d3 \rightarrow \overline{ds3} \\ R6(\mu_{R6}): & ds5 \vee ds4 \vee ds3 \rightarrow \overline{ds6} \end{array} \right. \quad [49]$$

En étudiant chacune des règles, nous pouvons écrire les expressions des degrés de crédibilité. A titre d'exemple, pour la règle R_1 , nous obtenons :

$$R1 \begin{cases} d1 \vee d2 \rightarrow \overline{ds1} \\ (d1 \rightarrow \overline{ds1}) \wedge (d2 \rightarrow \overline{ds1}) \\ \mu_{R1} = \min(\mu_{1,s1}, \mu_{2,s1}) \end{cases}$$

$$\Rightarrow \alpha_{s1} = \max(\alpha_1, \alpha_2) \cdot \min(\mu_{1,s1}, \mu_{2,s1}) \quad [50]$$

En exploitant les autres règles, nous pouvons écrire :

$$\Rightarrow \alpha_{s5} = \min(\alpha_{s1}, \alpha_3) \cdot 1 \quad [51]$$

$$\alpha_{s5} = \min\left(\left(\max(\alpha_1, \alpha_2) \cdot \min(\mu_{1,s1}, \mu_{2,s1})\right), \alpha_3\right)$$

$$\Rightarrow \alpha_{s4} = \min(\alpha_1, \alpha_4) \cdot \max(\mu_{4,s4}, \mu_{1,s4}) \quad [52]$$

$$\Rightarrow \alpha_{s2} = \max(\alpha_1, \alpha_4) \cdot \min(\mu_{4,s2}, \mu_{1,s2}) \quad [53]$$

$$\Rightarrow \alpha_{s3} = \min(\alpha_{s2}, \alpha_3) \quad [54]$$

$$\alpha_{s3} = \min\left(\max(\alpha_1, \alpha_4) \cdot \min(\mu_{4,s2}, \mu_{1,s2}), \alpha_3\right)$$

$$\Rightarrow \alpha_{s6} = \max(\alpha_{s3}, \alpha_{s4}, \alpha_{s5}) \quad [55]$$

Détermination du vecteur S des signaux conditionnels pour les transitions;

$$S = \begin{bmatrix} s1 \\ s2 \\ s1 \\ s4 \\ (s1+s2) \times s3 \\ s1 \times s4 \\ (s1+s4) \times s3 \\ (s1+s2) \times s3 \\ s1 \times s4 \\ (s1+s4) \times s3 \end{bmatrix} \begin{matrix} \mathbf{T1} \\ \mathbf{T2} \\ \mathbf{T3} \\ \mathbf{T4} \\ \mathbf{T5} \\ \mathbf{T6} \\ \mathbf{T7} \\ \mathbf{T8} \\ \mathbf{T9} \\ \mathbf{T10} \end{matrix} \quad [56]$$

Calcul du marquage M du RdPFS en utilisant la formule [43];

Matrice Post :

$$Post = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad [57]$$

Evolution du marquage :

$$m = m0 \oplus (Post \otimes S); \quad [58]$$

a) Calcul du vecteur **MU** des valeurs floues pour le RdPFS avec [46] :

$$m' = diag(m); \quad MP = m'$$

$$MU = A \cdot MP \quad [59]$$

Dans le cas le plus défavorable, quand tous les quatre défauts de base se sont produits, nous obtenons ainsi :

$$MU = \begin{bmatrix} \alpha1(t) \\ \alpha2(t) \\ \alpha3(t) \\ \alpha4(t) \\ \alpha s1(t) \\ \alpha s2(t) \\ \alpha s3(t) \\ \alpha s4(t) \\ \alpha s5(t) \\ \alpha s6(t) \end{bmatrix} \quad [60]$$

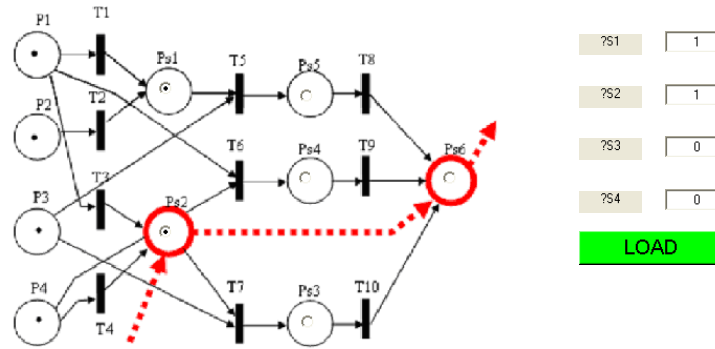
Le vecteur **MU** contient les coefficients de vérité associés à chaque défaut possible. Il indique en même temps leur dépendance du temps, donc le caractère variable des informations.

b) Comparaison de v avec le vecteur *critique des défauts* **A** afin d'évaluer l'opportunité de la maintenance.

Une interface a été construite pour montrer l'évolution du marquage du RdPFS en fonction des signaux flous envoyés par la détection directe :

Test – Signaux de détection directe $[1 \ 1 \ 0 \ 0]^T$

Detection directe	Vecteur S	Marquage RdPFS	Vecteur de valeurs floues
1	1	1	0.950129285
1	1	1	0.231138514
0	1	1	0.606842584
0	0	1	0.485982469
	0	1	0.891298966
	0	1	0.762096833
	0	0	0
	0	0	0
	0	0	0
	0	0	0



Les informations en provenance des places critiques du RdPFS constituent un déclencheur pour des actions de maintenance préventive. Ceci est réalisé par l'intermédiaire d'un outil de la même famille, capable de modéliser l'expérience humaine dans ce domaine.

6.6 ACTION DE MAINTENANCE EN UTILISANT UNE BASE DE REGLES FLOUES

6.6.1 APPROCHE FLOUE DANS LA FONCTION DE MAINTENANCE

L'action de maintenance basée sur une action de surveillance utilisant la logique floue, réclame un nouvel outil capable d'intégrer les informations temporelles floues émises par le système de surveillance RdPFS. L'outil proposé doit élaborer une action anticipative de correction par rapport à une base de règles logiques floues. Notre proposition suppose qu'aucun modèle n'est disponible pour décrire les relations de cause à effet. L'approche floue s'utilise dans les cas où le modèle mathématique est absent ou difficile à trouver. La seule connaissance repose sur l'expertise humaine confortée par un solide retour d'expérience. Dans cette catégorie, on retrouve les méthodes basées sur l'intelligence artificielle.

L'outil que nous proposons dans (Minca, 2004) porte le nom de Réseau de Petri Flou pour la Maintenance (RdPFM). Il correspond à un système expert dans le domaine de la maintenance.

L'indépendance entre la base de connaissances et le moteur d'inférences est un élément essentiel des systèmes experts. Elle permet une représentation des connaissances sous forme purement déclarative, c'est-à-dire sans lien avec la manière dont ces connaissances sont utilisées. L'avantage de ce type d'architecture est qu'il est possible de faire évoluer les connaissances du système sans avoir à agir sur le mécanisme de raisonnement.

La qualité d'un système expert est définie par la richesse de l'expérience prise en compte ainsi que par la richesse des scénarios considérés. Dans notre cas, le modèle de type RdP est interfacé avec les informations temps réel envoyés par le modèle de commande du système surveillé. Il est plus flexible et moins compliqué dans la définition des relations entrées-sorties. L'intégration de l'aspect flou lui confère une certaine flexibilité et plus de finesse dans l'évaluation des différents contextes de défaillances.

L'outil RdPFM proposé, fonctionne en relation de synchronisation avec le modèle pour la surveillance RdPFS. Les informations attachées aux signaux flous, décrivent l'état des

variables logiques associées aux places du chemin critique du modèle de surveillance RdPFS (Figure 110). Cette approche correspond à une vision en boucle fermée proposée pour modéliser les actions détection directe - analyse - maintenance préventive.

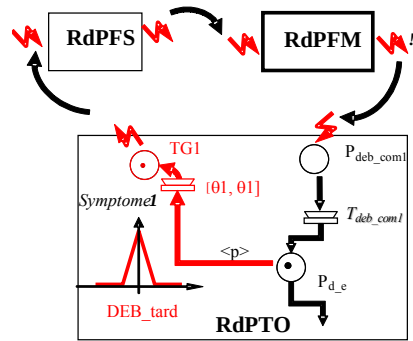


Figure 110. Structure globale de la boucle détection/maintenance préventive

Nous renvoyons le lecteur soucieux de plus de détails sur ce développement vers la thèse de E. Minca (Minca, 2004).

6.7 APPLICATION TEMPS REEL DE SURVEILLANCE DANS UNE SOLUTION LOGICIELLE DE E-MAINTENANCE

6.7.1 SYSTEME ETUDIE

Pour illustrer l'intérêt de l'application de nos outils pour la supervision industrielle, nous avons choisi l'exemple d'un système flexible de production de l'Institut de Productique de Besançon (voir description paragraphe 7.3.4).

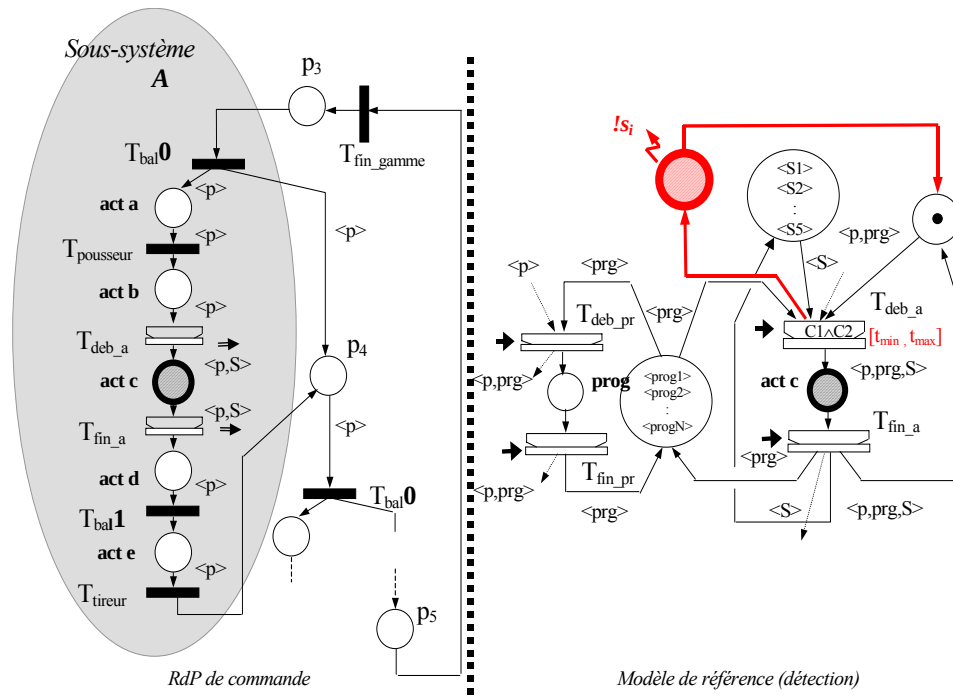


Figure 111. Modélisation d'une station de travail par RdPTO

Pour implémenter un superviseur flou associé au sous-système correspondant à une station de travail, nous avons modélisé chaque station en utilisant deux réseaux de Petri coopératifs de type RdPTO (Figure 111) : un réseau pour la surveillance/commande et l'autre pour modéliser les états du système physique (réseau de référence).

6.7.2 IMPLEMENTATION DES FONCTIONS DETECTION ET SURVEILLANCE DYNAMIQUE

Afin d'illustrer notre démarche, nous considérons l'expression logique F de l'arbre de défaillance « palette bloquée au pousseur » associé au système flexible (**Proos, 2001**) :

$$F = [(a+b+c+d)* e] + b + c \quad [61]$$

Les opérateurs + et * représentent l'union ou l'intersection des variables logiques {a, b, c, d, e}. Elles ont la signification suivante :

- a = 'pousseur bloqué',
- b = 'stoppeur S1 bloqué en bas',
- c = 'détecteur D1 en panne',
- d = 'poste saturé (D4=1)',
- e = 'le poste étudié est bien le destinataire de la palette'.

L'expression booléenne F peut être modélisée par des blocs logiques ou elle peut être traduite par une base de règles floues :

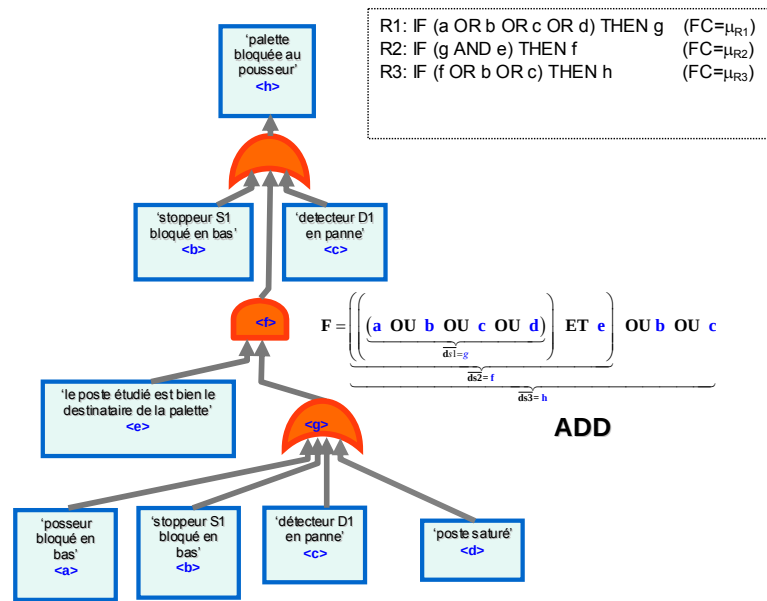


Figure 112. a) Représentation de l'expression booléenne F ; b) Base de règles logiques floues correspondant à F

Le RdPFS qui modélise l'évolution des défauts dans l'arbre de défaillance étudié est représenté dans la Figure 113. L'occurrence du défaut 'a' sera accompagnée par un signal de synchronisation avec le RdPTO au niveau de la transition T1. D'autre part, l'occurrence du défaut 'b' sera signalée par l'injection des deux signaux dans RdPFS qui valident les transitions T2 et T7. Les places P3, P4 et P5 sont mises en correspondance avec les défauts

c, d et e. Le marquage initial du réseau (les places P1:P5) signifie les défauts de base qui peuvent apparaître.

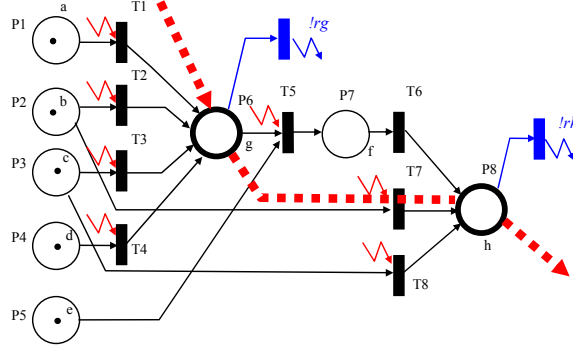


Figure 113. Représentation de l'expression F par le RdPFS

En appliquant l'algorithme présenté précédemment, l'analyse temporelle du RdPFS obtenu permet de définir le vecteur des défauts de base et dérivés : $\langle a, b, c, d, e, g, f, h \rangle$; le vecteur critique des défauts $\Lambda = [\lambda_g, \lambda_h]$, $\{P6, P8\} \in \{\text{chemin critique}\}$, la base de règles logiques $R = \{R1, R2, R3\}$ permettant de calculer les valeurs de crédibilité des défauts dérivés :

$$R1 = \begin{cases} a \vee b \vee c \vee d \rightarrow g \Leftrightarrow \\ (a \rightarrow g) \wedge (b \rightarrow g) \wedge (c \rightarrow g) \wedge (d \rightarrow g) \\ \mu_{R1} = \min(\mu_{ag}, \mu_{bg}, \mu_{cg}, \mu_{dg}) \end{cases} \quad [62]$$

$$\Rightarrow \alpha_g = \max(\alpha_a, \alpha_b, \alpha_c, \alpha_d) \cdot \mu_{R1}$$

$$R2 = \begin{cases} g \wedge e \rightarrow f \Leftrightarrow \\ (g \rightarrow f) \vee (e \rightarrow f) \\ \mu_{R2} = \max(1, \mu_{ef}) = 1 \end{cases} \quad [63]$$

$$\Rightarrow \alpha_f = \min(\alpha_g, \alpha_e)$$

$$R3 = \begin{cases} f \vee b \vee c \rightarrow h \\ (f \rightarrow h) \wedge (b \rightarrow h) \wedge (c \rightarrow h) \\ \mu_{R3} = \min(1, \mu_{bh}, \mu_{ch}) \end{cases} \quad [64]$$

$$\Rightarrow \alpha_h = \max(\alpha_f, \alpha_b, \alpha_c) \cdot \mu_{R3}$$

donnant le vecteur A des valeurs floues correspondant à la chaque place:

$$A = \begin{bmatrix} \alpha_a \\ \alpha_b \\ \alpha_c \\ \alpha_d \\ \alpha_e \\ \alpha_g \\ \alpha_f \\ \alpha_h \end{bmatrix} \text{ et } \begin{bmatrix} \alpha_a \\ \alpha_b \\ \alpha_c \\ \alpha_d \\ \alpha_e \\ \alpha_g \\ \alpha_f \\ \alpha_h \end{bmatrix} = \begin{bmatrix} \alpha_a \\ \alpha_b \\ \alpha_c \\ \alpha_d \\ \alpha_e \\ \min(\mu_a, \mu_b, \mu_c, \mu_d) \cdot \max(\alpha_a, \alpha_b, \alpha_c, \alpha_d) \\ 1 \cdot \min(\alpha_e, \alpha_g) \\ \min(\mu_b, \mu_c) \cdot \max(\alpha_b, \alpha_c, \alpha_f) \end{bmatrix}$$

Le vecteur de marquage du RdPFS :

$$M = M_0 \oplus (Post \otimes S)$$

$$M = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} \min(s1, 1) \\ \min(s2, 1) \\ \min(s3, 1) \\ \min(s4, 1) \\ \min((s1+s2+s3+s4) \times s5, 1) \\ \min((s1+s2+s3+s4) \times s5, 1) \\ \min(s2, 1) \\ \min(s3, 1) \end{bmatrix}$$

permettant ainsi d'aboutir au vecteur des valeurs floues du RdPFS

$$MU = A \cdot \text{diag}(M_0 \oplus (Post \otimes S))$$

qui sera lui-même comparé au *vecteur critique des défauts* $\mathbf{A}$, afin d'évaluer l'opportunité d'actions de maintenance curative.

6.7.3 IMPLEMENTATION DE LA FONCTION DE MAINTENANCE (REPRISE)

Le chemin critique indique les places (g respectivement h) qui envoient des signaux flous vers le RdPFM (Figure 114) modélisant le système expert flou (SEF).

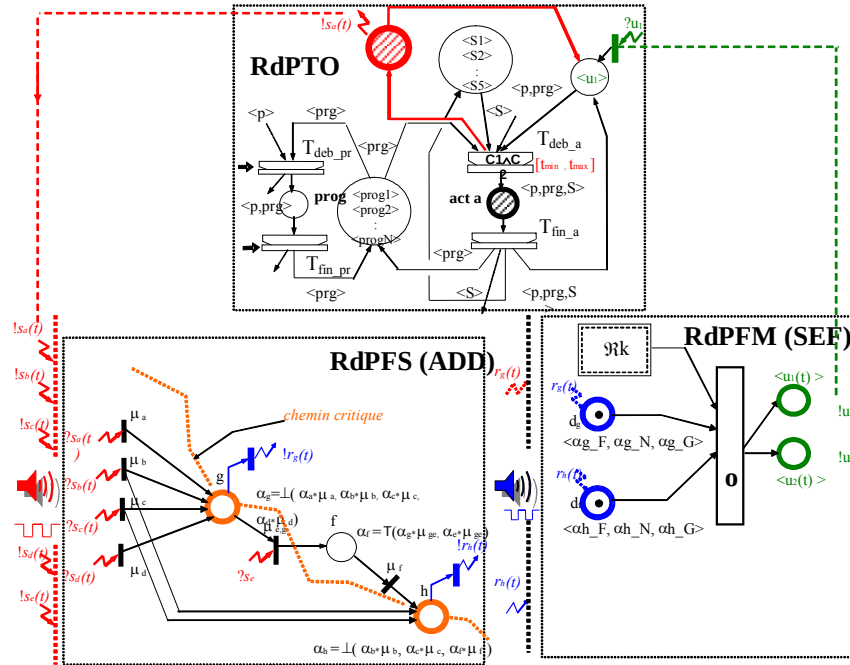


Figure 114. Modélisation de l'ensemble du système temps-réel de surveillance-commande (ADD = arbre de défaillance, SEF = système expert flou)

Ces signaux flous peuvent déclencher la maintenance si et seulement si toutes les places concurrentes avec la base de règles $\mathfrak{R}$ positionnées en amont de la transition « $\circ$ », ont reçu les signaux $!r_i$ de synchronisation avec le RdPFS. Les variables de sortie u_1 et u_2 représentent les commandes de maintenance. Le mécanisme de maintenance modélisé par RdPFM constitue l'implémentation d'une base de règles logiques floues :

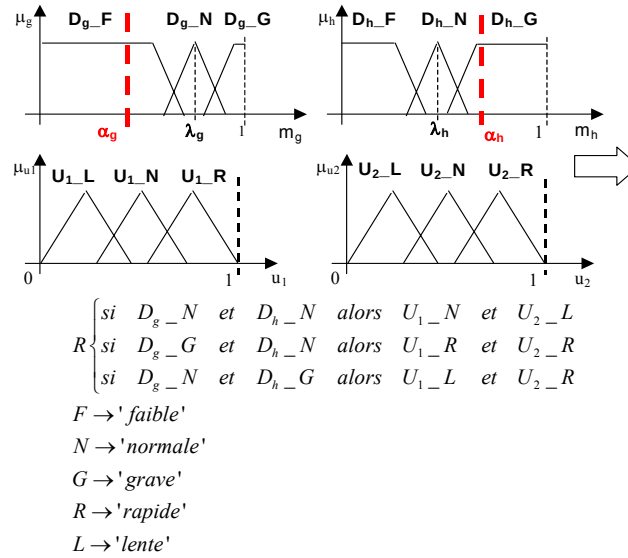


Figure 115. Base des règles floues pour la maintenance

6.7.4 INTEGRATION DE L'APPROCHE GLOBALE DE SURVEILLANCE FLOUE DANS UNE PLATE-FORME LOGICIELLE

Afin de permettre une utilisation des concepts proposés de façon transparente pour l'utilisateur, dans le cadre d'une plate-forme d'aide à la décision en ingénierie de maintenance, nous avons conçu une plate-forme logicielle capable d'intégrer l'approche de supervision floue développée précédemment.

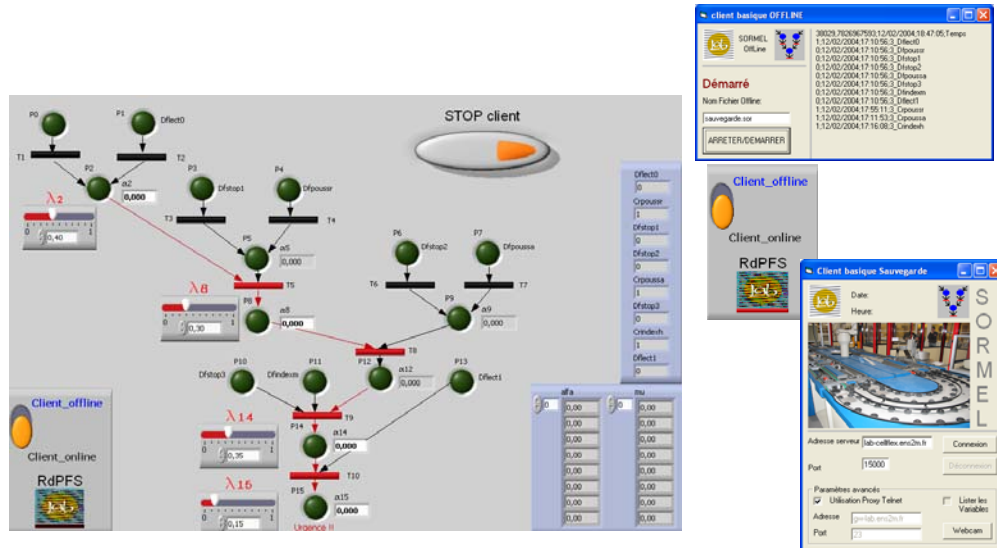


Figure 116. Interface du système de surveillance dynamique

Les signaux de défaut envoyés par le modèle du système, sont des variables acquises par les fonctions d'acquisition spécifiques. Nous avons testé le logiciel dans le contexte d'un scénario qui correspond à l'occurrence du défaut $b =$ 'stoppeur S1 bloqué en bas' (le signal ?S1 sera envoyé par la détection directe). Cette fois on a supposé des valeurs aléatoires pour les variables floues α_a , α_b , α_c , α_d , α_e qui décrivent le degré de certitude (ou

la possibilité) d'apparition de ces défauts. Seuls les signaux $s_1 = !r_g$ et $s_2 = !r_h$, appartenant au chemin critique du RdPFS, sont envoyés vers le RdPFM à l'instant t .

Pour les défauts g et h , ainsi que pour u_1 , nous proposons une description par fonctions d'appartenance aux classes floues :

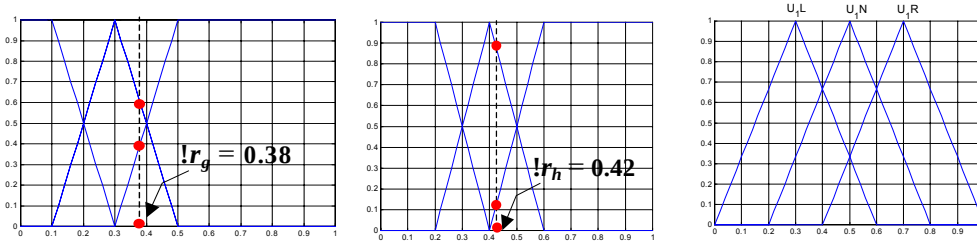


Figure 117. Fonctions d'appartenance de la variable g , h et u_1 aux classes floues

Pour modéliser chaque règle en appliquant l'approche ZMA, on effectue le produit cartésien de la variable logique floue (D_{g_N} ET D_{h_N}) et U_{1_N} . Cela implique l'intersection de l'extension cylindrique de (D_{g_N} ET D_{h_N}) - et de l'extension cylindrique de U_{1_N} . Pour chaque règle, nous obtenons la matrice des nombres flous correspondants. La base de règles R correspond à l'agrégation des règles $R1$, $R2$, $R3$.

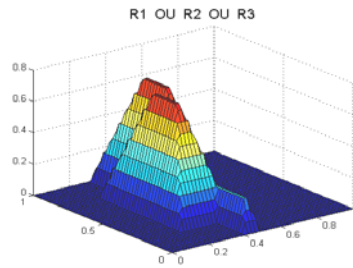


Figure 118. Représentation de la base de règles floues R

L'agrégation des règles est suivie par le mécanisme d'inférence avec les variables d'entrée $!r_g = s_1$ et $!r_h = s_2$. Cela implique aussi l'extension cylindrique des variables s_1 et s_2 puis l'intersection avec R .

Le résultat final est représenté par la fonction d'appartenance de la variable u_1 qui sera l'agrégation des résultats partiels obtenus par inférence. Par défuzzyfication, nous obtenons enfin la commande du système pour la maintenance.

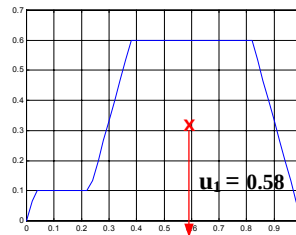


Figure 119. Défuzzyfication de la variable floue $U1$

Un signal de maintenance sera envoyé vers le modèle du système dans la partie commande.

6.8 CONCLUSION ET PERSPECTIVES

Les travaux présentés dans ce chapitre ont été orientés vers la surveillance dynamique et l'aide à la maintenance (reprise) des systèmes de production par intégration des *connaissances imprécises*. Nous avons opté pour la modélisation de la *surveillance* et de la *maintenance floues* par les réseaux de Petri, outil reconnu comme étant bien adapté pour la modélisation des systèmes à événements discrets. La nouveauté de notre approche consiste dans l'intégration de la logique floue dans la modélisation des raisonnements logiques par les réseaux de Petri. La prise en compte du temps dans les réseaux est calquée sur le modèle du *réseau de Petri temporel* qui nous permet l'intégration facile des mécanismes conventionnels pour la modélisation des fonctions de la surveillance ou la maintenance des systèmes.

La logique floue permet un traitement original de l'incertitude, en formalisant le traitement de la connaissance partielle et de l'inconsistance des systèmes d'information. Dans ce sens, ce travail de recherche a évalué les méthodes et théories applicables dans la surveillance des systèmes qui utilisent comme outil descriptif les raisonnements logiques flous. Nous avons proposé l'étude des concepts de détection/ pronostic/ maintenance dans l'apparition/ propagation/ correction des défauts, en exploitant une base d'informations imprécises. Le point fort de la logique floue est de permettre la représentation par des règles logiques des connaissances imprécises, incertaines ainsi que des connaissances expérimentales et pratiques qui sont difficiles à modéliser par les techniques de la logique classique.

En synthèse, nous avons proposé une approche globale homogène de modélisation en faisant appel à deux outils de modélisations de la même famille : les réseaux de Petri flous pour la surveillance (RdPFS) et les réseaux de Petri flous pour la maintenance (RdPFM). Ces outils sont basés essentiellement sur la modélisation floue de la variable "instant d'apparition" de l'événement discret. L'aspect temporel de l'approche formalisée par les bases de règles logiques, permet l'interfaçage du modèle avec des réseaux de Petri temporels de type différent.

Le modèle réseau de Petri flou pour la surveillance (RdPFS) est construit sur l'ensemble des règles logiques correspondant à l'arbre de défaillance du système surveillé. Nous avons montré que cet outil permet une analyse dynamique des défauts par une approche temporelle floue. Cette analyse présente un grand intérêt pour la surveillance dynamique et – de part sa nature prédictive – dans le pronostic industriel.

L'action de surveillance est suivie, dans la conception globale de la fonction de supervision, de l'action de reprise. Nous nous sommes intéressé à l'étude de cette problématique, en modélisant la fonction de décision floue pour la maintenance à l'aide des réseaux de Petri flous pour la maintenance (RdPFM).

Les modèles du système (RdPTO), du processus de surveillance (RdPFS) et du processus de maintenance (RdPFM), s'échangent des informations par l'intermédiaire de *signaux de synchronisation* flous. Ceci permet une maîtrise avancée des *dégradations* du système durant le fonctionnement, une analyse dynamique permanente de son état de dégradation et le déclenchement d'actions de maintenance préventive. Cette action précoce constitue une fonction de *pronostic* qui se trouve ainsi intégré dans un outil global et homogène de supervision.

Un prototype de plate-forme logicielle d'aide à la décision en *maintenance prévisionnelle* a été développé sur la base de la méthodologie de supervision proposée.

Les perspectives de cette étude consistent à aborder le problème sous un angle différent, de recherche de cause pour le diagnostic/pronostic, en utilisant des outils de la même famille - RdP flous. Des approches abductives existantes (Looney *et al.*, 2002, 2003) ont été adaptées et ont fait l'objet de développements supplémentaires dans notre axe **(Ghernaut et al., 2005)**, concernant la prise en compte de la qualité dans la maintenance. En effet, la plupart des actions de maintenance sont déclanchées en prenant en compte uniquement des informations liées aux machines et aux outils. Une information capitale est représentée par la qualité du produit fini. En collaboration avec Alstom Belfort et Cegelec, nous avons ainsi mis en évidence une problématique intéressante, permettant une approche conjointe maintenance – qualité pour le diagnostic et le pronostic. Les travaux concernant ces développements font l'objet des travaux de thèse en cours de A. Ghernaut, travaux qui ne seront pas développés ici.

Par ailleurs, une perspective intéressante est constituée par l'utilisation des propriétés d'apprentissage dynamique des outils tels que le RRFR et des capacités de prise en compte des connaissances imprécises et incertaines offertes par la logique floue pour en faire un outil neuro-flou adaptatif et évolutif d'aide au diagnostic industriel. Ces travaux font l'objet du chapitre suivant.

Chapitre 7 - Vers un Outil Hybride Neuro-Flou pour la Surveillance Dynamique

7.1 INTRODUCTION

Comme nous l'avons souligné précédemment, la recherche des causes des défaillances est un problème fondamental, négligé dans la plupart des approches dites de « diagnostic par reconnaissances de formes ». Ceci correspond en fait à une classification, correspondant à la détection du défaut. En effet, si chaque défaut était parfaitement catalogué avec sa cause possible claire et l'identification précise de l'organe défaillant, le processus de diagnostic pourrait en effet se réduire à une simple reconnaissance de formes.

Dans la réalité industrielle, les choses se passent souvent différemment. L'identification de l'organe défaillant et la recherche de la cause sont des processus beaucoup plus complexes, prenant en compte l'information disponible (pas toujours exhaustive) sur le système concerné, son évolution, les différentes révisions et évolutions qu'il a pu subir. Par ailleurs, l'historique de maintenance (pas toujours complet) est souvent insuffisant pour permettre la constitution d'un AMDEC ou d'un arbre de défaillance suffisamment pertinent et mis à jour pour constituer à lui tout seul un outil efficace. Mise à part la complexité du processus de recherche de causes, le diagnostic est donc souvent tributaire de la pertinence et la validité des informations disponibles.

Dans ce contexte, nous nous sommes proposés d'étudier un outil d'aide au diagnostic capable de prendre en compte l'information de maintenance au format industriel (AMDEC, ADD, GMAO) l'expérience humaine avec ses imprécisions et ses incertitudes, tout en enrichissant – au cours de son fonctionnement - sa base de connaissances par apprentissage dynamique des faits nouveaux.

Nous retrouvons dans ces exigences, des caractéristiques des systèmes neuro-flous. En effet, ces systèmes combinent les capacités d'apprentissage et de généralisation des réseaux de neurones, avec la formalisation de règles imprécises ou entachées d'incertitudes correspondant à une information humaine approximative ou à une information partielle ou partiellement erronée, comme c'est souvent le cas dans les applications industrielles.

7.2 GENERALITES SUR LES SYSTEMES NEURO-FLOUS

Actuellement les systèmes à inférence floue sont employés dans de nombreux domaines industriels. Utilisés principalement lorsque le modèle mathématique du système physique est difficile à élaborer, ils exploitent des règles floues tirées d'une expertise humaine pour modéliser le comportement dynamique du système. Ces règles sont du type :

« SI la vitesse est grande et la distance est petite, ALORS freiner très fortement »

Les principaux avantages des techniques floues sont l'approche naturelle de la modélisation et la bonne interprétabilité de la description, en employant des règles linguistiques. Cependant, comme il n'y a aucune méthode formelle pour déterminer ses paramètres (ensembles et règles floues), l'exécution d'un système flou peut prendre beaucoup de temps. Dans ce sens, il serait intéressant de disposer d'algorithmes permettant l'apprentissage automatique de ces paramètres.

L'une des méthodes qui permet de répondre à ces exigences est la théorie des réseaux de neurones qui emploie des échantillons (données d'observation) pour l'apprentissage. La combinaison des deux techniques nous donne les systèmes neuro-flou.

Définition du neuro-flou (Nauck et al., 1997)

Les systèmes neuro-flous sont des systèmes flous formés par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones. La technique d'apprentissage opère en fonction de l'information locale et produit uniquement des changements locaux dans le système flou d'origine.

Les règles floues codées dans le système neuro-flou représentent les échantillons imprécis et peuvent être vues en tant que prototypes imprécis des données d'apprentissage. Un système neuro-flou ne devrait par contre pas être vu comme un système expert (flou), et il n'a rien à voir avec la logique floue dans le sens strict du terme. On peut aussi noter que les systèmes neuro-flous peuvent être utilisés comme des approximateurs universels. (Lee et al., 2001).

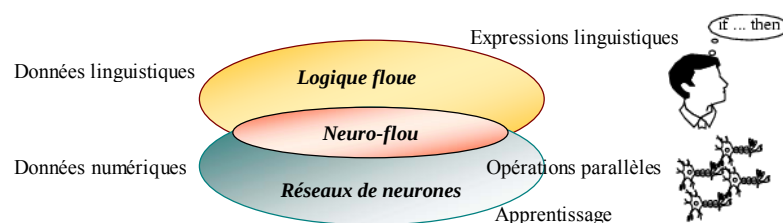


Figure 120. Le système neuro-flou

7.2.1 AVANTAGES ET INCONVENIENTS DE LA LOGIQUE FLOUE ET DES RESEAUX DE NEURONES

L'utilisation conjointe des réseaux de neurones et de la logique floue, permet de tirer les avantages des deux méthodes : les capacités d'apprentissage de la première et la lisibilité et la souplesse de la seconde.

Afin de résumer l'apport du neuro-flou, le Tableau 10 regroupe les avantages et les inconvénients de la logique floue et des réseaux de neurones.

Les systèmes neuro-flous sont créés afin de synthétiser les avantages et de surmonter les inconvénients des réseaux neuronaux et des systèmes flous. De cette manière, les algorithmes d'apprentissage peuvent être employés pour déterminer les paramètres des systèmes flous. Ceci revient à créer ou améliorer un système flou de manière automatique, au moyen des méthodes spécifiques aux réseaux neuronaux.

Un aspect important est que le système reste toujours interprétable en termes de règles floues, vu qu'il est basé sur un système flou.

Réseaux de neurones	Logique floue
avantages	
- Le modèle mathématique non requis - Aucune connaissance basée sur les règles - Plusieurs algorithmes d'apprentissage sont disponibles	- Le modèle mathématique non requis - La connaissance antérieure sur les règles peut être utilisée - Une interprétation et implémentation simple
inconvénients	
- Boite noire (manque de traçabilité) - L'adaptation aux environnements différents est difficile et le réapprentissage est souvent obligatoire (sauf pour le RBF) - la connaissance antérieure ne peut pas être employée (apprentissage à partir de zéro) (sauf pour le RBF) - Aucune garantie sur la convergence de l'apprentissage	- Les règles doivent être disponibles - Ne peut pas apprendre - Adaptation difficile au changement de l'environnement - Aucunes méthodes formelles pour l'ajustement

Tableau 10. Comparaison entre la logique floue et les réseaux de neurones

7.2.2 METHODES ET COMBINAISONS NEURO-FLOUES

Diverses associations de ces deux méthodes ont été développées depuis 1988 et sont le plus souvent orientées vers la commande de systèmes complexes et les problèmes de classification. Il existe ainsi trois méthodes neuro-floues :

Première méthode neuro-floue :

L'une des premières méthodes neuro-floues est le codage du système d'inférences floue sous la forme d'un réseau de neurones multicouches dans lequel les poids correspondent aux paramètres du système. L'architecture du réseau dépend du type de règle et des méthodes d'inférence, d'agrégation et de défuzzification choisies.

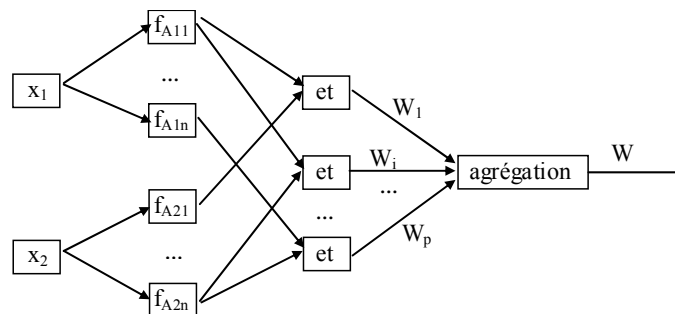


Figure 121. Exemple de réseau de neurones pour la commande floue

D'après la figure précédente, pour des règles de la forme :

$$"SI V_1 \text{ est } A_{1i} \text{ et } V_2 \text{ est } A_{2i} \text{ ALORS } W = w_i"$$

nous obtenons un réseau de neurones qui admet pour entrée les valeurs x_1 et x_2 prises par les variables V_1 et V_2 et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenance A_{ij} pour l'entrée x_i et à celui de la valeur prise par la conjonction des conditions de chaque règle utilisant un opérateur adéquat. Les fonctions d'appartenance sont considérées comme des paramètres ajustés par les poids entrant dans la première couche cachée. Les conclusions W_i des règles sont également des paramètres ajustables par l'intermédiaire des poids associés à la dernière couche.

Seconde méthode neuro-floue :

Cette seconde approche consiste à utiliser les réseaux de neurones pour remplacer chacune des composantes d'un système de commande floue. Ces réseaux sont destinés à l'apprentissage des fonctions d'appartenance, au calcul de l'inférence et à la réalisation de la phase d'agrégation et de défuzzification. Ils peuvent réaliser l'extraction des règles floues en analysant la corrélation qui existe entre les entrées et les sorties du réseau de neurones.

Ces approches permettent de résoudre deux problèmes importants de la logique floue : le déterminisme des fonctions d'appartenance et l'adaptation à l'environnement du système.

Troisième méthode neuro-floue :

Elle utilise des réseaux de neurones et des systèmes flous associés en série ou en parallèle. Plusieurs variantes d'utilisation sont ainsi possibles :

- Le réseau de neurones fonctionne en amont du système flou. Les variantes d'entrées du système flou sont déterminées à partir des sorties du réseau de neurones (dans le cas où elles ne sont pas mesurables directement) ou encore un réseau de neurones effectue une tâche de classification ou de reconnaissance de formes, suivie d'un système flou d'aide à la décision.

- Un réseau de neurones qui fonctionne en aval du système flou, dans le but d'ajuster les sorties d'un système de commande floue à de nouvelles connaissances obtenues, les variables de sorties étant les erreurs sur les variables de sortie du système flou.

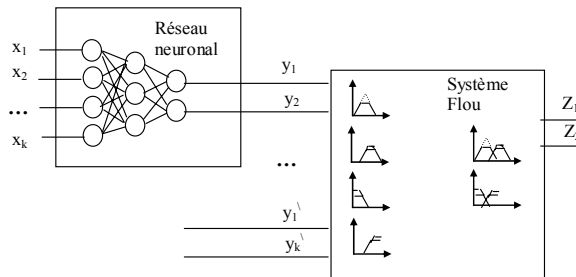


Figure 122. Exemple d'association en série d'un réseau de neurone et d'un système flou

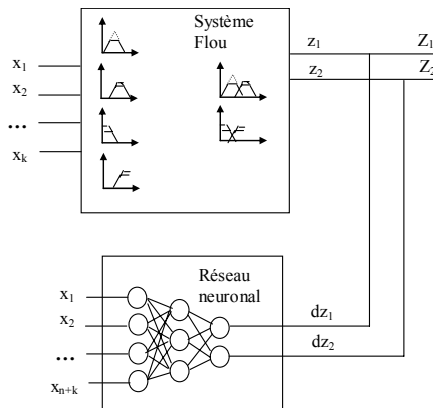


Figure 123. Exemple d'association en parallèle d'un réseau de neurone et d'un système flou

Il existe quatre grandes catégories de combinaisons des réseaux de neurones avec la logique floue :

- Réseau flou neuronal :

Des techniques floues sont utilisées pour augmenter les possibilités d'apprentissage ou l'exécution d'un réseau neuronal. Ce genre d'approche ne doit pas être confondu avec les approches neuro-floues proprement dites

- Système neuronal/flou simultanément :

Le réseau neuronal et le système flou fonctionnent ensemble sur la même tâche, mais sans s'influencer, c.-à-d. ni l'un ni l'autre n'est employé pour déterminer les paramètres de l'autre. Habituellement le réseau neuronal traite les entrées, ou post-traite les sorties du système flou.

- Modèles neuro-flous coopératifs :

Le réseau neuronal est employé pour déterminer les paramètres (les règles et les ensembles flous) d'un système flou. Après la phase d'apprentissage, le système flou fonctionne sans le réseau neuronal. C'est une forme simple des systèmes neuro-flous. Elle est largement répandue dans des outils de développement flous commerciaux, quoique les problèmes sémantiques puissent surgir.

- Modèles neuro-flous hybrides :

Les approches neuro-floues modernes sont de cette forme. Un réseau neuronal et un système flou sont combinés dans une architecture homogène. Le système peut être interprété comme un réseau neuronal spécial avec des paramètres flous ou comme un système flou mis en application sous une forme distribuée parallèle.

Notre étude s'intéressera particulièrement à ce dernier type d'architecture (**Racocceanu et al., 2003**), qui tire un meilleur parti de l'utilisation conjointe des deux principes.

7.2.3 SYSTEMES NEURO-FLOUS HYBRIDES

a) ARCHITECTURES NEURO-FLOUES HYBRIDES

Une architecture à 3 modèles flous est proposée dans (Lee et al., 2001) :

La règle: j ($j = 1, 2, \dots, J$): **SI** x_1 est A_1^j et ... et x_n est A_n^j , **ALORS** y_1 est f_1^j et ... et y_m est f_m^j .

$$\text{où } f_k^j = \begin{cases} B_k^j & \text{(type I)} \\ \theta_k^j & \text{(type II)} \\ b_{0k}^j + b_{1k}^j x_1 + \dots + b_{nk}^j x_n & \text{(type III)} \end{cases} \quad (\text{IV.1})$$

avec x_i ($i = 1, 2, \dots, n$), y_k ($k = 1, 2, \dots, m$) respectivement les variables d'entrées et de sortie, A_i^j les ensembles flous d'entrée; B_k^j , θ_k^j et $b_{0k}^j + b_{1k}^j x_1 + \dots + b_{nk}^j x_n$ représentant respectivement, des ensembles flous de sortie, des composants singleton et des combinaisons linéaires des variables d'entrées.

La Figure 124 montre les différentes architectures des systèmes neuro-flous. FALCON (a) et GARIC (b) interprètent la règle floue de type I avec une structure à 5 couches, NEFCLASS (c) interprète la règle floue de type II avec une structure à 3 couches et ANFIS (d) interprète la règle floue de type III avec une structure à 6 couches.

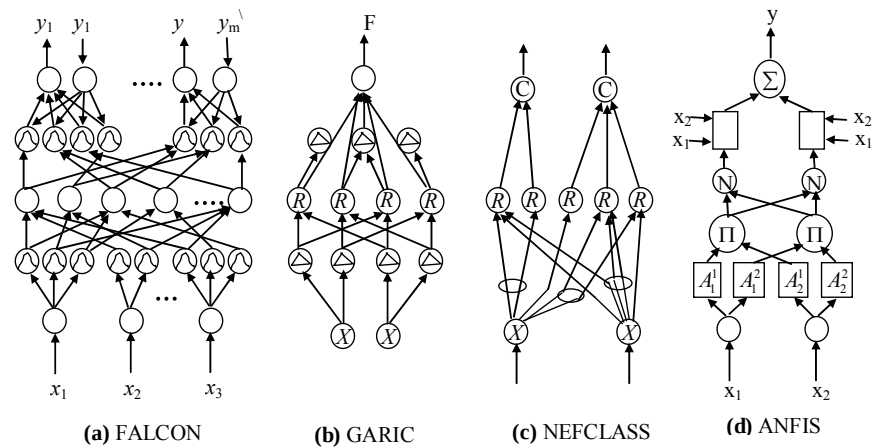


Figure 124. Différentes architectures des systèmes neuro-flous

a.1 ANFIS (Adaptive-Neuro-based Fuzzy Inference System)

ANFIS représente un système à inférence floue mis en application dans le cadre des réseaux adaptatifs. Il utilise la procédure d'apprentissage Hybride (Shing *et al.*, 1993). Cette architecture (Figure 124 d) affine les règles floues obtenues par des experts humains pour décrire le comportement d'entrée-sortie d'un système complexe. Il est implémenté dans la boîte à outils « Neuro-Fuzzy » de MATLAB. Ce modèle donne de très bons résultats en poursuite de trajectoire, approximation non linéaire, commande dynamique et traitement du signal.

a.2 FALCON et GARIC

Il s'agit de modèle à 5 couches, utilisant la fuzzyfication en entrée et la défuzzyfication en sortie. Ceci correspond à l'interprétation juste de la technique de Mamdani. La précision accrue des résultats provoque une lenteur dans l'exécution du système. Ce modèle est rarement utilisé en pratique mais il reste le meilleur pour la commande (Figure 124 a,b)

a.3 NEFCLASS

Modèle utilisé généralement en classification, il est constitué de 3 couches : une couche d'entrée avec les fonctions d'appartenance, une couche cachée représentée par des règles et une couche de sortie définissant les classes (Nauck *et al.*, 1996) (Figure 124 c)

Ce modèle est facile à mettre en application, il évite l'étape de défuzzyfication, tout en étant précis dans le résultat final, avec une rapidité bien supérieure aux autres architectures.

a.4 NEFCON

Modèle formé de 3 couches. Une couche cachée formée par des règles, une couche d'entrée incluant les nœuds d'entrée avec les sous-ensembles flous d'antécédences et une couche de sortie avec un nœud de sortie et les sous-ensembles des conséquences. L'élaboration des règles est similaire à l'architecture NEFCLASS, avec une différence en sortie (Kruse *et al.*, 1998). Cette architecture est généralement utilisée en approximation de fonctions et en control flou (Figure 125).

Le processus d'apprentissage du NEFCON peut être divisé en deux phases. La première phase consiste à trouver les règles de base initiales. Si les connaissances antérieures ne sont pas disponibles les règles de base seront apprises avec difficulté. Et si cette règle est

définie par un expert l'algorithme les complètes. Dans la seconde phase, les règles de base sont optimisées par modification des sous-ensembles flous des règles. Les deux phases utilisent l'erreur floue, cette erreur peut être trouvée avec la différence entre la sortie désirée et celle obtenue.

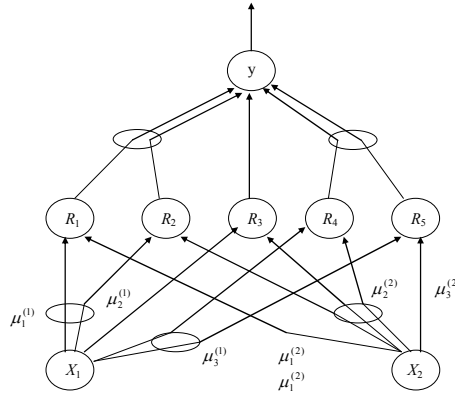


Figure 125. Modèle NEFCON avec deux entrées, 5 règles et une sortie

a.5 NEFPROX (Neuro Fuzzy function appROXimator)

Modèle obtenu par l'association des deux architectures : NEFCLASS et NEFCON, il est utilisé dans différentes applications comme la classification et l'approximation de fonctions (Nauck *et al.*, 1997). Le NEFCLASS utilise un algorithme supervisé pour définir les règles floues, le NEFCON utilise un algorithme d'apprentissage non supervisé avec le calcul de l'erreur de sortie. Les deux modèles emploient la rétropropagation afin de définir les sous-ensembles flous.

Comparé au modèle ANFIS, NEFPROX est beaucoup plus rapide, mais ANFIS donne de meilleurs résultats en approximation.

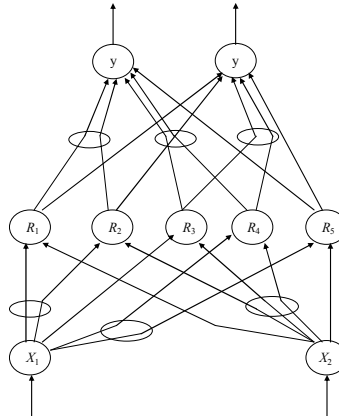


Figure 126. Modèle NEFPROX avec deux entrées, 5 règles et deux sorties

Le NEFPROX est le premier système interprétable et lisible, dédié à l'approximation de fonction. Néanmoins, ses résultats en classification restent moins bons que ceux donnés par le NEFCLASS.

a.6 Architecture SANFIS

Une autre structure connexionniste basée sur la fonction floue de base (FBF), a été développée récemment (Lee *et al.*, 2000) (Figure 127). Pour ces 3 modèles qu'on peut

comparer à ceux de la Figure 124, les entrées sont les mêmes, alors que les sorties sont différentes. Ils sont utilisés selon le type d'applications :

- Le modèle de type I (modèle de Mamdani) est linguistiquement compréhensible, car il utilise les ensembles flous dans les antécédentes et les conséquences des règles floues. Avec sa meilleure capacité de généralisation, ce modèle est plus approprié pour la commande et les systèmes experts.
- Le type II est une version simplifiée du premier. Il réduit le nombre de paramètres et évite le processus de défuzzification, ce type est généralement utilisé en classification.
- Le type III réduit encore le nombre des règles floues pour modéliser les systèmes évolués ; il est utilisé dans les fonctions d'approximation non linéaires, la commande dynamique et le traitement du signal.

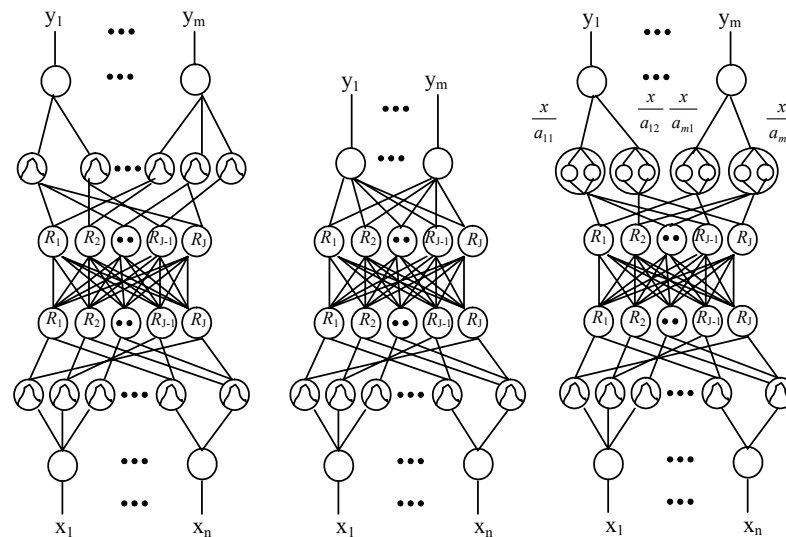


Figure 127. Les trois différents types de SANFIS

Par la suite, nous allons passer en revue les différents algorithmes d'optimisation et de classification associés aux systèmes neuro-flous hybrides. Ces algorithmes portent le nom d'algorithmes de groupage.

b) ALGORITHMES DE GROUPAGE ASSOCIES AUX SYSTEMES NEURO-FLOUS HYBRIDES

Un système flou peut être ajusté automatiquement. Si, en plus de l'expertise humaine, on dispose de données expérimentales décrivant le même phénomène, il devient possible de mettre en œuvre un algorithme d'apprentissage sur ces données. L'algorithme ajuste progressivement les coefficients du modèle (en général, les paramètres définissant les trapèzes des ensembles flous) de façon à minimiser l'erreur de prédiction commise par le système de règles floues. Cet apprentissage, tout à fait semblable à celui que subit un réseau de neurones, a fait qualifier la méthode de "neuro-floue". Cette analogie est renforcée par le fait que l'on peut donner à partir d'un ensemble de règles floues, une représentation graphique ressemblant à celle d'un réseau de neurones.

b.1 Groupage flou

Les algorithmes de groupage flou forment une famille bien identifiée des techniques d'induction de règles. L'objectif initial de ces algorithmes est d'organiser des données en groupes homogènes, pour, notamment dans le cadre de la classification (Ahmedou et al.,

97), rendre compte de l'appartenance d'un individu à plusieurs classes. L'idée d'associer une règle à chaque groupe, et ainsi d'utiliser ces techniques pour l'induction de règles, est apparue dans un deuxième temps. Pour ce faire, le partitionnement des entrées correspondant à la prémisse de la règle est déduit des caractéristiques du groupe : prototype et zone d'influence.

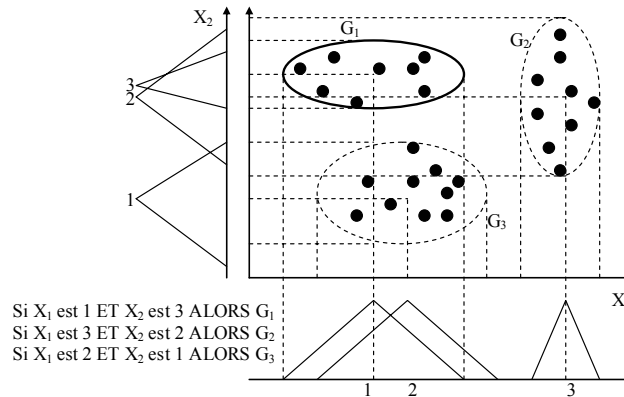


Figure 128. Exemple de groupage

b.2 Techniques de groupage

Selon le type d'apprentissage qui nous permet de déterminer et d'optimiser le nombre de classes, on peut citer deux méthodes de groupage :

✚ Méthodes statiques d'adaptation

Dans cette catégorie, un algorithme de groupement commence par un nombre fixe de classes déterminées par l'utilisateur. Pendant le procédé de groupage, la configuration des classes est mise à jour par un processus d'optimisation.

Dans cette catégorie de méthodes, on retrouve le *Hard Clustering-C-Means* (HCM) et le *soft clustering-Fuzzy C-Means* (FCM) ou ISODATA.

Les algorithmes HCM et FCM divisent une collection de vecteurs en classes et trouvent les centres des classes tel qu'une fonction objective de dissimilitude (ou de distance) est réduite au minimum (Pal *et al.*, 1995) et (Lee *et al.*, 2001). Ces algorithmes ne garantissent pas une convergence à une solution optimale, leur efficacité est fortement liée aux choix initiaux des centres de classes. De même, en raison du nombre important de classes, la configuration finale peut être difficile à interpréter.

✚ Méthodes dynamiques d'adaptation

Deviner ou déterminer le nombre de classes par l'être humain peut s'avérer être impossible. Par conséquent les algorithmes d'apprentissage capables de déterminer automatiquement le nombre de classes ont suscité un intérêt croissant. On peut citer deux catégories de méthodes (Lee *et al.*, 2001), (Ahmedou *et al.*, 1997) : *Growing Partitioning Algorithms* (GPA) ou algorithme de partitionnement croissant, et *Shrinking Partitioning Algorithms* (SPA) ou algorithme de partitionnement décroissant.

- *Growing Partitioning Algorithms* : Algorithme qui commence par des classes vides et adapte les nouvelles classes jusqu'à ce que le critère spécifier soit satisfait. La théorie adaptative de résonance (ART) et l'algorithme de groupement aligné (ACA) sont deux exemples représentatifs du GPA.

- *Shrinking Partitioning Algorithms* : Mountain clustering method (MCM), et les deux variantes : Subtractive clustering method (SCM) et Mapping-constrained agglomerative (MCA), sont des exemples représentatif de l'algorithme SPA. Dans le but d'accroître le nombre de classes produites, le SPA commence par un nombre raisonnable de classes et élimine celles qui ont une densité faible (poids négligeable), durant le processus d'évaluation seulement.

7.2.4 METHODOLOGIE DE MODELISATION NEURO-FLOUE

La modélisation neuro-floue comporte trois phases (Chen *et al.*, 2001) :

- Création du modèle flou initial, qui consiste en la collection des règles floues
- Sélection des variables d'entrées significatives, en se basant sur le modèle flou initial
- Le nombre optimal des règles floues est déterminé par l'algorithme FCM.

La figure suivante nous montre le schéma général de la modélisation neuro-floue :

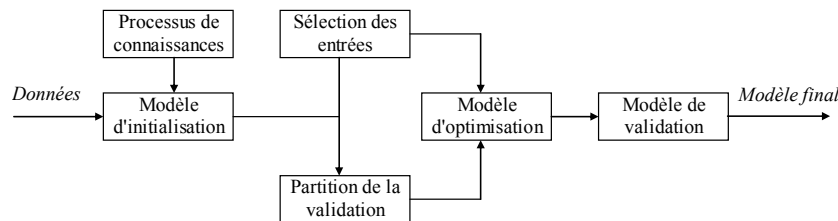


Figure 129. Schéma général de la modélisation neuro-floue

7.2.5 APPROCHE NEURO-FLOUE EN CLASSIFICATION

a) CLASSIFICATION A BASE DE REGLES FLOUES

L'idée générale que sous-tend cette approche, est l'hypothèse de l'existence d'une relation entre les valeurs d'un vecteur d'attributs et la classe à laquelle appartient l'échantillon. Par exemple, on peut dire : « Les objets de la classe C_1 sont caractérisés par des vitesses élevées et des altitudes moyennes ou parfois basses ». cela peut être traduit en relation floue de la manière suivante :

« **SI** un objet se déplace rapidement et à altitude moyenne, **ALORS** il est très vraisemblable qu'il appartient à la classe C_1 , mais il est possible aussi qu'il appartienne à la classe C_2 ».

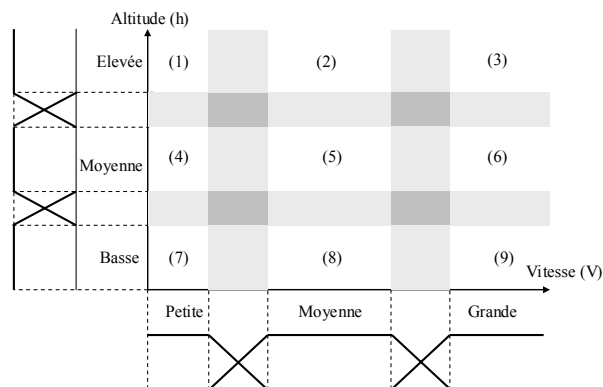


Figure 130. Partition floue des attributs

On commence par faire une partition floue des attributs avec des termes linguistiques. De cette manière, l'espace représentant l'objet sera divisé en sous-ensembles flous. Pour effectuer la classification d'un individu inconnu, on compose la relation floue avec le vecteur d'appartenance de l'individu aux sous domaines flous ainsi définis. La Figure 130 présente une partition floue des attributs. Dans notre exemple, les attributs altitudes (h) et vitesse (v) se divisent chacun en trois sous-ensembles flous (Ayat *et al.*, 2000).

L'espace D représente l'objet est défini par 9 sous domaines flous (numérotés de 1 à 9) que l'on va relier à l'espace des classes par des relations (ou règles) floues. Ces relations seront représentées sous forme d'une matrice, dont chaque élément ij exprime la force de la relation entre la classe i et le sous domaine j .

Sous-domaine Classe	1	2	3	4	5	6	7	8	9
C_1	0	0	0	0	0	1	0	0	0,7
C_2	-	-	-	-	-	0,3	-	-	-

Tableau 11. Degrés flous de relation entre les classes et les sous domaines de la partition

b) APPRENTISSAGE PAR TECHNIQUES NEURONALES

Sauf cas très simple, les relations floues ne peuvent être déterminées à la main. Elles doivent provenir de l'expertise ou bien de l'apprentissage. Le premier cas correspond à un système expert nécessitant l'opinion d'un spécialiste dans le domaine. Cependant, sauf unanimité des experts, il n'est pas toujours facile de déterminer exactement toutes les relations susceptibles d'intervenir dans le processus. L'apprentissage, de ce fait, peut apporter une solution. Il s'agit dans ce cas d'un système capable de générer des relations à partir des exemples (ou échantillons d'apprentissage).

7.3 CONCEPTION D'UN OUTIL NEURO-FLOU POUR L'AIDE AU DIAGNOSTIC

7.3.1 UTILISATION DES SYSTEMES NEURO-FLOUS POUR LE DIAGNOSTIC

Dans la bibliographie liée à l'utilisation des systèmes neuro-flous pour le diagnostic, nous avons recensé quelques contributions importantes :

- Dans (How *et al.*, 1999, 2000), les auteurs présentent une méthode utilisant la logique floue pour le diagnostic des transformateurs. Ils envisagent une utilisation ultérieure des réseaux de neurones pour la génération automatique de règle à partir des données.
- Dans (Tromp, 2000), l'auteur associe un modèle dynamique des liens causaux entre les défaillances, au modèle numérique de traitement du signal. Ce choix permet la prise en compte d'informations liées à la fiabilité, en plus des informations en provenance des capteurs. Les outils utilisés sont les réseaux de Petri et les chaînes de Markov Cachées (HMM – Hidden Markov Model)
- Dans (Pencolé, 2002), l'auteur présente une méthode de diagnostic réparti pour les systèmes à événements discrets comme les réseaux de télécommunication. Une technique de fusion des diagnostics locaux est utilisée pour obtenir un diagnostic global.

Dans le travail de thèse de N. Palluat, nous proposons l'utilisation de techniques neuro-floues pour une aide dynamique au diagnostic. Il est à noter que cette thèse a démarré dans le cadre technique de collaboration du work packaging « decision systems » du projet européen Proteus (**PROTEUS, 2004**) et s'est continuée par une phase de prototypage soutenue par l'Anvar (**Palluat et al., 2004b**).

Cette méthode comprend plusieurs étapes (**Palluat et al., 2004a**) :

- *L'acquisition des informations pertinentes du système :*

En utilisant les informations extraites du système (analyse des modes de défaillances, de leurs effets et de leur criticité – AMDEC, l'arbre de défaillance et de dégradations - ADD, l'analyse fonctionnelle - AF, ...), et avec l'aide des opérateurs et des experts de l'installation industrielle concernée, il est nécessaire d'extraire les zones critiques à surveiller, ainsi que les informations utiles de ces zones : informations statiques telles que l'ADD, les AF et les informations dynamiques (gestion de la maintenance assistée par ordinateur – GMAO, historique, capteur caractéristiques, etc.).

- *Application du système de détection basé sur les réseaux de neurones dynamiques :*

En entrée du système de détection, nous trouvons les informations données par les capteurs ; elles peuvent être binaires ou réelles. En sortie, les experts identifient le mode opératoire (le symptôme) de l'élément supervisé. L'utilisation des réseaux de neurones artificiels est justifié par leur capacité d'apprentissage, leur robustesse, leur parallélisme dans le traitement, leur capacité à faire face à des problèmes inhérents à la non linéarité des systèmes et leur vitesse de calcul lorsqu'ils sont implémentés dans un circuit intégré.

- *Application du système de diagnostic basé sur un réseau neuro-flou :*

En entrée du système de diagnostic, nous trouvons le degré d'appartenance de chaque mode opératoire donné par le système de détection. Nous trouvons aussi bien des entrées qualitatives ou quantitatives, comme des informations données par les opérateurs pour améliorer le diagnostic. En sortie, nous obtenons une liste des causes possibles classées par degré de crédibilité, avec le degré de gravité associé. Ces informations pertinentes peuvent être d'un grand soutien pour la planification des tâches de maintenance.

Le principe du système d'aide au diagnostic proposé est présenté dans la figure suivante

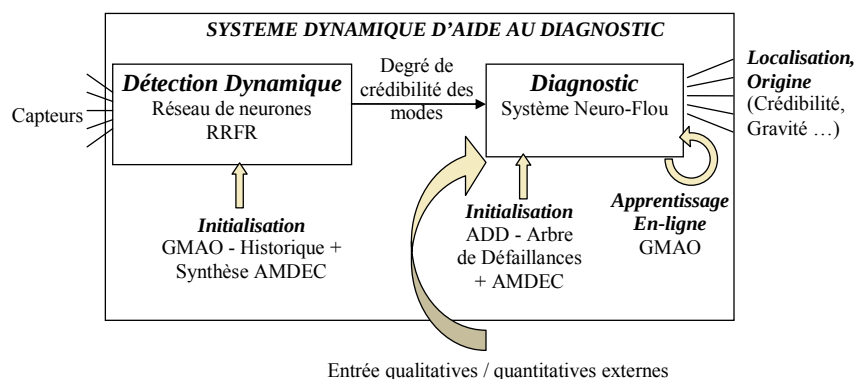


Figure 131. Schéma du système d'aide au diagnostic.

En fonctionnement, le système de détection scrute en permanence le système (ou le sous-système critique) surveillé. Lorsqu'une panne ou une défaillance se produit ou lorsque l'une des deux est susceptible de se produire, une alarme est déclenchée et un diagnostic est lancé, le système de détection n'arrêtant pas sa tâche. En utilisant les informations

provenant du système de détection, le système d'aide au diagnostic propose à l'opérateur les causes possibles de ce problème, ainsi que les interprétations floues de ces causes.

7.3.2 SPECIFICATIONS DE L'OUTIL EN UTILISANT LA DEMARCHE UML

Notons que l'essentiel de cette partie a fait l'objet d'une publication dans la conférence internationale (**Palluat *et al.*, 2005b**).

Notre approche de conception suit l'approche UML (Unified Modelling Language) (Larman, 2002), (Rumbaugh, *et al.*, 1999). Plusieurs raisons nous ont conduit à ce choix :

- La première est sa normalisation par l'OMG⁵⁵ (OMG, 2003). L'historique a montré que la profusion des notations est, à moyen terme, préjudiciable aux entreprises comme à leurs fournisseurs. Toute norme doit donc être considérée avec le plus grand sérieux, en particulier les normes du domaine public, comme c'est le cas pour l'UML.
- La deuxième raison est l'intérêt montré par les informaticiens pour ce langage de modélisation. Ceci a permis de trouver un terrain d'entente lors de travaux développés dans le cadre du projet européen PROTEUS. Il est ainsi intéressant de disposer d'un ensemble de modèles communs permettant une validation des partenaires, ainsi que le passage ultérieur à la phase de prototypage et de développement.
- La troisième raison est la possibilité d'utiliser le même atelier de génie logiciel, depuis l'expression des besoins jusqu'à la génération de tout ou partie de l'application.
- La dernière raison, mais non la moindre, est d'utiliser les principes et concepts objet pour enrichir la démarche de conception de systèmes d'aide à la décision. On en attend des améliorations dans le sens tout à la fois de la richesse, d'une modularité, d'une cohérence et d'une rigueur accrues.

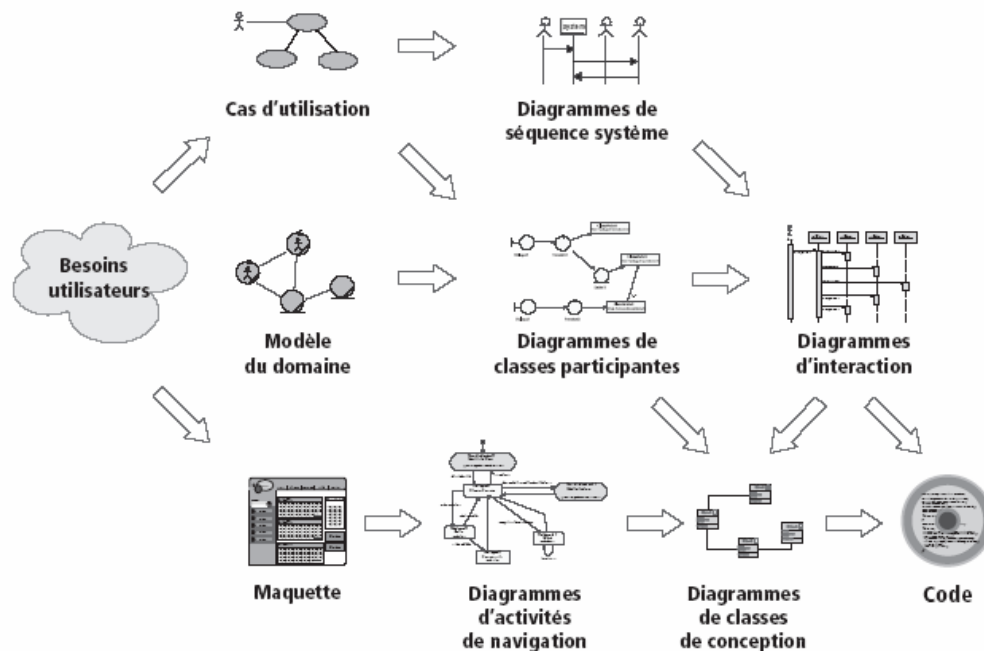


Figure 132. Schéma global du processus de modélisation UML

⁵⁵ Site Internet de l'OMG : <http://www.omg.org/>, site Internet de l'OMG lié à UML : <http://www.uml.org/>

UML s'articule autour de plusieurs types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Le schéma suivant montre comment, en partant des besoins utilisateurs formalisés par des cas d'utilisation et une maquette, et avec l'apport du modèle du domaine, on peut aboutir à des diagrammes de conception qui permettent de dériver du code assez directement (Roques, 2003).

a) DEFINITION DES BESOINS DES UTILISATEURS

Lors de l'étude des besoins des utilisateurs, nous avons spécifié les exigences fondamentales d'un outil d'aide au diagnostic moderne :

- utilisation simple
- capacité d'interfaçage avec les outils industriels d'acquisition de données (SCADA – Supervisory Control And Data Acquisition – Télésurveillance et Acquisition de Données, Ethernet, ...)
- possibilité d'utilisation en absence d'une base de données et de connaissances complète, avec la possibilité ultérieure de prendre en compte les nouvelles expériences (si possible en ligne)
- capacité d'identifier les fausses alarmes
- interfaçage avec les outils industriels de gestion de la maintenance (GMAO) et compréhension des résultats par un professionnel de la maintenance (paramétrage similaire aux méthodes industrielles de type AMDEC, ADD)
- possibilité d'interfaçage avec des IHM - clients légers (PDA, ordinateurs portables)
- intégration de l'outil dans un ordinateur industriel ou un système embarqué

b) SPECIFICATION DES CAS D'UTILISATION DE L'OUTIL

La première étape nécessaire à l'établissement des cas d'utilisation de l'outil est constituée par la définition des acteurs. Dans notre cas, les acteurs humains sont constitués par le responsable de la maintenance, l'expert de l'outil et l'opérateur de maintenance. Nous prenons aussi en compte les acteurs non humains, à savoir le SCADA⁵⁶, l'ADD, la GMAO, et l'AMDEC.

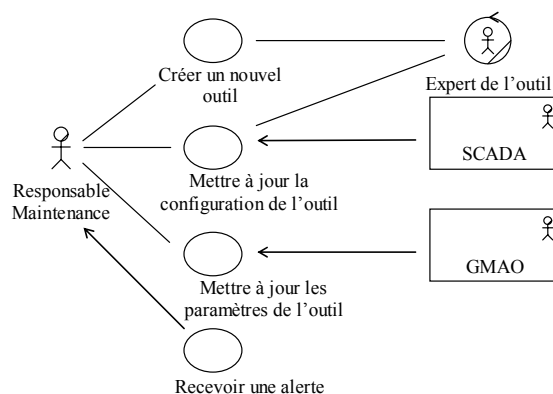


Figure 133. Cas d'utilisation concernant le Responsable de Maintenance

En ce qui concerne le responsable maintenance, ses cas d'utilisation principaux ont été mis en évidence par l'expression des besoins, à savoir :

- Créer un nouvel outil, qui fait intervenir l'acteur Expert de l'outil ;

⁵⁶ SCADA = Supervisory Control and Data Acquisition - système de supervision et d'acquisition de données des processus industriels

- Mettre à jour la configuration de l'outil, qui fait intervenir l'acteur Expert de l'outil et le système SCADA ;
- Mettre à jour les paramètres de l'outil, qui fait intervenir le système GMAO ;
- Recevoir une alerte.

En ce qui concerne l'opérateur de maintenance, il ne possède qu'un seul cas d'utilisation, à savoir : Demander une aide au diagnostic :

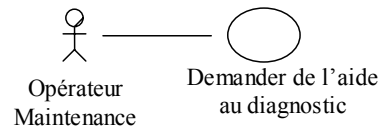


Figure 134. Cas d'utilisation de l'Opérateur de Maintenance

Enfin, pour l'acteur Expert de l'Outil, nous pouvons trouver les cas d'utilisation suivants :

- créer un nouvel outil, déjà présent dans les cas d'utilisation du responsable maintenance ;
- configurer l'outil, qui fait intervenir deux systèmes SCADA et ADD ;
- initialiser l'outil, qui fait intervenir trois systèmes SCADA, AMDEC et GMAO ;
- mettre à jour la configuration de l'outil, déjà présent dans les cas d'utilisation du responsable maintenance.

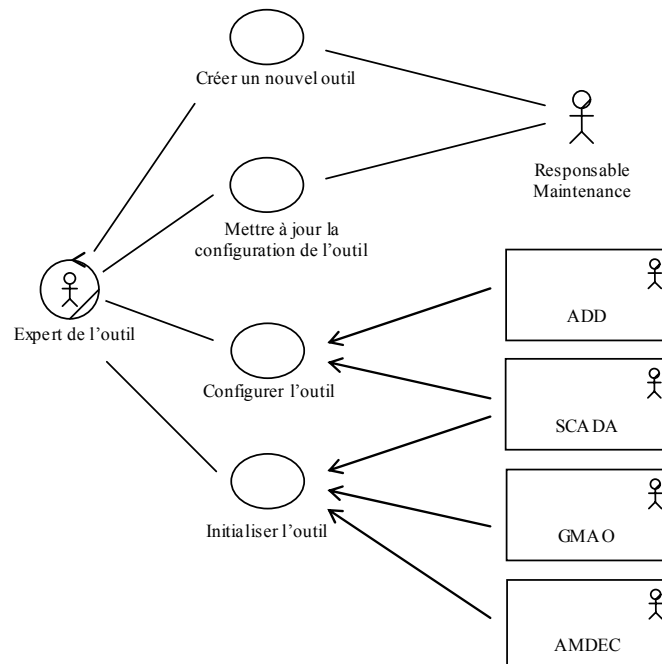


Figure 135. Cas d'utilisation de l'acteur Expert de l'Outil

c) STRUCTURATION EN PAQUETAGES

Pour optimiser l'organisation de notre système, nous avons organisé les cas d'utilisation en les regroupant en ensembles fonctionnels cohérents. Pour ce faire nous utilisons le concept général d'UML qui s'appelle le *package* ou *paquetage*. Les acteurs ont été ainsi

regroupés dans un paquetage séparé sur lequel s'appuient deux paquetages de cas d'utilisation :

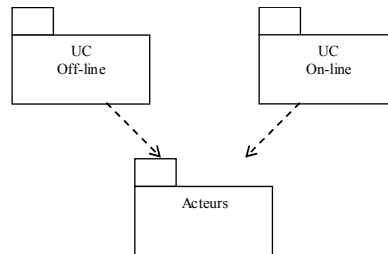


Figure 136. Organisation des cas d'utilisation et des acteurs en paquetages.⁵⁷

Parmi les cas d'utilisation de l'expert de l'outil, trois cas sont reliés par des relations d'inclusion : la création d'un nouveau système nécessite la configuration et l'initialisation de l'outil, qui nécessite elle-même la configuration. Nous retrouvons ainsi les paquetages UC Off-line et UC On-line détaillés :

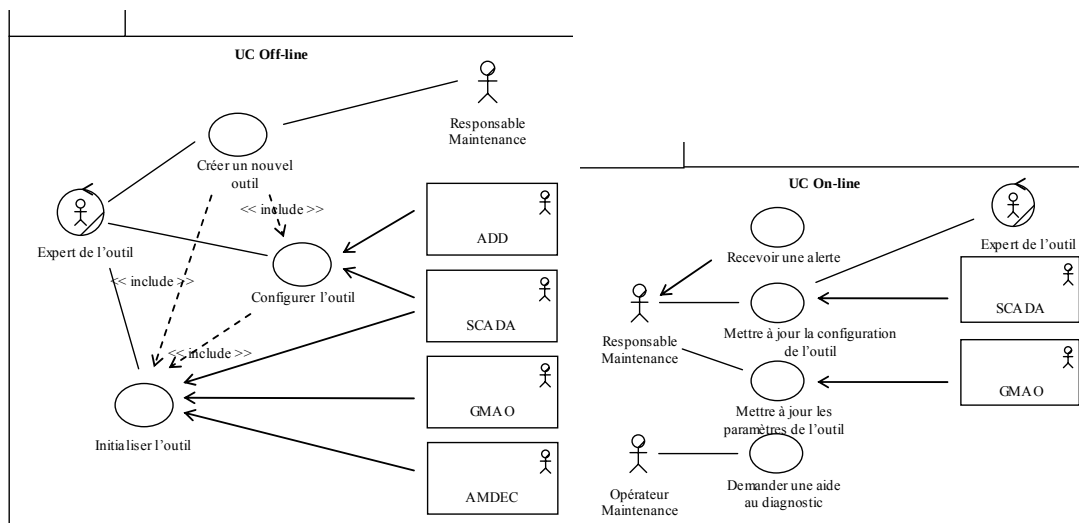


Figure 137. UC Off-line et On-line

d) MODELE DU DOMAINE

Après avoir effectué une modélisation par les cas d'utilisation qui correspond à une description fonctionnelle, il est nécessaire de passer à une modélisation structurale, statique du système à réaliser, sous forme d'ensemble de classes logicielles, éventuellement regroupées en paquetages.

Nous avons ainsi identifié les concepts métier participant à chaque cas d'utilisation, en prenant en compte les associations entre ces concepts et leurs attributs. En considérant les concepts majeurs identifiés, nous aboutissons à la représentation suivante :

⁵⁷ L'acronyme « UC » pour use case est souvent utilisé pour raccourcir les noms de paquetage.

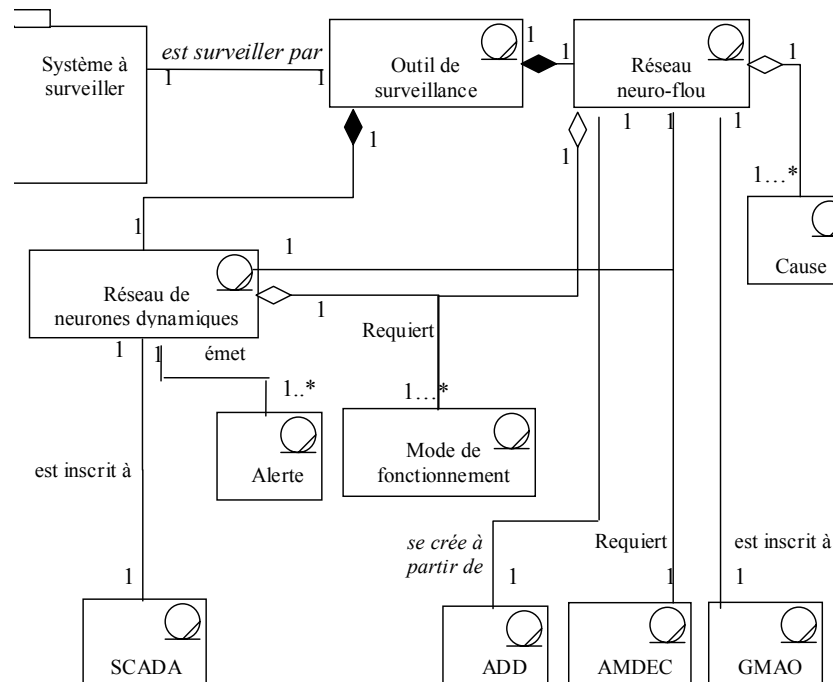


Figure 138. Cartographie des concepts majeurs du domaine

Pour la séparation en paquets, nous avons fait le choix de découper le modèle en trois paquets, le premier concerne le système à surveiller, le deuxième est l'outil de surveillance appelé outil DNFS – *Dynamic Neuro Fuzzy System* – et enfin le troisième est le paquetage regroupant les outils de la maintenance :

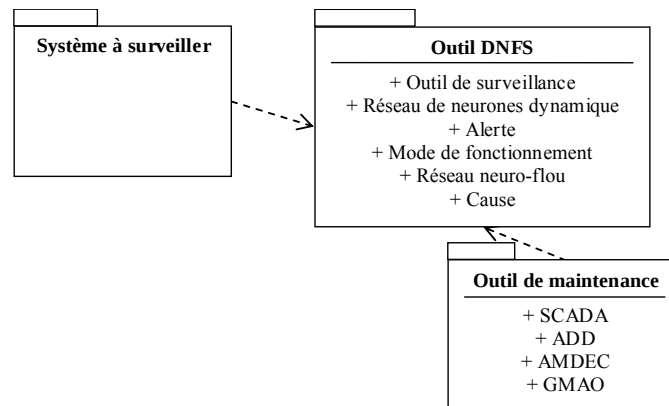


Figure 139. Représentation synthétique des paquets et de leur contenu

e) SPECIFICATION DETAILLEE DES EXIGENCES

A partir des cas d'utilisations établis précédemment, nous avons procédé à la description détaillée de ces cas, afin d'obtenir une expression précise des besoins précise, avant d'aborder l'analyse et la conception d'objets.

Nous avons ainsi spécifié de manière détaillée chacun des cas d'utilisation :

- création d'un nouvel outil
- configuration de l'outil
- initialisation de l'outil

- réception d'une alerte
- demande d'aide au diagnostic
- mise à jour de la configuration de l'outil
- mise à jour des paramètres de l'outil

A partir de cette description, nous avons pu définir des diagrammes de séquence système, les diagrammes des classes participantes, les diagrammes d'activités de navigation ainsi que les diagrammes d'interaction. Ceci nous a permis d'obtenir les requêtes spécifiques nécessaires, les interfaces nécessaires au développement de l'outil, ainsi que les procédures correspondant aux différentes phases d'utilisation de l'outil.

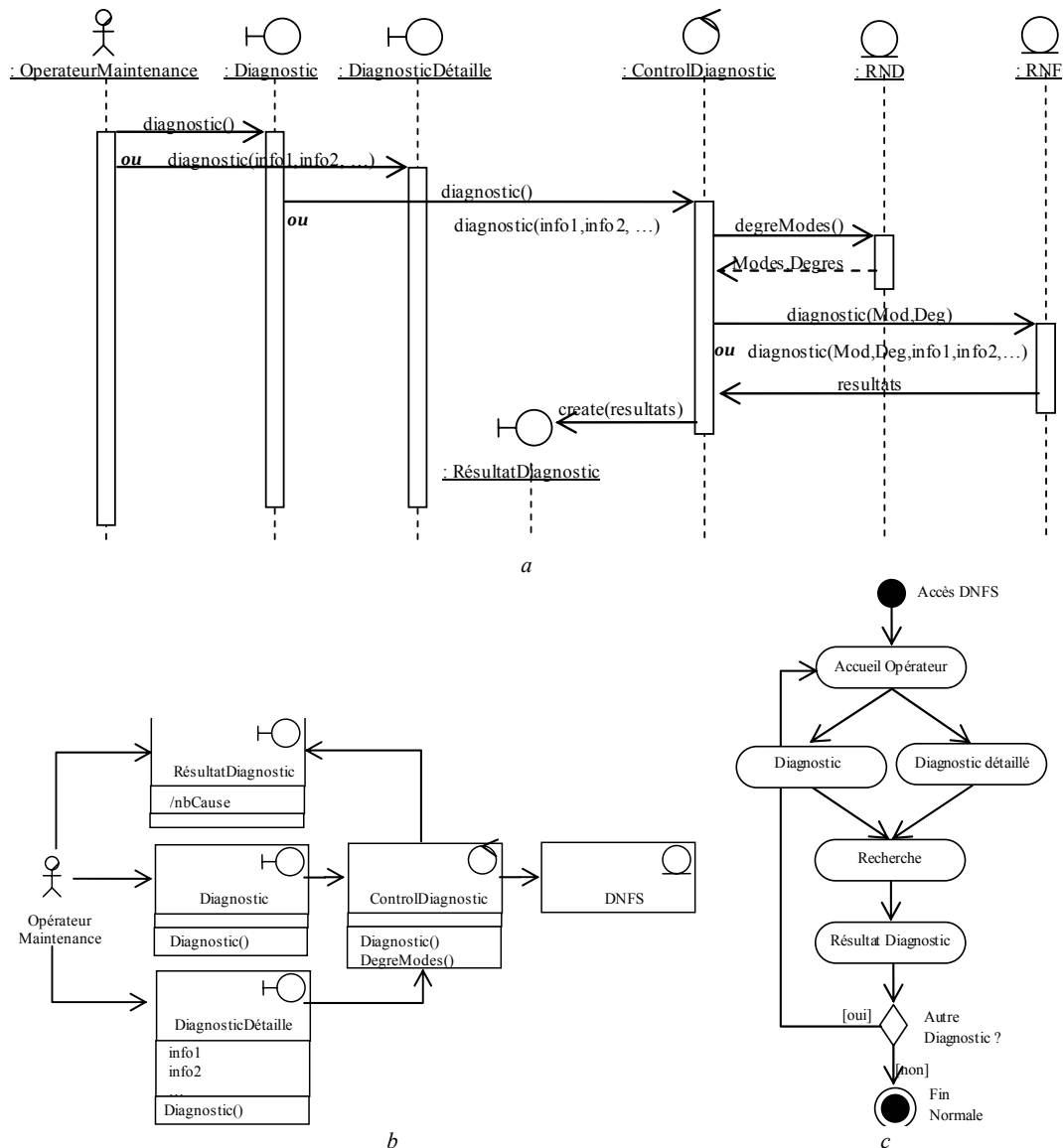


Figure 140. Diagramme de séquence (a), diagramme de classes participantes (b) et diagramme de navigation (c) pour le cas d'utilisation « Demander une aide au Diagnostic »

7.3.3 DESCRIPTION DE L'OUTIL NEURO-FLOU UTILISE

Comme nous l'avons précisé (Figure 131), l'outil est formé de deux parties, une partie neuronale récurrente (RRFR) pour la détection dynamique et une deuxième partie – neuro-floue – pour l'aide au diagnostic.

Comme l'étude de la partie RRFR a déjà été présentée dans les chapitres précédents, nous allons étudier dans ce qui suit, la partie aide au diagnostic. Comme nous l'avons précisé précédemment, le cas d'utilisation « *configuration de l'outil* » nous permet de capitaliser l'information de l'ADD sous la forme d'un système neuro-flou.

Les règles utilisées respectent le principe abductif de recherche de cause et présentent la forme suivante :

- Porte ET : $a \text{ ET } b \rightarrow c : c = a \wedge b$.
- Porte OU : $a \text{ OU } b \rightarrow c : c = a \vee b$.

Les équations obtenues sont entièrement développées mais ne sont pas réduites. La réduction entraîne des pertes d'événements de base (pouvant constituer des causes primaires potentielles). C'est la raison pour laquelle nous n'en faisons pas appel.

Les implications précédentes sont transformées en réseau neuro-flou, dans lequel les défaillances sont représentées par des neurones sigmoïdes ou linéaires :

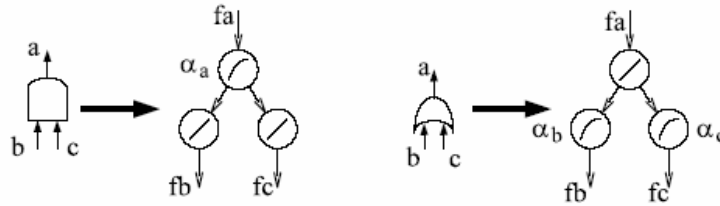


Figure 141. Principe de transformation des portes ET et OU de l'ADD en système neuro-flou

Un événement de base est représenté par un neurone de fonction d'activation linéaire.

Dans la fonction sigmoïdale utilisée :

$$x_i = \frac{e^{\alpha_i \times I_i} - 1}{e^{\alpha_i \times I_i} + 1}, \text{ avec } I_i \text{ l'entrée du neurone et } x_i \text{ sa sortie,}$$

$\alpha_i \leq 1$ représente un facteur de fréquence de la défaillance. Ce paramètre s'avère déterminant lors de la phase de configuration de l'outil.

En phase d'*initialisation*, selon la description UML, l'expert de l'outil extrait des données de l'AMDEC, en les intégrant dans l'outil de diagnostic - en occurrence le système neuro-flou. L'étape d'extraction consiste en la prise en compte, dans le système neuro-flou, de la fréquence et de la gravité de chaque défaillance ou dégradation enregistrée dans l'AMDEC :

```

Pour chaque ligne l de l'AMDEC DO
  IF verifcause(l) THEN
    NewElement (0) = cause (l)
    NewElement (1) = frequency (l)
    NewElement(2) = severity (l)
    ExtAMDEC = ExtAMDEC + NewElement
  ENDIF
RETURN Ext AMDEC

```

Dans un premier temps, pour chaque évènement de base de l'ADD présent dans l'AMDEC, nous allons effectuer le calcul du degré de crédibilité α associé, en fonction de la fréquence d'apparition de cet évènement. Ceci correspond à une opération de fuzzyfication.

En fait, cette fuzzyfication est une version plus élaborée d'une opération courante dans l'industrie lors de la construction d'un AMDEC : il s'agit des tables de correspondances, associant les valeurs du MTBF (Mean Time Between Failures) à des valeurs numériques.

MTBF [h]	Fréquence	α
Jamais apparu		0,8
$3200 < \text{MTBF}$	1	1,3
$2200 < \text{MTBF} \leq 3200$	2	1,9
$1200 < \text{MTBF} \leq 2200$	3	2,5
$200 < \text{MTBF} \leq 1200$	4	3,3
$\text{MTBF} \leq 200$	5	4,4

Tableau 12. Exemple d'équivalence entre le MTBF, la fréquence de l'AMDEC et le facteur α de l'outil d'aide au diagnostic

La valeur de la gravité de la cause constitue une indication supplémentaire fournie à l'opérateur.

A partir du moment où toutes les données disponibles dans l'AMDEC sont exploitées, il est possible que certains évènements de base n'aient pas de fréquence associée, car l'historique n'est pas assez fourni. Dans ce cas, nous considérons que cette cause a des chances infimes (quasi-nulles) d'apparaître. Cette information sera mise à jour ultérieurement, en fonction des nouveaux évènements, données par la GMAO.

Afin de déclencher une procédure de diagnostic à partir d'un symptôme, un lien sera créé entre le symptôme généré par la détection dynamique (RRFR) et l'évènement correspondant du système le diagnostic (système neuro-flou construit à partir de l'ADD). Cet évènement représente l'étape de localisation du défaut ou de la dégradation et se traduit par une entrée supplémentaire au niveau du neurone correspondant à cet évènement :

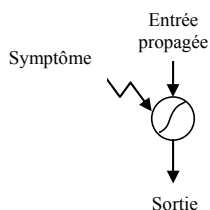


Figure 142. Déclencheurs au niveau d'un neurone du système neuro-flou d'aide au diagnostic

Après les phase de configuration (en utilisant l'ADD) et d'initialisation (utilisation des paramètres de l'AMDEC), l'outil d'aide au diagnostic est autonome et peut servir à *générer des alertes/alarmes* et à proposer une *aide au diagnostic/pronostic*. Durant ces cas d'utilisation, l'outil RRFR de détection dynamique effectue une surveillance continue du sous-système critique surveillé. Quand une défaillance (ou une dégradation) intervient, une alarme (respectivement une alerte) est émise et le système d'aide au diagnostic (respectivement au pronostic) est lancé. En fonction de l'information (symptôme) fournie par le système de détection dynamique, le système propose ainsi à l'opérateur ou aux

responsables de maintenance les causes possibles, leur degré de crédibilité ainsi que la gravité associé.

7.3.4 EXEMPLE D'APPLICATION DE L'OUTIL D'AIDE AU DIAGNOSTIC / PRONOSTIC

Nous illustrons l'utilisation de l'outil neuro-flou d'aide au diagnostic/pronostic sur un exemple simple, associé au système flexible industriel du laboratoire. Ce système permet de déplacer des palettes pouvant recevoir des composants à assembler. Il est équipé de 5 stations de travail, pilotées chacune par un automate programmable industriel. Les 5 automates communiquant entre eux via un réseau local industriel.

La plate-forme est constituée de deux anneaux. Un anneau intérieur et un extérieur. L'anneau intérieur permet de faire circuler les palettes entre toutes les stations. L'anneau extérieur est dédié au traitement des tâches à effectuer sur les palettes. C'est sur cet anneau que sont installés les robots et le manipulateur. L'architecture de l'installation permet de faire transiter une palette d'une station à la suivante sans la faire revenir sur l'anneau intérieur. Ceci est valable pour toutes les stations sauf pour le passage de la station n°2 à la station n°5 :

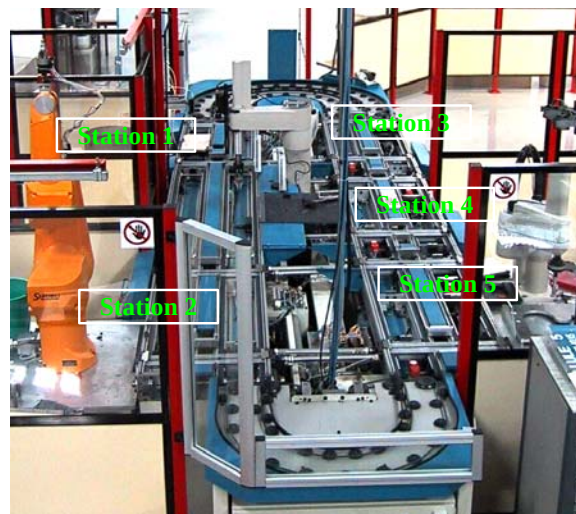


Figure 143. Vue d'ensemble de la plate-forme flexible du LAB

Les palettes circulent sur la plate-forme grâce à un convoyeur à courroies plates. Chaque station dispose d'un moteur électrique, entraînant les courroies. Les palettes sont munies d'une étiquette magnétique qui leur sert de mémoire embarquée. Ces mémoires peuvent être lues dans chaque station grâce à des têtes magnétiques de lecture / écriture (PLE – plots de lecture / écriture). Les étiquettes permettent de mémoriser la gamme d'assemblage des produits, pour savoir sur quel(s) poste(s), les palettes doivent subir une opération.

Toutes les stations sont identiques. Chacune d'elles est composée de trois vérins pneumatiques (pousseur, indexeur, tireur), 9 détecteurs, 6 stoppeurs et de 2 PLE :

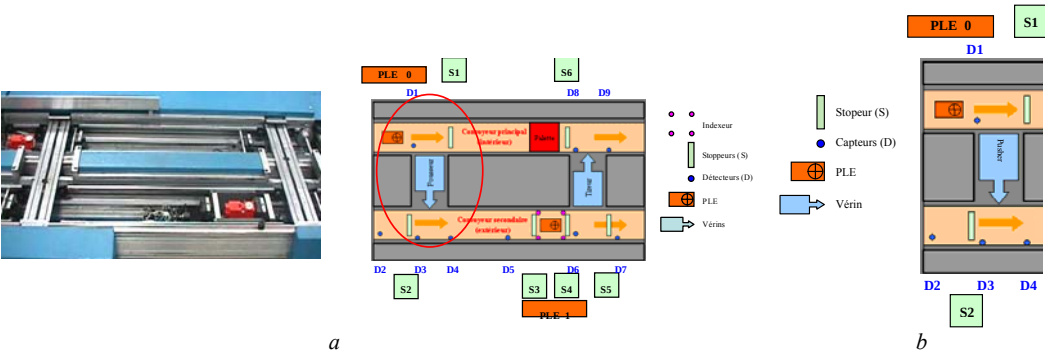


Figure 144. Détail d'une station (a) et entrée du poste(b)

Le sous-système critique que nous avons surveillé représente le module d'entrée d'un poste de travail dont nous présentons un schéma ainsi que l'AMDEC et l'ADD associés.

Désignation	Fonctions	Modes de défaillance	Cause	Intitulé effet système	F	G	ND	C
Entrée de poste	Permettre à la palette de rester sur l'anneau intérieur ou extérieur ou de passer de l'anneau intérieur vers l'anneau extérieur	Palette bloquée sur l'anneau intérieur	Détecteur D1 est en panne	Arrêt	4	2	1	8
Entrée de poste	Permettre à la palette de rester sur l'anneau intérieur ou extérieur ou de passer de l'anneau intérieur vers l'anneau extérieur	Palette bloquée sur l'anneau intérieur	Stoppeur S1 bloqué en position basse	Arrêt	3	4	1	12
Entrée de poste	Permettre à la palette de rester sur l'anneau intérieur ou extérieur ou de passer de l'anneau intérieur vers l'anneau extérieur	Palette bloquée sur l'anneau intérieur	Le pousseur est en panne	Arrêt	1	2	1	2

Tableau 13. AMDEC pour l'entrée du poste

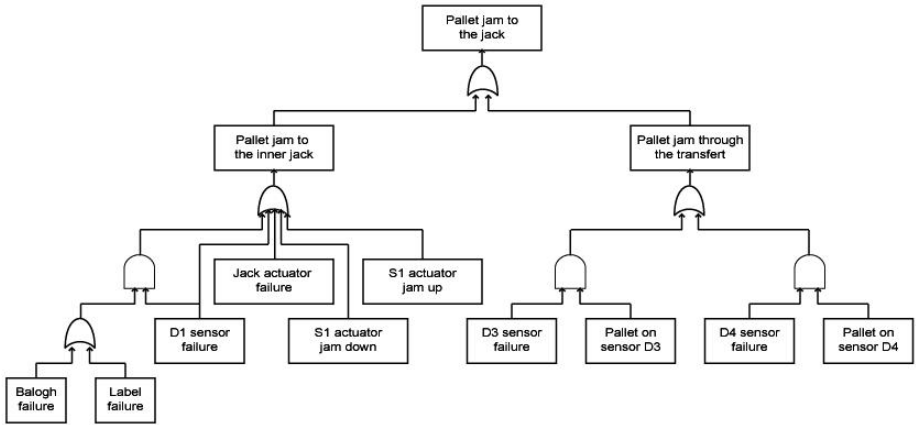


Figure 145. ADD pour l'entrée du poste

En utilisant le principe de transformation des portes ET eu OU de l'arbre de défaillance et dégradations en bloc neuro-flous, nous obtenons le système neuro-flou suivant, qui fait appel à deux types de neurones : linéaires et sigmoïdaux.

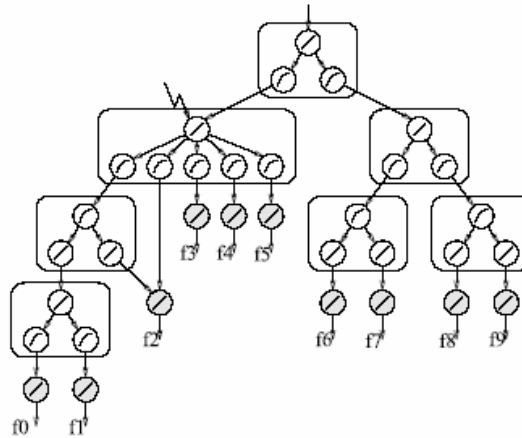


Figure 146. Transformation de l'ADD en système neuro-flou (configuration)

En fonction de la fiabilité du symptôme envoyé par le RRFR (détection dynamique), nous avons une observation d'évènement « Palette bloquée au pousseur » avec un degré de confiance de 0.57. Ce degré est propagé dans le système neuro-flou. La base de connaissances représentée par l'outil neuro-flou nous permet d'obtenir une classification des causes possibles en fonction de leur pertinence (degré de crédibilité). La cause originelle la plus plausible (degré de crédibilité de 83 %) est f2 : défaillance du capteur D1.

Cette information constitue une aide au diagnostic et permet une planification efficace des actions de maintenance du système. Le système neuro-flou permet la mise en œuvre d'une approche abductive de recherche de causes, déclenchés par l'apparition d'un symptôme. Par ailleurs, l'outil est capable d'intégrer la connaissance incertaine ou imprécise, correspondant à l'environnement industriel réel d'expertise et d'action en maintenance.

7.3.5 PROTOTYPAGE DE L'OUTIL NEURO-FLOU D'AIDE AU DIAGNOSTIC

Dans le cadre d'un projet Anvar (**Palluat et al., 2004b**), nous avons eu l'opportunité de développer un prototype logiciel basé sur les principes exposés précédemment, eux même le fruit d'une succession d'étapes d'étude de marché et de spécifications, en conformité avec les exigences industrielles du domaine.

Notre choix de logiciels s'est porté sur le logiciel LabVIEW de National Instrument, un environnement complet, permettant de combiner acquisition, traitement et présentation des données, évitant ainsi le stockage intermédiaire de celles-ci, une caractéristique très utile pour une analyse en temps réel. De plus, son implantation dans un environnement industriel est facilitée car facilement intégrable dans un ordinateur industriel, une carte FPGA ou tout autre cible embarquée. Enfin, ce logiciel s'adapte à de nombreuses plateformes et peut être aisément intégré dans un PDA, l'un des objectifs secondaires du projet.

Dans le cadre de notre application l'outil développé en LabVIEW a été embarqué dans un ordinateur industriel PXI de National Instrument. Des développements sont en cours pour une communication par onde radio grâce à la technologie WiFi. Pour cela, nous avons fait l'acquisition de deux boîtiers convertisseur Ethernet – WiFi, l'un permettant de relier le PXI et l'autre l'ordinateur fixe. Enfin, des tests avec un PDA sont effectués afin de faciliter l'utilisation de notre outil pour un opérateur de maintenance.

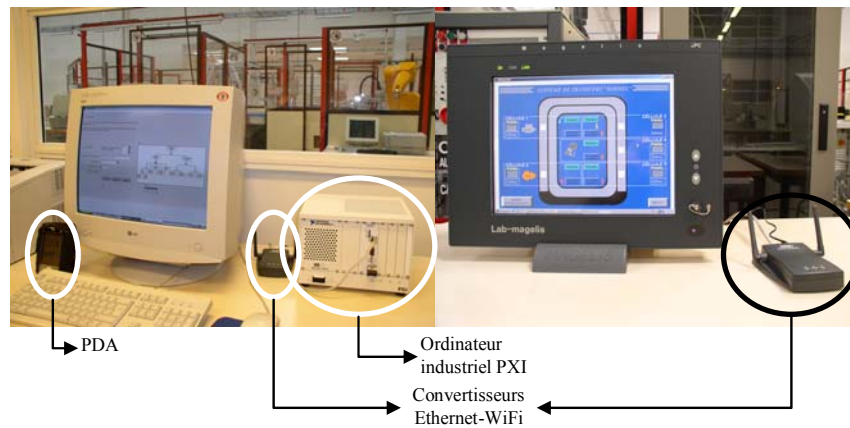


Figure 147. Différents éléments constituant l'outil.

Comme précisé dans la phase de conception de l'outil d'aide au diagnostic, les éléments de départ sont constitués par l'AMDEC et l'ADD associé au système critique surveillé. A partir de ces éléments et en utilisant les spécifications obtenues, nous avons pu développer les principales interfaces, correspondant aux cas d'utilisation souhaités.

Nous présentons ci-dessous les aperçus de trois cas d'utilisation :

- la configuration
- l'initialisation
- l'aide au diagnostic

Le cas d'utilisation de configuration est demandé par l'expert de l'outil suite à une demande du responsable maintenance. L'aperçu suivant montre la phase de transformation de l'Arbre de Défaillance effectué par l'Expert de l'Outil, qui reprend chacune des informations de l'arbre de défaillance (à droite) pour créer le réseau neuro-flou.

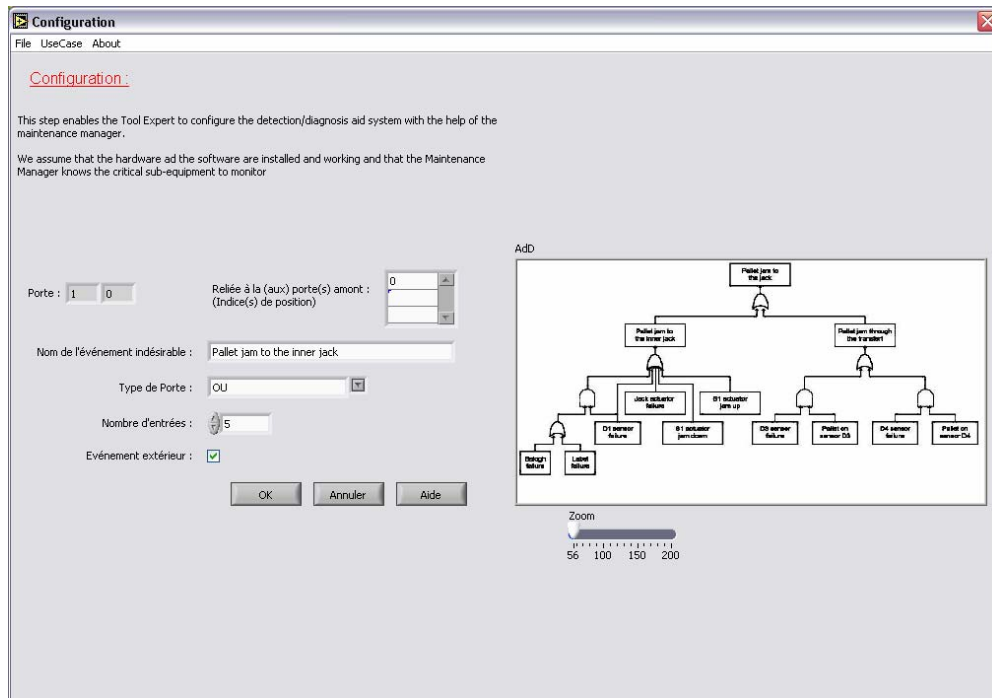


Figure 148. Interface de configuration

Dans la situation d'initialisation (étape qui suit naturellement la configuration), l'expert de l'outil doit ajouter les informations tirées de l'AMDEC. Les aperçus qui suivent montrent différentes phases lors de l'initialisation de l'outil.

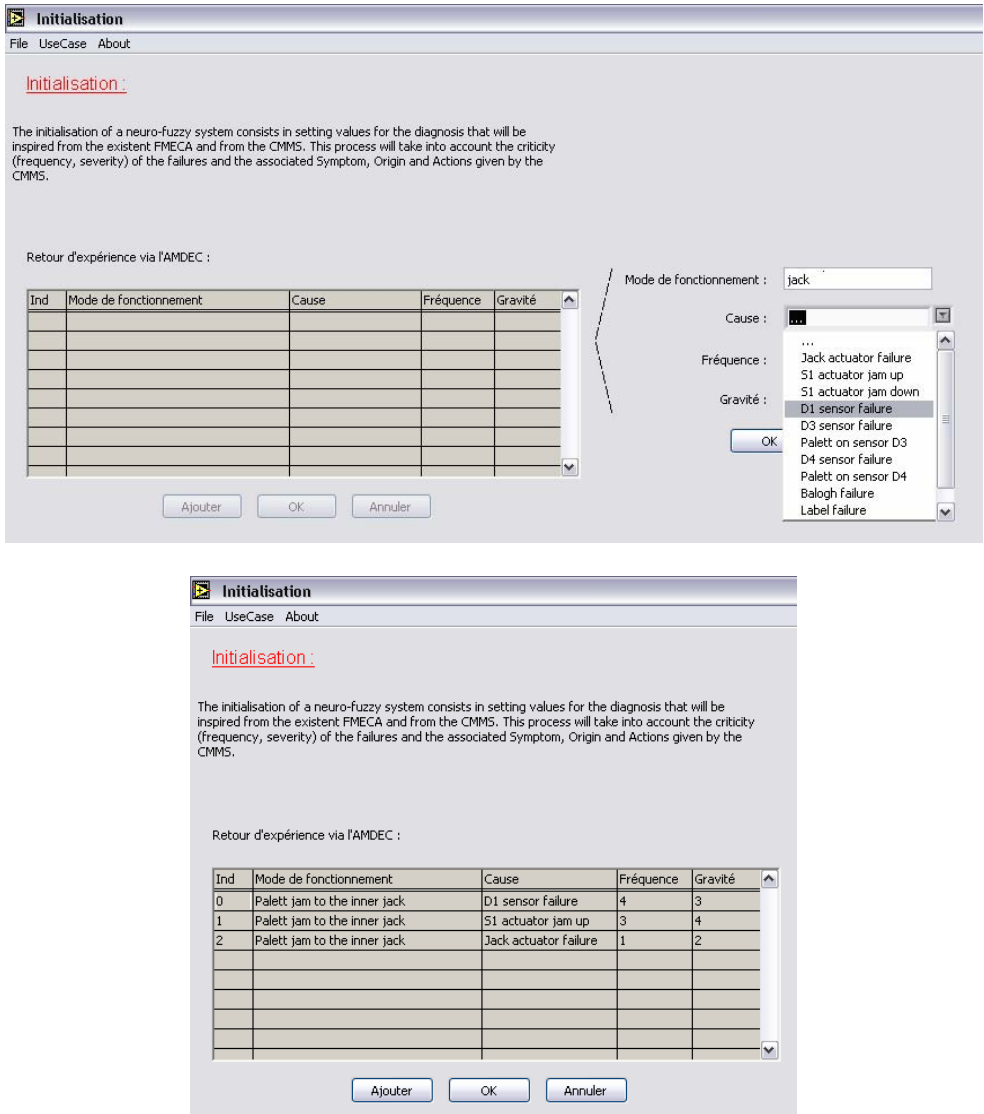


Figure 149. Interfaces d'initialisation

Enfin, la partie détection de l'outil donne l'alerte au responsable maintenance qui décide d'organiser un diagnostic du système. L'opérateur de maintenance se rend sur place, mais le problème possédant différentes causes, il décide de faire appel à l'outil de diagnostic pour l'aider dans son action. Nous avons présenté ci-dessous un exemple de réponse de l'outil, dans ce cas d'utilisation :

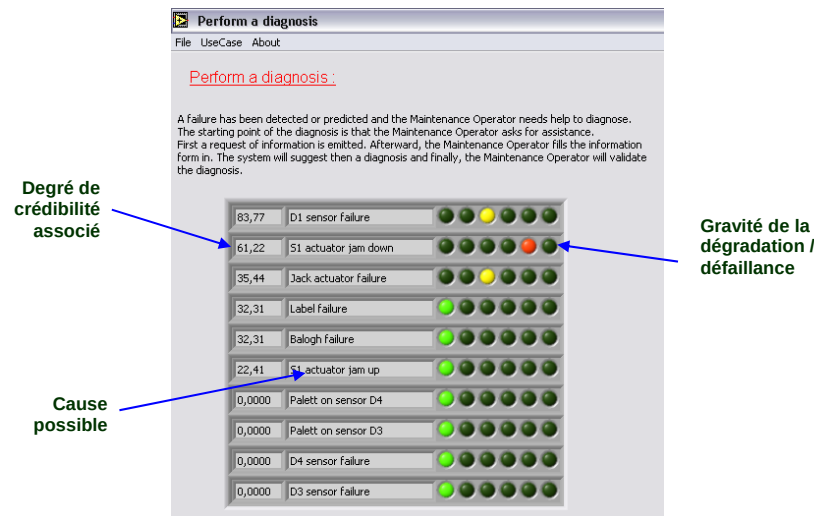


Figure 150. Interface d'aide au diagnostic

Sur ce schéma, nous retrouvons sur le schéma l'ensemble des causes primaires associés au symptôme constaté, avec :

- le nom de la cause possible ;
- la possibilité d'apparition (en pourcentage) ;
- le degré de gravité (associé à un code de cinq couleurs, de vert à rouge, liées à la gravité de la cause, de la moins grave, à la plus grave)

Ces causes sont triées par ordre décroissant des possibilités d'apparition.

7.4 CONCLUSION ET PERSPECTIVES

Nous avons présenté un nouvel outil de surveillance dynamique neuro-flou en suivant l'approche UML. La modélisation se caractérise par sept cas d'utilisation : Création d'un nouvel outil, Configuration de l'outil, Initialisation de l'outil, Réception d'une alerte, Demande d'aide au diagnostic, Mise à jour la configuration de l'outil, et Mise à jour le modèle de l'outil.

Nous avons illustré l'utilisation de notre outil d'aide au diagnostic sous la forme d'un prototype Labview embarqué dans un PC industriel PXI (National Instrument). Ce développement s'est fait dans le cadre d'un projet Anvar.

Le système est capable de travailler en surveillance classique, comme en surveillance dynamique. Dans ce dernier cas, l'action de diagnostic se transforme en action de diagnostic prédictif, donc de pronostic. Dans cette configuration, l'ensemble des principes de base utilisées (apprentissage et modélisation de l'incertain) prennent toute leur ampleur, en permettant la mise en œuvre d'une véritable base de connaissances opérationnelle, avec une mise à jour permanente, en fonction des événements nouveaux reçus de la GMAO et des informations capteurs.

Actuellement, le travail de prototypage continu parallèlement avec la suite de nos recherches sur ce sujet. En effet, des points comme l'apprentissage pratique des séquences dynamiques (validées par simulation) et l'étude de la robustesse de l'apprentissage du système neuro-flou restent à approfondir.

Conclusions Générales et Perspectives de Recherche

BILAN, CONCLUSIONS

En suivant les principales tendances d'évolution des systèmes automatisés et de la productique, nos travaux de recherche se positionnent dans l'optique de la mutation de la Productique, induite par l'intégration des Sciences et Technologies de l'Information et de la Communication.

En effet, cette intégration n'est pas uniquement technique ou technologique, mais permet la mise en œuvre de concepts organisationnels nouveaux et l'élaboration de services inimaginables il y a encore une dizaine d'années.

Dans ce sens, la décentralisation de l'intelligence et sa migration vers les niveaux bas a été largement facilitée, en donnant plus de réactivité aux systèmes de production et en permettant la mise en place de techniques de management novatrices.

Dans le domaine de la maintenance et de manière plus large, dans la Fonction Maintenance, nous retrouvons ainsi des concepts comme la Télémaintenance et la E-maintenance, concepts qui ont constitué – tout au long de nos recherches – un pivot stratégique, qui nous a permis une communication très aisée avec les industriels et les instituts de transfert. Ces domaines ont ainsi fédéré un ensemble d'institutions, industriels de marque et laboratoires prestigieux autour de projets européens, nationaux et régionaux, qui nous ont permis d'acquérir très rapidement un retour d'expérience inestimable.

Dans le cadre des domaines ainsi abordés, nos contributions se sont orientées précisément vers la mise au point de systèmes intelligents distribués de diagnostic et de pronostic, dans le cadre d'une maintenance et une surveillance dite intelligente. Comme nous l'avons montré dans nos états de l'art, ce concept fait partie des tendances dans la recherche mondiale, ce qui nous conforte dans l'idée que nos travaux se situent sinon en avance sur certains points précis, au moins à l'état de l'art mondial au moment du début de nos travaux.

Concernant la surveillance, nos études ont abordé des problématiques liées à la détection dynamique, le diagnostic et le pronostic (diagnostic prédictif). Caractéristiques essentielle du diagnostic, la recherche de cause d'une défaillance a constitué une de nos préoccupations majeures. Dans ce sens, l'utilisation de l'approche abductive de recherche de cause, a été considérée dans la recherche d'outils d'aide au diagnostic liés à des impératifs de :

- *facilité d'acquisition de l'information*
- *prise en compte de l'incertain par l'intégration des aspects liés à l'utilisation de la logique floue dans le processus de diagnostic et de pronostic*
- *généricité des outils développés et*

- *validation des résultats par développement de prototypes de test en collaboration avec des industriels experts du domaine de la maintenance.*

En prenant en compte les tendances évoquées précédemment, les techniques de l'intelligence artificielle ont constitué la colonne vertébrale de nos études, avec des travaux allant de la conception théorique de nouveaux outils dynamiques et la simulation, jusqu'à la validation, le prototypage et l'exploitation industrielle d'outils de détection dynamique de défaillances et de dégradations, ainsi que pour l'aide au diagnostic et au pronostic industriel.

Dans le contexte de la surveillance dynamique, la détection se fait par un apprentissage dynamique permanent, facilitée par la mise au point de *réseaux de neurones récurrent* innovants – les *réseaux récurrents à fonctions de base radiales : RRFR* –, capables d'apporter une approche de reconnaissance locale indispensable au diagnostic et une détection précoce des paliers de dégradation et des fausses alarmes.

Une étude comparative du RRFR par rapport à d'autres structures neuronales dynamiques connues, comme le R2BF et le DGNN, montre l'efficacité du RRFR dans les applications de type surveillance dynamique, en mettant en évidence les avantages liés au paramétrage, aux techniques d'apprentissage et à l'apprentissage dynamique propre à cet outil.

L'exploitation industrielle du nouvel outil constitue une étape importante. Elle permet la mise en œuvre d'une application de surveillance dynamique en temps réel d'un système, en utilisant l'apprentissage dynamique distant via le web. Cette exploitation industrielle a donné lieu à un brevet d'invention, déposé en collaboration avec une SSII bisontine. Ce dépôt conjoint démontre l'intérêt que cette société manifeste par rapport au concept, ainsi que par rapport à son potentiel très intéressant d'application dans le domaine de la surveillance dynamique via les TIC.

La prise en compte de l'incertain, particularité fondamentale du métier de la maintenance, est traitée en utilisant la *logique floue*. L'utilisation de la logique floue dans la surveillance permet d'affiner les informations acquises sur le terrain. Cette technique s'est montrée très prometteuse, grâce à son potentiel d'utilisation en surveillance dynamique et pronostic, avec la possibilité de rester proche de l'opérateur humain, en prenant en compte sa façon de raisonner et en offrant une traçabilité intéressante pour le retour d'expérience, aspect essentiel de toute fonction de maintenance industrielle.

Dans ce contexte, nous avons étudié des outils intégrés de type réseaux de Petri flous (RdPF) dédiés à la surveillance et la reprise des systèmes de production. Ces outils sont basés essentiellement sur la modélisation floue de la variable "instant d'apparition" de l'événement discret. Un mécanisme d'interfaçage approprié est intégré à chaque outil, afin de rendre possible l'utilisation de modules spécialisés dans la surveillance et la reprise. L'intégration de la logique floue dans un tel système d'aide à la décision, permet une formalisation du savoir-faire et de la culture d'entreprise, ainsi qu'une flexibilité et une dimension prédictive très intéressante.

Le modèle réseau de Petri flou pour la surveillance (RdPFS) est ainsi construit à partir des règles logiques induites par l'arbre de défaillance du système surveillé. Nous avons montré que cet outil permet une analyse dynamique des défauts par une approche temporelle floue. Cette analyse présente un grand intérêt pour la surveillance dynamique et – de par sa nature prédictive – dans le pronostic industriel.

L'action de surveillance est suivie de l'action de reprise. Nous avons proposé une modélisation de la fonction de décision floue pour la maintenance à l'aide des réseaux de Petri flous pour la maintenance (RdPFM).

Les modèles du système (RdPTO), du processus de surveillance (RdPFS) et du processus de maintenance (RdPFM), s'échangent des informations par l'intermédiaire de *signaux de synchronisation* flous. Ceci permet une maîtrise avancée des *dégradations* du système durant le fonctionnement, une analyse dynamique permanente de son état de dégradation et le déclenchement d'actions de maintenance préventive. Cette action précoce constitue une fonction de *pronostic*, qui se trouve ainsi intégré dans un outil global et homogène de supervision.

Un prototype de plate-forme logicielle d'aide à la décision en *maintenance prévisionnelle* a été développé en Labview sur la base de la méthodologie de supervision proposée.

Enfin, les avantages des outils neuronaux dynamiques et de la logique floue ont été mis à l'œuvre conjointement dans le cadre de systèmes hybrides *neuro-flous* d'aide au diagnostic.

Essentiels pour l'industrialisation des systèmes proposés, l'information de départ est considérée sous un format industriel classique (ADD, AMDEC, ...) et la mise à jour est assurée par une réelle ouverture du système traduite par des liens permanents avec les systèmes d'acquisition (capteurs, SCADA, ...) et les systèmes de gestion (GMAO, ...). Ces caractéristiques ont largement facilité le prototypage et la validation industrielle des principes proposés, ainsi que la création d'une véritable base de connaissances dynamique, réutilisable par l'entreprise. Ceci est prouvé par le nombre important de collaborations industrielles nourries par nos travaux de recherche.

Pour finir, parmi les verrous technologiques abordés lors de nos contributions, nous pouvons énumérer :

- *la surveillance dynamique par apprentissage en ligne*
- *la rapidité et la flexibilité de l'apprentissage*
- *le temps de réponse*
- *le traitement de grands flux de données*
- *et les méthodes de validation utilisant le test sur des benchmarks et le prototypage industriel.*

PERSPECTIVES DE RECHERCHE

Comme nous l'avons déjà précisé précédemment, les systèmes mécatroniques modernes sont caractérisés par une décentralisation marquée de l'intelligence au niveau de l'action, dans une configuration souvent répartie. Dans ce contexte et vu les activités menées dans le domaine de la surveillance par intelligence artificielle, nos travaux ouvrent des perspectives dans le domaine des *réseaux de capteurs intelligents*. Cette perspective pourra ainsi intégrer des tendances technologiques liées aux connecteurs intelligents (« smart connectors ») basés sur le multiplexage et sur les courants porteurs, afin de réduire le volume de câblage, des points de connexion et afin de mieux prendre en considération les contraintes de sûreté de fonctionnement de ce type de réseaux, contraintes qui – dans un avenir très proche – seront vraisemblablement déterminantes.

Cette problématique fait intervenir des concepts comme :

- l'intelligence artificielle,
- la fusion de capteurs,
- les technologies de l'information et de la communication ainsi que
- les problématiques du temps réel :
 - o le déterminisme des réponses sur le réseau,
 - o la sûreté de fonctionnement des réseaux de capteurs,
 - o la flexibilité,
 - o l'ouverture,

A cela se rajoute un environnement technologique orienté de plus en plus vers la miniaturisation des composants, qui pose des problèmes d'intégration de ces capteurs dans les modules décentralisés.

Dans un contexte régional de création d'un pôle de compétitivité « microtechniques » en Franche-Comté, la problématique de la fabrication des microsystèmes est une problématique présentant un intérêt croissant. En effet, de nombreux prototypes de microsystèmes sont proposés en phase de prototype, avec une création liée à une manière plutôt artisanale, tout à fait compréhensible – vu les technologies innovantes utilisées -, mais qui ne se prête pas à une commercialisation de ces produits. De ce fait, l'étude des microsystèmes de production semble être un domaine porteur et permettant de garder des emplois européens – souvent menacés par la délocalisation.

Dans le domaine de la « micro productique » - domaine abordé depuis 2002 au LAB – je considère pouvoir apporter ma contribution sur des sujets liés à la *surveillance et la sûreté de fonctionnement des microsystèmes de production*. Dans ce sens, il serait nécessaire :

- d'identifier les verrous du micro monde par rapport à une action de type surveillance (détection, diagnostic, pronostic, ...)
- définir les bases d'une approche systémiques dans la surveillance des microsystèmes de production
- utilisation des approches IA ou des approches mixtes (de type boîte grise - modélisation + IA) pour la surveillance des microsystèmes de production

Enfin, le *diagnostic à partir de l'imagerie médicale en utilisant les techniques de l'intelligence artificielle* (correspondant au sujet de ma demande en délégation au laboratoire IPAL – Image Processing and Application Lab – FRE CNRS 2339 Singapour) me semble un domaine où mes compétences en recherche (surveillance par IA) et en enseignement (traitement de l'image) peuvent se donner la main pour aborder une thématique nouvelle, présentant un grand intérêt dans le domaine des biotechnologies, dans lequel la France et en particulier la Franche-Comté sont bien engagées depuis un certain nombre d'années. Sur ce sujet, quelques points me semblent incontournables :

- utilisation conjointe des techniques de traitement de l'image et du diagnostic par imagerie médicale
- mise en œuvre de techniques de l'intelligence artificielle permettant l'apprentissage de l'expérience médicale et la formalisation de raisonnements approximatifs pour l'aide au diagnostic/pronostic.

Table des figures

FIGURE 1.	POIDS DES DISCIPLINES ENSEIGNEES ENTRE 1999 ET 2004 EN HEURES ETUDIANT	34
FIGURE 2.	REPARTITION MOYENNE DES HEURES REELLES PAR TYPE DE FORMATION ENTRE SEPT 1999 ET SEPT 2004.....	34
FIGURE 3.	REPARTITION MOYENNE DES HEURES REELLES PAR TYPE D'ACTIVITE ENTRE SEPT 1999 ET AOUT 2004	34
FIGURE 4.	POIDS DES DISCIPLINES ENSEIGNEES DEPUIS 2004 EN HEURES ETUDIANT	37
FIGURE 5.	REPARTITION MOYENNE DES HEURES REELLES PAR TYPE DE FORMATION APRES 2004 (LMD)	37
FIGURE 6.	REPARTITION DES HEURES REELLES PAR TYPE D'ACTIVITE A PARTIR DE SEPT 2004 (LMD)	37
FIGURE 7.	SCHEMA GENERIQUE DE LA TELEMANTENANCE.	51
FIGURE 8.	SCHEMA GENERIQUE DE LA E-MAINTENANCE.....	51
FIGURE 9.	DIFFERENTES POLITIQUES DE MAINTENANCE	52
FIGURE 10.	HIER : CONCEPT MODULAIRE AVEC INTELLIGENCE CENTRALISEE - DEMAIN : CONCEPT MODULAIRE AVEC INTELLIGENCE REPARTIE.....	53
FIGURE 11.	TENDANCES DANS LES SYSTEMES DE CONTROLE INDUSTRIEL ET D'AUTOMATISMES	53
FIGURE 12.	COMPOSANTES DE LA SURVEILLANCE INDUSTRIELLE.....	55
FIGURE 13.	ARCHITECTURE GENERALE D'UN SYSTEME DE SUPERVISION EN LIGNE.....	56
FIGURE 14.	APPROCHE UNIFIEE POUR LA SURVEILLANCE	56
FIGURE 15.	SURVEILLANCE CLASSIQUE ET SURVEILLANCE DYNAMIQUE.....	57
FIGURE 16.	CLASSIFICATION DES METHODES DE SURVEILLANCE - PREMIERE APPROCHE	58
FIGURE 17.	CLASSIFICATIONS DES OUTILS D'IA SUIVANT LES METHODES DE DIAGNOSTIC.	62
FIGURE 18.	CLASSIFICATION GLOBALE DES METHODES DE SURVEILLANCE.....	62
FIGURE 19.	STRUCTURE D'UN SYSTEME EXPERT	65
FIGURE 20.	REPRESENTATION DES DIFFERENTES ZONES DE DEUX DISTRIBUTIONS GAUSSIENNES.....	65
FIGURE 21.	CYCLE DU RAISONNEMENT A PARTIR DE CAS.....	66
FIGURE 22.	RECONNAISSANCE DES FORMES PAR RESEAU DE NEURONES.	69
FIGURE 23.	EXEMPLE D'ARCHITECTURE DE SYSTEME DE DIAGNOSTIC PAR LOGIQUE FLOUE.	70
FIGURE 24.	EXEMPLE DE RESEAU BAYESIEN POUR LE DIAGNOSTIC DE VOITURE (PIECHOWIAK, 2004).....	73
FIGURE 25.	EXEMPLE DE GRAPHE CONTEXTUEL.....	75
FIGURE 26.	DEPENDANCES ENTRE LES DEFAUTS	76
FIGURE 27.	RESEAU DE DEFAUTS (FAULT NET).....	77
FIGURE 28.	CONVERSION DES PORTES LOGIQUES DE L'ARBRE DE DEFAILLANCE.....	77
FIGURE 29.	RdP ET GRAPHE DE CAUSALITE.....	77
FIGURE 30.	THEMATIQUES DE RECHERCHES ABORDEES. POSITIONNEMENT DES TRAVAUX DE RECHERCHE CO-DIRIGES PAR RAPPORT AUX PROBLEMATIQUES DE LA SURVEILLANCE	83
FIGURE 31.	POSITIONNEMENT DE TRAVAUX CO-ENCADRES PAR RAPPORT AUX DIFFERENTS OUTILS DE L'INTELLIGENCE ARTIFICIELLE	84
FIGURE 32.	APPLICATION DES RESEAUX DE NEURONES EN SURVEILLANCE.	86
FIGURE 33.	DIFFERENCE DES CAPACITES DE GENERALISATION ENTRE LE PMC (A) ET LES RFR (B).....	88
FIGURE 34.	COMPLEXITE D'UN RESEAU DE NEURONES.	89
FIGURE 35.	EXPLOITATION DES RESEAUX STATIQUES POUR DES TRAITEMENTS DYNAMIQUES PAR UTILISATION DE FENETRE TEMPORELLE (A) OU PAR UTILISATION DE CONNEXIONS RECURRENTES (B). ...	95
FIGURE 36.	REPRESENTATION DU TEMPS DANS LES RESEAUX DE NEURONES.	95
FIGURE 37.	EXEMPLE D'UN RESEAU AVEC DES CONNEXIONS A DELAIS.	97
FIGURE 38.	SOMMATION SPATIO-TEMPORELLE : ADDITION DES POTENTIELS EVOQUES A LA FOIS DANS L'ESPACE ET DANS LE TEMPS. A) 2 POTENTIELS EVOQUES (REPERES PAR LES FLECHES) NE DEPASSENT PAS LA VALEUR SEUIL. B) 2 POTENTIELS EVOQUES QUI DEPASSANT LA VALEUR SEUIL GENERENT UN POTENTIEL D'ACTION.	97
FIGURE 39.	REPRESENTATION DU COMPORTEMENT STATIQUE D'UN NEURONE STATIQUE.	98
FIGURE 40.	REPRESENTATION DU COMPORTEMENT DYNAMIQUE D'UN NEURONE RECURRENT.....	98
FIGURE 41.	ARCHITECTURES RECURRENTES : JORDAN (A), ELMAN (B).	99
FIGURE 42.	L'ARCHITECTURE DE MOAKES(A) ET DE M.W. MAK (B) POUR UN RFR RECURRENT.	100
FIGURE 43.	L'ARCHITECTURE DE MIYOSHI POUR UN RFR RECURRENT.....	100

FIGURE 44.	L'ARCHITECTURE DE FRASCONI POUR UN RFR RECURRENT (LE R^2BF).	101
FIGURE 45.	TOPOLOGIE DU DGNN	101
FIGURE 46.	RESEAU RRFR (RESEAUX RECURRENTS A FONCTIONS DE BASE RADIALES).	106
FIGURE 47.	REPRESENTATION DES DIFFERENTES ARCHITECTURES LRGF.	107
FIGURE 48.	ARCHITECTURE GENERALE D'UN RESEAU LRGF AVEC BOUCLAGE DE LA SORTIE (MODELE DE FRASCONI-GORI-SODA). LES τ_1 REPRESENTENT DES RETARDS UNITAIRES.	108
FIGURE 49.	ARCHITECTURE SIMPLIFIEE D'UN RESEAU LRGF AVEC RETOUR DE LA SORTIE.	108
FIGURE 50.	ALLURE DE LA SIGMOÏDE EN FONCTION DU PARAMETRE B.	109
FIGURE 51.	POINTS D'EQUILIBRE DU NEURONE BOUCLE EN FONCTION DU PRODUIT bw_{ii} .	109
FIGURE 52.	COMPORTEMENT D'OUBLI : A) CONVERGENCE DE LA SORTIE DU NEURONE BOUCLE VERS LE POINT D'EQUILIBRE STABLE $a = a_0$ ($w_{ii} > 0$), B) ETAPES DU COMPORTEMENT D'OUBLI	110
FIGURE 53.	COMPORTEMENT D'OUBLI : A) OSCILLATIONS DUES A LA VALEUR NEGATIVE DE L'AUTO-CONNEXION $w_{ii} < 0$. B) ETAPES DU COMPORTEMENT D'OUBLI.	111
FIGURE 54.	MEMORISATION : A) CONVERGENCE DE LA SORTIE DU NEURONE BOUCLE VERS LE POINT D'EQUILIBRE STABLE a_0^+ POUR $a(0) > 0$ ET VERS a_0^- POUR $a(0) < 0$, B) ETAPES DE LA MEMORISATION.	111
FIGURE 55.	LIMITES DE MEMORISATION A DEUX POINTS D'EQUILIBRE STABLE DU NEURONE BOUCLE.	112
FIGURE 56.	LONGUEUR DE LA MEMOIRE DYNAMIQUE DU NEURONE BOUCLE : A) EN FONCTION DU PRODUIT bw_{ii} POUR UN Q DONNE, B) EN FONCTION DE Q AVEC UNE CONFIGURATION DE LONGUEUR DE MEMOIRE MAXIMUM ($bw_{ii} = 2$) OBTENUE A PARTIR DU PREMIER GRAPHE.	113
FIGURE 57.	ARCHITECTURE GENERALE D'UN RESEAU LRGF AVEC BOUCLAGE DE LA SORTIE (MODELE DE PODDAR-UNNIKRISHNAN). LES τ_1 REPRESENTENT DES RETARDS UNITAIRES.	113
FIGURE 58.	INFLUENCE DES PARAMETRES α ET bw_{ii} SUR LE COMPORTEMENT DU NEURONE BOUCLE DE PODDAR-UNNIKRISHNAN : A) COMPORTEMENT D'OUBLI, B) COMPORTEMENT DE MEMORISATION.	114
FIGURE 59.	LONGUEUR DE LA MEMOIRE DU NEURONE DE PODDAR-UNNIKRISHNAN : A) EN FONCTION DE bw_{ii} ET α POUR UN Q DONNE, B) EN FONCTION DE Q ET α AVEC UNE CONFIGURATION DE LONGUEUR DE MEMOIRE MAXIMUM ($bw_{ii} = 2$) OBTENUE A PARTIR DU PREMIER GRAPHE.	114
FIGURE 60.	ARCHITECTURE GENERALE D'UN RESEAU LRGF AVEC BOUCLAGE DE L'ACTIVATION. LES τ_1 REPRESENTENT DES RETARDS UNITAIRES.	115
FIGURE 61.	ARCHITECTURE SIMPLIFIEE D'UN RESEAU LRGF AVEC RETOUR DE L'ACTIVATION	115
FIGURE 62.	COMPORTEMENT D'OUBLI. CONVERGENCE DE L'ACTIVATION DU NEURONE VERS LE POINT D'EQUILIBRE STABLE $a = a_0$: A) SANS OSCILLATIONS POUR DES VALEURS POSITIVES DE L'AUTO-CONNEXION, B) AVEC OSCILLATIONS POUR DES VALEURS NEGATIVES DE L'AUTO-CONNEXION.	116
FIGURE 63.	COMPORTEMENT DE MEMORISATION. DIVERGENCE DE L'ACTIVATION DU NEURONE VERS UNE VALEUR INFINIE A) SANS OSCILLATIONS, B) AVEC OSCILLATIONS.	117
FIGURE 64.	COMPARAISON DE LA LONGUEUR DE LA MEMOIRE DES TROIS ARCHITECTURES LRGF PRESENTEES PRECEDEMMENT : (A) NEURONE A RETOUR LOCAL DE L'ACTIVATION, (B) NEURONE BOUCLE DE FRASCONI-GORI-SODA, (C) NEURONE DE PODDAR-UNNIKRISHNAN.	118
FIGURE 65.	EVOLUTION DU NEURONE BOUCLE PAR RAPPORT A DEUX EVENEMENTS DISTINCTS (SE PRODUISANT A DES INSTANTS DIFFERENTS).	122
FIGURE 66.	A) SYSTEME D'EVENEMENTS DISCRETS ETUDIE, B) MODELISATION PAR RESEAU DE PETRI TEMPORISES, C) ARCHITECTURE DU RRFR POUR L'APPRENTISSAGE DES SEQUENCES DU SYSTEME.	123
FIGURE 67.	EVOLUTION DES SORTIES DES NEURONES RECURRENTS FACE AUX EVENEMENTS ASSOCIES A CHAQUE NEURONE. LES VALEURS DE FIN DE SEQUENCE SONT MEMORISEES PAR LA MEMOIRE STATIQUE.	124
FIGURE 68.	A – RESEAU RRFR AVEC MEMOIRE DYNAMIQUE A UN NEURONE LINEAIRE ET UNE CASCADE DE NEURONES BOUCLES, B – APPRENTISSAGE D'UN PALIER DE DEGRADATION.	124
FIGURE 69.	COMPARAISON ENTRE LA SORTIE DE LA MEMOIRE DYNAMIQUE AVEC NEURONES BOUCLES ET CELLE AVEC NEURONE A RETOUR LOCAL DE L'ACTIVATION.	125
FIGURE 70.	DISSOCIATION ENTRE UN CHANGEMENT BRUSQUE, ET UN PALIER DE DEGRADATION.	125
FIGURE 71.	SCHEMA D'APPLICATION DE LA SURVEILLANCE D'UN MOTEUR.	126
FIGURE 72.	MESURE DE LA VITESSE DE ROTATION ET REPONSE DU NEURONE BOUCLE AVEC LES DEUX TYPES DE PERTURBATIONS : FROTTEMENTS ET BRUITS DE MESURES. MISE EN EVIDENCE DE LA FAUSSE ALARME (BRUIT DE MESURE) MALGRE SON AMPLITUDE IMPORTANTE.	126

FIGURE 73.	APPLICATION DU RESEAU RRFR POUR LA SURVEILLANCE D'UN BRAS DE ROBOT. DIFFERENTS TYPES DE COLLISIONS POSSIBLES : FRONTALE, PAR DERRIERE, A GAUCHE OU A DROITE.	127
FIGURE 74.	REPONSES DES CAPTEURS DE FORCE (F_x , F_y ET F_z) POUR CHAQUE TYPE DE COLLISION.	127
FIGURE 75.	A) SERIE MACKEY-GLASS, B) RRFR POUR LA PREDICTION DE $X(T+6)$	129
FIGURE 76.	RESULTATS DE L'APPLICATION DE L'ALGORITHME RCE SUR LE PROBLEME DE LA SERIE MACKEY-GLASS : A - ERREURS D'ESTIMATION ; B - NOMBRE DE PROTOTYPES CREES	129
FIGURE 77.	COMPARAISON DES TROIS RESEAUX APRES APPRENTISSAGE PAR L'ALGORITHME RCE SUR LE PROBLEME DE PREDICTION DE LA SERIE MACKEYGLASS.	129
FIGURE 78.	INFLUENCE DU PARAMETRE K SUR LES PERFORMANCES DE PREDICTION DU RESEAU RRFR : A) ERREUR MOYENNE DE PREDICTION, B) DEVIATION STANDARD DE L'ERREUR MOYENNE DE PREDICTION.	131
FIGURE 79.	INFLUENCE DU PARAMETRE θ SUR LES PERFORMANCES DE L'ALGORITHME : A) NOMBRE DE PROTOTYPES CREES EN FONCTION DES VARIATIONS DU SEUIL θ , B) LIMITES DE LA ZONE DES PROTOTYPES CREES APRES CONVERGENCE DE L'ALGORITHME DES K-MOYENNES RENFORCEE.	133
FIGURE 80.	STABILITE DU CALCUL DES CENTRES AVEC LA VERSION RENFORCEE DES K-MOYENNES (A) ET (B), PAR RAPPORT A VERSION SIMPLE DE L'ALGORITHME (C).	134
FIGURE 81.	SURVEILLANCE D'UN FOUR A GAZ PAR LE RESEAU DE NEURONES RRFR.	134
FIGURE 82.	A) CONCENTRATION DU CO_2 EN SORTIE DU FOUR A GAZ, B) DEBIT DU GAZ EN ENTREE.	135
FIGURE 83.	NOMBRE DE PROTOTYPES OBTENUS EN FONCTION DES VARIATIONS DU SEUIL θ DE L'ALGORITHME FUZZY MIN MAX.	135
FIGURE 84.	COMPARAISON DES PERFORMANCES ENTRE LES DEUX TYPES DE MEMOIRES DYNAMIQUES ET LE RESEAU RFR STATIQUE SUR UNE APPLICATION DE PREDICTION	136
FIGURE 85.	APPRENTISSAGE DU RESEAU RRFR A REPRODUIRE LES 300 PREMIERES VALEURS DE LA SERIE MACKEY-GLASS, A) DONNEES REELLES DE LA SERIE, B) SORTIE DU RESEAU RRFR AVEC L'EVOLUTION DE LA SORTIE DU NEURONE BOUCLE APRES EXCITATION.	137
FIGURE 86.	APPRENTISSAGE DU RESEAU RRFR DE LA COMMANDE $U(T)$ DU PROBLEME DU FOUR A GAZ, A) RRFR AVEC NEURONE BOUCLE, B) RRFR AVEC NEURONE A RETOUR LOCAL DE L'ACTIVATION.	138
FIGURE 87.	CLASSIFICATION DES RESEAUX DE NEURONES TEMPORELS A L'AIDE DE DEUX CRITERES : TEMPOREL ET ARCHITECTURAL.	139
FIGURE 88.	ARCHITECTURES RRFR (A) ET R^2BF (B) POUR LE PRONOSTIC.	141
FIGURE 89.	SIMPLIFICATION DU RESEAU DGNN (A), STRUCTURE DGNN UTILISEE POUR LE PRONOSTIC(B). .	142
FIGURE 90.	STRUCTURE RRFR (A) ET R^2BF (B) POUR LA SURVEILLANCE DYNAMIQUE	143
FIGURE 91.	SCHEMATISATION DU RESEAU DGNN POUR LA SURVEILLANCE DYNAMIQUE.	144
FIGURE 92.	RESEAU RECURRENT A BASE DE FONCTIONS RADIALES AVEC SES DEUX MEMOIRE (DYNAMIQUE ET STATIQUE) EMBARQUE DANS L'AUTOMATE PROGRAMMABLE.	146
FIGURE 93.	SOLUTION DE SURVEILLANCE EN LIGNE AVEC APPRENTISSAGE DISTANT VIA LE PROTOCOLE DE COMMUNICATION INTERNET TCP/IP	148
FIGURE 94.	DIFFERENTES INTERACTIONS ENTRE CHAQUE COMPOSANT DE L'APPLICATION DE SURVEILLANCE.	149
FIGURE 95.	ORGANIGRAMME DU RESEAU RRFR CHARGE DANS LA CPU DE L'AUTOMATE. LA COMMUNICATION AVEC LE COUPLEUR WEB SE FAIT A TRAVERS DES VARIABLES SITUEES DANS LE COMPOSANT DB.	150
FIGURE 96.	ORGANIGRAMME DU PROCESSUS D'APPRENTISSAGE DU RESEAU DE NEURONES. LE PROGRAMME JAVA CHARGE DANS LE COUPLEUR PEUT ETRE EXECUTE A DISTANCE (TCP/IP). UNE BASE DE DONNEES EST EGALEMENT DISPONIBLE AU NIVEAU DU COUPLEUR ET CONSULTABLE A DISTANCE.	151
FIGURE 97.	INTERFACE (IHM) POUR UN UTILISATEUR PASSIF - VISUALISATION DES VALEURS DES PARAMETRES DE SURVEILLANCE, DU MODE DE FONCTIONNEMENT ET DE L'HISTORIQUE DE L'EQUIPEMENT.	151
FIGURE 98.	INTERFACE (IHM) POUR UN UTILISATEUR AVEC POUVOIR (EXPERT ACTIF) - SEULE PERSONNE A POUVOIR CONFIGURER LE PROGRAMME NEURONAL CHARGE DANS L'UNITE CENTRAL DE L'AUTOMATE.	151
FIGURE 99.	MAQUETTE DE TEST.	152
FIGURE 100.	PLACE DES MODELES DE DETECTION (RdPTO) ET DE SURVEILLANCE DYNAMIQUE (RdPFS)... ..	160
FIGURE 101.	MODELISATION DU SYSTEME SURVEILLE PAR RdPTO. INTEGRATION DE LA DETECTION DIRECTE.	161
FIGURE 102.	PRINCIPE DE MODELISATION DE LA PROPAGATION DES DEFAUTS	163
FIGURE 103.	MODELISATION DE L'IMPLICATION LOGIQUE PAR UNE TRANSITION TEMPORELLE	164
FIGURE 104.	REPRESENTATION DE LA FONCTION F PAR LE RdPFS	165
FIGURE 105.	PROPOSITIONS LOGIQUES ELEMENTAIRES MODELISEES PAR LE RdPFS	166

FIGURE 106.	MODELISATION DES RAISONNEMENTS LOGIQUES CONCURRENTS $D_1 \wedge D_j \rightarrow D_k$ PAR RDPFS.....	169
FIGURE 107.	MODELISATION DES DEFAUTS CONCURRENTS EN ANTECEDENCE PAR RDPFS	170
FIGURE 108.	MODELISATION DE LA SURVEILLANCE SEPARÉE PAR LE RDPFS.....	173
FIGURE 109.	MODELISATION DE LA REGLE $d1 \vee d2 \rightarrow ds1$ PAR RDPFS AVANT A) ET APRES B) FRANCHISSEMENT.....	174
FIGURE 110.	STRUCTURE GLOBALE DE LA BOUCLE DETECTION/MAINTENANCE PREVENTIVE.....	178
FIGURE 111.	MODELISATION D'UNE STATION DE TRAVAIL PAR RdPTO	178
FIGURE 112.	A) REPRESENTATION DE L'EXPRESSION BOOLEENNE F ; B) BASE DE REGLES LOGIQUES FLOUES CORRESPONDANT A F	179
FIGURE 113.	REPRESENTATION DE L'EXPRESSION F PAR LE RDPFS.....	180
FIGURE 114.	MODELISATION DE L'ENSEMBLE DU SYSTEME TEMPS-REEL DE SURVEILLANCE-COMMANDE (ADD = ARBRE DE DEFAILLANCE, SEF = SYSTEME EXPERT FLOU)	181
FIGURE 115.	BASE DES REGLES FLOUES POUR LA MAINTENANCE	182
FIGURE 116.	INTERFACE DU SYSTEME DE SURVEILLANCE DYNAMIQUE.....	182
FIGURE 117.	FONCTIONS D'APPARTENANCE DE LA VARIABLE G, H ET U_1 AUX CLASSES FLOUES.....	183
FIGURE 118.	REPRESENTATION DE LA BASE DE REGLES FLOUES R.....	183
FIGURE 119.	DEFUZZIFICATION DE LA VARIABLE FLOUE U_1	183
FIGURE 120.	LE SYSTEME NEURO-FLOU	188
FIGURE 121.	EXEMPLE DE RESEAU DE NEURONES POUR LA COMMANDE FLOUE.....	189
FIGURE 122.	EXEMPLE D'ASSOCIATION EN SERIE D'UN RESEAU DE NEURONE ET D'UN SYSTEME FLOU	190
FIGURE 123.	EXEMPLE D'ASSOCIATION EN PARALLELE D'UN RESEAU DE NEURONE ET D'UN SYSTEME FLOU..	190
FIGURE 124.	DIFFERENTES ARCHITECTURES DES SYSTEMES NEURO-FLOUS.....	192
FIGURE 125.	MODELE NEFCON AVEC DEUX ENTREES, 5 REGLES ET UNE SORTIE.....	193
FIGURE 126.	MODELE NEFPROX AVEC DEUX ENTREES, 5 REGLES ET DEUX SORTIES	193
FIGURE 127.	LES TROIS DIFFERENTS TYPES DE SANFIS.....	194
FIGURE 128.	EXEMPLE DE GROUPEMENT.....	195
FIGURE 129.	SCHEMA GENERAL DE LA MODELISATION NEURO-FLOUE.....	196
FIGURE 130.	PARTITION FLOUE DES ATTRIBUTS.....	196
FIGURE 131.	SCHEMA DU SYSTEME D'AIDE AU DIAGNOSTIC.....	198
FIGURE 132.	SCHEMA GLOBAL DU PROCESSUS DE MODELISATION UML	199
FIGURE 133.	CAS D'UTILISATION CONCERNANT LE RESPONSABLE DE MAINTENANCE.....	200
FIGURE 134.	CAS D'UTILISATION DE L'OPERATEUR DE MAINTENANCE.....	201
FIGURE 135.	CAS D'UTILISATION DE L'ACTEUR EXPERT DE L'OUTIL.....	201
FIGURE 136.	ORGANISATION DES CAS D'UTILISATION ET DES ACTEURS EN PAQUETAGES.	202
FIGURE 137.	UC OFF-LINE ET ON-LINE.....	202
FIGURE 138.	CARTOGRAPHIE DES CONCEPTS MAJEURS DU DOMAINE.....	203
FIGURE 139.	REPRESENTATION SYNTHETIQUE DES PAQUETAGES ET DE LEUR CONTENU	203
FIGURE 140.	DIAGRAMME DE SEQUENCE (A), DIAGRAMME DE CLASSES PARTICIPANTES (B) ET DIAGRAMME DE NAVIGATION (C) POUR LE CAS D'UTILISATION « DEMANDER UNE AIDE AU DIAGNOSTIC »	204
FIGURE 141.	PRINCIPE DE TRANSFORMATION DES PORTES ET ET OU DE L'ADD EN SYSTEME NEURO-FLOU	205
FIGURE 142.	DECLENCHEURS AU NIVEAU D'UN NEURONE DU SYSTEME NEURO-FLOU D'AIDE AU DIAGNOSTIC	206
FIGURE 143.	VUE D'ENSEMBLE DE LA PLATE-FORME FLEXIBLE DU LAB.....	207
FIGURE 144.	DETAIL D'UNE STATION (A) ET ENTREE DU POSTE(B).....	208
FIGURE 145.	ADD POUR L'ENTREE DU POSTE	208
FIGURE 146.	TRANSFORMATION DE L'ADD EN SYSTEME NEURO-FLOU (CONFIGURATION).....	209
FIGURE 147.	DIFFERENTS ELEMENTS CONSTITUANTS L'OUTIL.....	210
FIGURE 148.	INTERFACE DE CONFIGURATION.....	210
	INTERFACES D'INITIALISATION.....	211
FIGURE 149.		211
FIGURE 150.	INTERFACE D'AIDE AU DIAGNOSTIC.....	212

Liste des tableaux

TABEAU 1.	RESULTATS COMPARATIFS ENTRE LES PERFORMANCES DES ARCHITECTURES TEMPORELLES ...	104
TABEAU 2.	CALCUL DE DISTANCE EUCLIDIENNE ENTRE UN PALIER DE DEGRADATION ET UN PIC DE FAUSSE ALARME (NL = NEURONE LINEAIRE, NR = NEURONE RECURRENT).	126
TABEAU 3.	RESULTATS OBTENUS (RRFR AVEC LES DEUX TYPES DE MEMOIRE ET RFR).	127
TABEAU 4.	ETAPES DE L'ALGORITHME PROPOSE.	133
TABEAU 5.	SYNTHESE DES ESSAIS COMPARATIFS SUR LE PRONOSTIC	142
TABEAU 6.	TEMPS D'APPRENTISSAGE ET DE RECONNAISSANCE POUR LA SURVEILLANCE DYNAMIQUE	144
TABEAU 7.	ERREURS EN SURVEILLANCE DYNAMIQUE	144
TABEAU 8.	DIFFERENTS TEMPS DE CYCLE EN FONCTION DE PLUSIEURS DIMENSIONS DU RESEAU RRFR...	152
TABEAU 9.	DIFFERENTES APPROCHES DES RESEAUX DE PETRI FLOUS.....	158
TABEAU 10.	COMPARAISON ENTRE LA LOGIQUE FLOUE ET LES RESEAUX DE NEURONES	189
TABEAU 11.	DEGRES FLOUS DE RELATION ENTRE LES CLASSES ET LES SOUS DOMAINES DE LA PARTITION...	197
TABEAU 12.	EXEMPLE D'EQUIVALENCE ENTRE LE MTBF, LA FREQUENCE DE L'AMDEC ET LE FACTEUR α DE L'OUTIL D'AIDE AU DIAGNOSTIC	206
TABEAU 13.	AMDEC POUR L'ENTREE DU POSTE	208

Références Bibliographiques

- Aamodt, A. et E. Plaza (1994). Case-Based Reasoning : Foundational Issues, Methodological Variations and Systems Approaches, AI Communication, IOS Press, Vol. 7: 1, pp. 39-59.
- Adamson M. J. et R.I. Damper (1996). A recurrent network that learns to pronounce English text, *Proceedings of International Conference on Spoken Language Processing (ICSLP'96)* 4, pp. 1704-1707.
- Aghasaryan, A., R. Boubour, E. Fabre, C. Jard et A. Benveniste (1997). A Petri Net Approach to fault detection and diagnosis in distributed systems, *Armen Publication interne n°1117 IRISA*.
- Aghasaryan, A., (1998). *Formalisme HMM pour les réseaux de Pétri partiellement stochastiques : Application au diagnostic de pannes dans les systèmes répartis*, Thèse de doctorat université de Rennes1.
- Ahmedou, O. et N. Bonnet (1997). Une amélioration de l'algorithme possibiliste de classification non supervisée (PCM), *LFA'97 - Conférence sur la Logique floue et ses applications*.
- Almeida, L.B. (1988). Backpropagation in Perceptrons with feedback, in *NATO ASI Series, Vol. F41, Neural Computers*, Edited by R. Eckmiller and Ch. V. d. Malsburg.
- Amit, D.J. (1988). Neural Network Counting Chimes, *Proceedings of the National Academy of Sciences USA*, 85, pp. 2141-2145.
- Amodeo, L., (1999). *Contribution à la modélisation et la réduction des systèmes de production par les perturbations singulières*, Thèse de Doctorat, Université de Franche-Comté, Belfort, France.
- Anglano, C., L. Portinale (1994). B-W Analysis: a Backward Reachability Analysis for Diagnostic Problem Solving Suitable to Parallel Implementation, *Proceeding of the 15th International Conference on Application and Theory of Petri Nets*, Zaragoza, Espagne.
- Atiya, A.F., S.M. El-Shoura, S.I. Shaheen et M.S. El-Sherif (1999). A Comparison Between Neural Network Forecasting Techniques – Case Study : River Flow Forecasting, *IEEE Transactions on Neural Networks*, Vol. 10, N°2, pp. 402-409.
- Atiya, A.F. et A.G. Parlos (2000). New Results on Recurrent Network Training : Unifying the Algorithms and Accelerating Convergence, *IEEE Transactions on Neural Networks*, Vol. 11, N°3, pp. 697-709.
- Aussem, A. (1995). *Théorie et Application des Réseaux de Neurones Récurrents et Dynamiques à la Prédiction, à la Modélisation et au Contrôle Adaptatif des Processus Dynamiques*, Thèse de Doctorat, Université René Descartes – Paris V.
- Aussem, A. (2002). *Le Calcul du Gradient d'Erreur dans les Réseaux de Neurones Discrets Bouclés à Délais : Application aux Télécom et aux Sciences*

- Environnementales*, Habilitation à Diriger des Recherches, Université Blaise Pascal Clermont-Ferrand II/ France.
- Ayat, N.E., M. Cheriet et C.Y. Suen (2000). *Un système neuro-flou pour la reconnaissance de montants numériques de chèques arabes*, Rapport du laboratoire de l'Intelligence Artificielle de l'Université du Québec.
- Back, A.D. et A.C. Tsoi (1990). A Time Series Modeling Methodology Using FIR and IIR Synapses, *Proceeding Workshop on Neural Networks for Statistical and Economic Data*, Dublin, DOSES, Statistical Office of European Communities, F. Murtagh Ed., pp. 187-194.
- Basseville, M. (1988). Detecting Changes in Signals and Systems – A Survey, *Automatica*, Vol. 24, N°3, pp. 309-326.
- Basseville, M. et I. Nikiforov (1993). *Detection of Abrupt Changes – Theory and Applications*, Information and System Sciences Serie, Prentice Hall, Englewood Cliffs, N.J.
- Basseville, M. et M-O. Cordier (1996). Surveillance et diagnostic de systèmes dynamiques: approche complémentaire du traitement de signal et de l'intelligence artificielle, *Rapport INRIA*, N°2861.
- Bengio, Y., P. Frasconi, P. Simard (1993). The problem of learning long-term dependencies in recurrent networks, *Proc. of the IEEE International Conference on Neural Networks*, San Francisco, pp. 1183-1195.
- Bernauer, E. et H. Demmou (1993). Temporal sequence learning with neural networks for process fault détection, *IEEE International Conference on Systems, Man, and Cybernetics, IEEE-SMC 93*, vol. 2, Le Touquet, France, pp. 375-380.
- Bernauer, E., (1996). *Les réseaux de neurones et l'aide au diagnostic : un modèle de neurones bouclés pour l'apprentissage de séquences temporelles*, Thèse de doctorat, LAAS/Toulouse.
- Béroule, D. (1985). *Un Modèle de Mémoire Adaptative, Dynamique et Associative pour le Traitement de la Parole*, Thèse de Doctorat, Paris XI Orsay.
- Berthold, M. R. (1994a). A Time Delay Radial Basis Function Network for Phoneme Recognition, *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 7, pp. 4470-4473.
- Berthold, M. R. (1994b). The TDRBF: A shift invariant radial basis function network, *Proceedings of the fourth Irish Neural Network Conference*, Dublin, pp.7-12.
- Berthold, M. R. et J. Diamond (1995). Boosting the Performance of RBF Networks with Dynamic Decay Adjustment, *Advances in Neural Information Processing Systems*, Gerald Tesauro, David S. Touretzky, and Todd K. Leen editors, vol. 7, p. 521-528, MIT Press, Cambridge, MA.
- Bezdek, J.C. (1974). Numerical Taxonomy with Fuzzy Sets, *Journal of Mathematical Biology*, Vol.1, pp. 57-71.
- Billings, S.A., et C.F. Fung (1995). Recurrent Radial Basis Function Networks for Adaptive Noise Cancellation. *Neural Networks*, Elsevier Science Publishers, Vol. 8, N°2, pp. 273-290.
- Böhme, T., C.S. Cox, N. Valentin et T. Denoeux (1999). Comparaison of Autoassociative Neural Networks and Kohonen Maps for Signal Failure Detection and

- Reconstruction, In C.H. Dagli et al., editors, *Intelligent Engineering Systems through Artificial Neural Networks* 9, 637-644, New-York : ASME Press.
- Bonneval, A. (1993). *Mécanismes de reprise dans les systèmes de commande à événements discrets*, Thèse de doctorat, Informatique Industrielle, Université Paul Sabatier de Toulouse.
- Bouchon-Meunier, B., (1994). *La logique floue*. Collection Que sais-je, n.2702, Ed. Presses Universitaires de France, Paris.
- Bouchon-Meunier, B., (1995). *La Logique Floue et ses applications*, Edition Addison-Wesley.
- Bouchon-Meunier, B. et C. Marsala (2003). *Logique floue, principes, aide à la décision*, Ed. Hermes.
- Brézillon, P., R. Naveiro, M. Cavalcanti and J.-Ch. Pomerol (2000). *SART : An Intelligent Assistant System for Subway Control*, *Pesquisa Opreacional*, 20(2) pp. 247-268.
- Brézillon, P., L. Pasquier et J.-Ch. Pomerol (2001). Reasoning with contextual graphs, *European Journal of Research*, n° : 136, pp. 290-298.
- Brézillon, P. (2003). Context-based Modeling of Operator's Practices by Contextual Graphs, *Human Centered Processes: 14th Mini Euro Conference*, Luxembourg.
- Brusoni, V., L. Console, P. Terenziani et D. Theseider Dupré (1995). Characterizing temporal abductive diagnosis, *Proc. of International Workshop on Principles of Diagnosis*, pp. 34-40.
- Brusoni, V., L. Console, P. Terenziani et D. Theseider Dupré (1997). An Efficient Algorithm for Temporal Abduction, *Lecture notes in Artificial Intelligence* 1321 pp. 195-206.
- Bugarin, A. et S. Barro (1994). Fuzzy Reasoning Supported by Petri Nets, *IEEE Transaction on Fuzzy System*, 2, 2, pp. 135-150.
- Burg, T., Tschichold N., « Dynamic neurons with negative local feedback for time-series prediction », *Proc. Int. Workshop on Advanced Black-box Techniques for Nonlinear Modelling*, Katholieke Universiteit Leuven, Belgium, pp. 129-133, 1998.
- Camargo, H. (1998). The Design of Knowledge Bases through Hierarchical High Level Fuzzy Petri Nets, *Fuzziness in Petri Nets*, Springer-Verlag, J. Cardoso & H. Camargo (eds.) 1998.
- Camarinha-Matos, L.M., L. Seabra Lopes, et J. Barata (1996). Integration and Learning in Supervision of Flexible Assembly Systems, *IEEE Transactions on Robotics and Automation*, vol. 12, n°2, pp. 202-219.
- Campolucci, P., A. Uncini, F. Piazza et B.D. Rao (1999). *On-Line Learning Algorithms for Locally Recurrent Neural Networks*, *IEEE Transactions on Neural Networks*, Vol. 10, N°2, pp. 253-271.
- Cao, T., A.C. Sanderson (1993). A fuzzy Petri nets approach to reasoning about uncertainty in robotic system, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 317-322.
- Cardoso, J., R. Valette et D. Dubois (1990). Petri nets with uncertain markings, *LNCS Advances in Petri nets*, Vol. 483, G. Rozenberg, Ed. Springer Verlag, pp. 64 - 78.

- Cardoso, J. (1995). *Sur les réseaux de Petri avec marquage flou*, Thèse de doctorat, Informatique Industrielle, Université Paul Sabatier de Toulouse.
- Cardoso, J., R. Valette, D. Dubois (1996a). Fuzzy Petri Nets : An Overview, *13th Word Congress of IFAC*, Vol I : Identification II, Discrete Event Systems, San Francisco, CA, USA, pp. 443-448.
- Cardoso, J., G. Bittencourt, B. Pradin-Chezalviel et R. Valette (1996b). Petri net and logic, *Rapport LAAS*, N°96137.
- Cardoso, J. et H. Camargo (1999). Studies in Fuzziness and soft computing, vol. 22, *Physica Verlag*, ISBN 3-7908-1158-0, pp.3-24
- Chaillet-Subias, A. (1995). *Approche multi-modèles pour la commande et la surveillance en temps réel des systèmes a événements discrets*, Thèse de doctorat, Informatique Industrielle, Université Paul Sabatier de Toulouse.
- Chaillet-Subias, A. M. Combacau et E., Zamai (1996). Une approche de la surveillance en temps réel des systèmes a événements discrets, *Rapport LAAS*.
- Chang, F.J., J.M. Liang et Y.C. Chen (2001). Flood Forecasting Using Radial Basis Function Neural Network, *IEEE Transactions on Systems, Man and Cybernetics – Part C : Applications and Reviews*, Vol. 31, N° 4.
- Chappelier, J.C. (1996). *RST : une architecture connexionniste pour la prise en compte de relations spatiales et temporelles*. Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications de Paris.
- Chappelier, J.C. et A. Grumbach (1996). A Kohonen Map for Temporal Sequences, *Proceeding of neural Networks and Their Application, NEURAP'96, IUSPIM*, Marseille, France, pp. 104-110.
- Chen, M., D.A. Linkens (2001). *A sytematic neuro-fuzzy modeling framework with application to material property prediction*, *IEEE Trans, Cybern*, vol, 31, pp 781-790.
- Chen, S.-M., J.-S. Ke et J.-F. Chang (1990). Knowledge Representation Using Fuzzy Petri Nets, *IEEE Transaction on knowledge and data engineering*, Vol. 2, No.3.
- Cheng, Y.-L., (2000). Uncertainties in Fault Tree Analysis, *Tamkang Journal of Science and Engineering*, Vol.3, No. 1, Chine.
- Combacau, M., (1991). *Commande et Surveillance des Systèmes a Évènements Discrets Complexes : Application aux Ateliers Flexibles*, Thèse de doctorat, Université Paul Sabatier, Toulouse.
- Combacau, M., P. Berruet, P. Charbonnaud, A. Khatab (2000). Réflexion sur la terminologie "Surveillance – Supervision", *Groupeement de Recherche en Productique Systèmes de Production Sûrs de Fonctionnement SPSF*, état des réflexions⁵⁸.
- Combastel, C., (2000). *Méthodes d'aide à la décision pour la détection et la localisation de défauts dans les entraînements électriques*, Thèse de Doctorat INPG, Grenoble.
- Cussenot, C., (1996). *Surveillance et diagnostic de la chaîne de dépollution d'une automobile*, Thèse de doctorat, Université de Rennes 1.

⁵⁸ <http://www.laas.fr/%7Ecombacau/SPSF/sursup.html>

- Daniel, O. (1995). *Les réseaux de Pétri stochastiques pour l'évaluation des attributs de la sûreté de fonctionnement des systèmes manufacturiers*, Thèse de doctorat, Laboratoire d'Automatique de Grenoble, Institut National Polytechnique de Grenoble.
- Dash, S. et V. Venkatasubramanian (2000). Challenges in the industrial applications of fault diagnostic systems, *Proceedings of the Conference on Process Systems Engineering Computing & Chemical Engineering*, Keystone, Colorado, pp. 785-791.
- Day, S.P. et M.R. Davenport (1993). Continuous Time Temporal Back Propagation with Adaptable Time Delays, *IEEE Transaction on Neural Networks*, Vol. 4, N°2, p. 348-354.
- Deckker, R. et R.A. Skarf (1998). On the impact of optimisation models in maintenance decision making: state of the art, *Reliability Engineering and System Safety*, vol. 60, p. 111-119.
- Demmou, H. et E. Bernauer (1995). Using Self-Recurrent Neurons for Fault Detection and Diagnosis, *3^{ème} IFAC en WIMS*, Roumanie.
- Denoeux, T., Govaert G. (1996). « Combined supervised and unsupervised learning for system diagnosis using Dempster-Shafer theory », In P. Borne, M. Staroswiecki, J. P. Cassar and S. El Khattabi (Eds) *CESA'96 IMACS Multiconference, Computational Engineering in Systems Applications. Symposium on Control, Optimization and Supervision*, volume 1, pp. 104-109, Lille.
- Denoeux, T., M. Masson et B. Dubuisson (1998). Advanced pattern recognition techniques for system monitoring and diagnosis: a survey, *Journal Européen des Systèmes Automatisés (RAIRO-APII-JESA)*, 31(9-10), pp.1509-1539.
- Devauchelle, G., (1991). *Diagnostic mécanique des fatigues sur les structures soumises à des vibrations en ambiance de travail*, Thèse de Doctorat, Université de Paris IX Dauphine.
- DeVries, B., et J.C. Principe (1991). A Theory for Neural Networks with Time Delays, *Advances in Neural Information Processing Systems*, 3, R.P. Lippmann Ed., pp. 162-168.
- Dourado, A., A. Cardoso, P. Gil, J. Henriques, R. Paiva, C. Pereira, A. Santos, J. Victor (1999). Intelligent control and supervision at Centro de Informatic e Sistemas da Universidade de Coimbra, *Rapport CISUC - Centro de Informatic e Sistemas da Unoversidade de Coimbra*.
- Dreyfus, G., J-M. Martinez, M. Samuelides, M.B. Gordon, F. Badran, S. Thiria et L. Hérault (2002). *Réseaux de neurone, Méthodologies et Application*, Paris, Edition Eyrolles.
- Dubois, D. et H. Prade (1980). *Fuzzy Sets and Systems*, Academic Press.
- Dubois, D. et H. Prade (1989). Processing fuzzy temporal knowledge, *IEEE Trans. On Syst. Man and Cyb.*, 14, n°4.
- Dubois, D. et S. Gentil (1994). Intelligence Artificielle et Automatique, *Revue d'Intelligence Artificielle*, Vol. 8, N°1, pp. 7-27.
- Dubuisson, B., (1990). *Diagnostic et reconnaissance des formes*, Paris, Edition Hermès.

- Dubuisson, B., E. Boutleux, P. Dague, T. Denoeux, E. Didelet, Y. Gandvalet et M. Masson (2001). *Diagnostic, Intelligence Artificielle et reconnaissance de formes*, Ed. Hermes.
- Elman, J. (1990). Finding Structure in Time, *Cognitive Science*, 2(14), pp. 179-211.
- Emmanouilidis, C., J. MacIntyre et C. Cox (1998). *Neurofuzzy Computing Aided Machine Fault Diagnostic*, Proc. Of JCIS'98, The Fourth Joint Conference on Information Sciences, Research Triangle Park, North California, Usa, Vol. 1. pp.207-210.
- Evsukoff, A., (1998). *Raisonnement approché pour la surveillance des procédés*, Thèse de Doctorat, INPG, Grenoble.
- Fabre, E., A. Benveniste, C. Jard et M. Smith (2001) Diagnosis of distributed discrete event systems, a net unfolding approach, *Publication Interne IRISA*.
- Fache, A., O. Dubois et A. Billat (1999). On the invertibility of the RBF model in a predictive control strategy , *European Symposium on Artificial Intelligence Networks*, Bruges-Belgique, pp. 381-386.
- Ferariu, L. et T. Marcu (2002). Evolutionary Design of Dynamic Neural Networks Applied to System Identification, *15th IFAC World Congress*, Barcelone, Espagne.
- Frank, P.M., (1990). Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge Based Redundancy – A survey and New Results, *Automatica*, Vol. 26, pp. 459-474.
- Frasconi, P., M. Gori et G. Soda (1992). Local Feedback Multilayered Networks, *Neural Computation*, Vol. 4, pp. 120-130.
- Frasconi, P., M. Gori, M. Maggini et G. Soda (1995). Unified Integration of Explicit Knowledge and Learning by Exemple in Recurrent Networks, *IEEE Transaction on Knowledge and Data Engineering*, Vol. 7, N° 2.
- Frasconi, P., M. Gori, M. Maggini et G. Soda (1996). Representation of Finite State Automata in Recurrent Radial Basis Function Networks, *Machine Learning*, Vol. 23, pp. 5-32.
- Freitas, N., I.M. Macleod et J.S. Maltz (1999). Neural networks for pneumatic actuator fault détection, *Transactions of the SAIEE*, vol. 90, n° 1, pp. 28-34.
- Frélicot, C. (1992). *Un système adaptatif de diagnostic prédictif par reconnaissance des formes floues*, Thèse de doctorat, Université de technologie de Compiègne.
- Fukunaga, K., (1990). *Statistical Pattern Recognition*, Academic Press, 2^{ème} édition.
- Garcia-Salicetti, S. (1996). *Une Approche Neuronale Prédictive pour la Reconnaissance en-ligne de l'Ecriture Cursive*, Thèse de doctorat, Université de Paris VI.
- Garg, M.L., S.I. Ahsan et V. Gupta (1995). A fuzzy net for knowledge representation and reasoning, *Inform. Process. Letters* 39, 165-171.
- Gertler, J.J., (1988). Survey of model-based failure detection and isolation in complexe systems, *IEEE Control Systems Magazine*, Vol. 8, N° 6, p. 3-11.
- Gertler, J.J., (1998). *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker, Inc.
- A. Ghernaut, **D. Racocceanu** et N. Zerhouni (2005). Approche abductive utilisant les réseaux de Petri Flous pour le diagnostic et la qualité des systèmes mécatroniques, 6^{ème} *Congrès international du Génie Industriel*, Besançon.

- Ghosh, J., S. Beck et L. Deuser (1992). A Neural Network Based Hybrid System for Detection, Characterization and Classification of Short-Duration Oceanic Signals, *IEEE Jl. of Ocean Engineering*, vol. 17, n° 4, pp. 351-363.
- Ghosh, J. et A. Nag (2000) *Radial Basis Function Network*, in *Radial Basis Function Neural Network Theory and Applications*, R. J. Howlett and L. C. Jain (Eds), Physica-Verlag.
- Giles, C. L., C.B. Miller, D. Chen, G.Z. Sun, H.H. Chen et Y.C. Lee (1992). Extracting and learning an unknown grammar with recurrent neural networks, In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems* 4, pages 317–324, San Mateo, CA, Morgan Kaufmann Publishers.
- Gomes, L. (1998). Fuzzy modelling reactive real-time systems control using Reactive Petri nets, *Fuzziness in Petri Nets*, Springer-Verlag, J. Cardoso & H. Camargo (eds.).
- Gomes, L. et A. Steiger-Garção (1995). Programmable controller design based on a synchronized Coloured Petri net model and integrating fuzzy reasoning, *Lecture Notes in Computer Science* 935, pp. 218-237, Springer Verlag.
- Gori, M. (1989). An Extension of BPS, *Proceeding of the 2nd Intern Workshop on Neural Networks and their Applications*, Nimes, France.
- Gori, M., Y. Bengio et R.D. Mori (1989). BPS : A Learning Algorithm for Capturing the Dynamic Nature of Speech, *International Joint Conference on Neural Networks*, Vol. 2, pp. 417-423.
- Gori, M. et G. Soda (1990). Temporal Pattern Recognition Using EBPS, *EURASIP*.
- Grosclaude, I. et R. Quiniou (2000). Dealing with Interacting Faults in Temporal Abductive Diagnosis, *Proc. of the 8th International Workshop on Principles of Diagnosis*.
- Grosclaude, I., (2001a). *Diagnostic Abductif Temporel : Scenarios de Pannes, modèles causaux et traitement de l'interaction*, Thèse de Doctorat, Université de Rennes 1.
- Grosclaude, I., M.-O. Cordier et R. Quiniou (2001b). Causal interaction : from a High Level Representation to a Operational Event-Based Representation, *International Joint Conference on Artificial Intelligence, IJCAI 2001*.
- Hanna, M. (1994). Determination of product quality from an FMS cell using Petri nets, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol.2 San Antonio, USA, pp. 2002-2007.
- Hashem, S., P.E. Keller, R.T. Kouzes et L.J. Kangas (1995). Neural Network Based Data Analysis for Chemical Sensor Analysis», *Proceedings of SPIE's AeroSense '95 Conference*, Orlando, Floride, USA.
- Hassoun, M.H. (1995). *Fundamentals of Artificial Neural Networks*, Cambridge, MA : MIT Press.
- Haykin, S. (1999). *Neural Networks : a comprehensive foundation*, Prentice Hall International Editions, 2nd edition, ISBN 0-13-908385-5.
- Hebb, D.O. (1949). *The organization of the behavior*, New York :Wiley, 1949.
- Hernandez, N.G., *Système de diagnostic par Réseaux de Neurones et Statistiques : Application à la détection d'hypovigilance d'un conducteur automobile*, Thèse de doctorat, LAAS, Toulouse.

- Hines, J.W., D.W. Miller et B.K. Hajek (1995). Fault Detection and Isolation: A Hybrid Approach, *Proceedings of the 1995 American Nuclear Society Annual Meeting and Embedded Topical Meeting on Computer-Based Human Support Systems: Technology, Methods and Future*, Philadelphia, PA.
- Honda, K., T. Miyoshi et H. Ichihashi (1998). *Structural Learning of Recurrent RBF Networks with M-Apoptosis*, Proc. of IEEE International Joint Conference on Neural Networks, pp. 2390-2395, Anchorage/ USA.
- Hopfield, J.J. (1982). Neural networks and physical Sytems with emergent collective computational abilities, *Proceeding Nat. Acad. Sci. USA, Biophysics*, vol. 79, pp. 2554-2558.
- How, W. Y., M. Khalid, and S. Zain (1999). Transformer Fault Diagnosis Using Fuzzy Logic Interpretations. Conference Proceeding of Instrument Asia, Technical Symposium, Singapore.
- How, W.Y., M. Khalid et S.A.F. Syed Zain (1999). Transformer Fault Diagnosis Using Fuzzy Logic Interpretations. *Conference Proceeding of Instrument Asia, Technical Symposium*, Singapour.
- How, W.Y., M. Khalid et S.A.F. Syed Zain (2000). Transformer Fault Diagnosis Using Fuzzy Logic Interpretations. *Instruments and Electronics Developments*, Vol. 21 Issue No. pp. 56.
- Howell, A.J., et H. Buxton H. (1997). Recognising Simple Behaviours using Time-Delay RBF Networks, *Neural Processing Letters* Vol. 5, pp.97-105.
- Hutchinson, J.M. (1994). *A Radial Basis Function Approach to Financial Time Series Analysis*, Thèse de doctorat, Massachusetts Institute of Technology (MIT).
- Hwang, Y.S. et Bang S.Y. (1997) An efficient method to construct a Radial Basis Function Neural Network classifier, *Neural Networks*, vol. 10, n° 08, pp. 1495-1503.
- Isermann, R., (1984). Process Fault Detection Based on Modeling and Estimation Methods – A Survey, *Automatica*, Vol. 20, N°4, pp. 387-404.
- Jacquemin, C. (1994). A Temporal Connectionnist Approach to Natural Language, *Sigart Bulletin*, Vol. 5, N°03.
- Jordan, M. (1986). Serial Order : A Parallel Distributed Processing Approach, *Technical Report ICS Report No. 8604*, Institute for Cognitive Science, University of California, San Diego.
- Katsillis, G., M.J. Chantler (1997). Can dependency-based diagnosis cope with simultaneous equations?, *Working notes of the 8th International Workshop on Principles of Diagnosis DX'97*, Mont-Saint-Michel, France.
- Keller, P., R.T. Kouzes et L.J. Kangas (1994), Three Neural Network Based Sensor System for Environemental Monitoring, *Proceedings IEEE Electro94 Conference*, Boston, MA, USA.
- Koch, C. (1999). *Biophysics of Computation: Information Processing in Single Neurons*, Oxford University Press.
- Kohonen, T. (1989). *Self-Organization and associative memory*, Heidelberg: Springer-Verlag, Berlin, 3rd edition.
- Koivo, H.N. (1994). Artificial neural networks in fault diagnosis and control, *Control in engineering practice*, vol.2, n°1, p. 89-101.

- Kruse, R., and A. Nürnberger (1998). Learning Methods for Fuzzy Systems, *Manual, Department of Computer Science*, Université de Magdeburg, Allemagne.
- Lamontagne, L. et G. Lapalme (2002). Raisonnement à Base de Cas Textuels – état de l’art et perspectives, *Revue de l’intelligence artificielle*, volume 3/2002.
- Larman, C. (2002). *Applying UML and patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Prentice Hall PTR.
- Lee, G. et J.-S. Wang (2000). Self-Adaptive Neuro-Fuzzy Systems : Structure and learning, *IEE/RSJ, International Conference on Intelligent Robots and Systems*.
- Lee, G. et J.-S. Wang (2001). Efficient Neuro-Fuzzy Control Systems for Autonomous Underwater Vehicle Control, *IEEE International Conference on Robotics and Automation*, Seoul, Corée.
- Lefebvre, D., (2000). *Contribution à la modélisation des systèmes dynamiques à événements discrets pour la commande et la surveillance*, Habilitation à Diriger des Recherches, Université de Franche Comté/ IUT Belfort - Montbéliard.
- Leger, J.-B. (2004). Introduction à la e-maintenance, rapport de formation du personnel de la société DCN (Division des Constructions Navales) dans le cadre du contrat DCN-Institut de Productique de Besançon.
- Legrand, J.F. et P. Garda (2002). Prédiction de Trafic GSM par Méthodes Connexionnistes, *XIèmes Journées Neurosciences et Science pour l’Ingénieur NSI’2002*.
- Lipp, H-P. et R. Günter (1994) A study of Fuzzy Petri nets concepts, *Proceedings European Congress on Fuzzy and Intelligent Technologies*, Aachen, Germany, pp. 1461-1468.
- Liu, Y.H. et X.J. Wang (2001). Spike-frequency adaptation of a generalized leaky integrate and fire model neuron, *J. Comp. Neurosciences*, N°10, pp. 25-45.
- Loiez, E., (1997). *Contribution au diagnostic de systèmes analogique*, Thèse de doctorat de l’Université des sciences et technologies de Lille.
- Looney, C. (1988). Fuzzy Petri nets for rule-based Decision making, *IEEE Trans. on Systems Man and Cybernetics*, vol. 18, N 1, pp 178-183.
- Looney, C., L. Liang (2002). Inference via Fuzzy Bielef Networks, *Proceeding of the ISCA International Conference*.
- Looney, C., L. Liang (2003), Inference via Fuzzy Bielef Petri Nets, *15th IEEE International Conference on Tools with AI (ICTAI' 03)*, Sacramento, California, USA.
- Lopes, N. et B. Ribeiro (1999). Part Quality Prediction in an Injection Moulding Process Using Neural Networks, *Proceedings of WMC, ISM*.
- Mak, M. W. (1995). A Learning Algorithm for Recurrent Radial Basis Function Networks, *Neural Processing Letters*, Vol. 2, No. 1, pp. 27-31.
- Mak, M.W. et S.Y. Kung (2000). Estimation of Elliptical Basis Function Parameters by the EM Algorithms with Application to Speaker Verification, *IEEE Trans. on Neural Networks*, vol. 11, n° 4, pp. 961-969.
- Mackey, M. et L. Glass (1997). Oscillations and Chaos in Physiological Control System, *Science*, pp. 197-287.

- Malek, M. (1996). *Un Modèle Hybride de Mémoire pour le Raisonnement à Partir de Cas*, Thèse de doctorat de l'Université Joseph Fourier Grenoble1.
- Mangeas, M. (1996). *Propriétés Statistiques des Modèles Paramétriques non Linéaires de Prévisions de Séries Temporelles, Application aux Réseaux de Neurones à Propagation Directe*, Thèse de Doctorat, SAMOS / Université Paris I.
- Masson, M.H., B. Dubuisson et C. Frélicot (1996). Conception d'un module de reconnaissance des formes floues pour le diagnostic, *Journal Européen des Systèmes Automatisés (RAIRO-APII-JESA)*, pp. 319-341.
- McCulloch, W.S. et W. Pitts (1949) A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133.
- Meador, J., A. Wu, H. Tseng et T.S. Lin (1991). Fast Diagnosis of Integrated Circuit Faults Using Feedforward Neural Network, *IEEE International Joint Conference on Neural Networks*, Seattle, USA.
- Mellouk, A. (1994). *Développement d'un système hybride neuro-prédictif : application à la reconnaissance de la parole continue*, Thèse de Doctorat, LIP6/ Université de Paris 6.
- Mellouli, N. et B. Bouchon-Meunier (2000a). Fuzzy Approaches to Abductive Inference, *Conference on Non-Monotonic reasoning*, Berckenridge, Colorado.
- Mellouli, N. et B. Bouchon-Meunier (2000b). Modélisation de l'inférence abductive par inversion du modus ponens généralisé, *Rencontre francophones LEA*, La Rochelle, Cépaudès Edition, pp 79-86.
- Mellouli, N. et B. Bouchon-Meunier (2003). Abductive reasoning and measures of similitude in the presence of fuzzy rules, *Fuzzy Sets and Systems* 137, pp.177-188.
- Micchelli, C.A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constructive Approximation*, N°2, pp. 11-22.
- Minca, E., **D. Racocceanu** et N. Zerhouni (2002a). Monitoring Systems Modelling and Analysis Using Fuzzy Petri Nets, *Studies in Informatics and Control*, Vol. 11, No. 4.
- Minca E., **D. Racocceanu**, N. Zerhouni et I. Brezeanu (2002b). Fuzzy Based Petri Nets for the Production Systems Diagnosis, *Proceedings International Conference on Information Systems, Analysis and Synthesis, Electronics, Informatics and Control Systems – Invited Session*, Orlando, Vol. XII..
- Minca E., **D. Racocceanu** et N. Zerhouni (2003a). Fuzzy logic Petri nets for monitoring systems modeling, *IEEE International Conference on Robotics and Automation*, Taiwan.
- Minca E., **Racocceanu D.**, Zerhouni N. (2003b). Approche de détection par réseau de Petri flou, *Conférence Pentom2003 Performance et Nouvelles Technologies en Maintenance*, 26-28 mars, Valenciennes.
- Minca, E., **D. Racocceanu**, N. Zerhouni (2003c). Approche Temporelle Floue pour la Surveillance, *Qualita2003, 5ème Congrès International Pluridisciplinaire Qualité et Sécurité de fonctionnement*, Nancy.
- Minca, E., (2004). *Contribution à la surveillance des systèmes de production à l'aide des réseaux de Petri flous : Application à la e-maintenance*, Thèse de doctorat, Université de Franche-Comté.

- Mirea, L. et T. Marcu (2002). System Identification Using Functional Link Neural Networks with Dynamic Structure, *15th IFAC World Congress*, Barcelone, Espagne.
- Miyoshi, T., H. Ichihashi, S. Okamoto et T. Hayakawa (1995). Learning Chaotic Dynamics in Recurrent RBF Network, *Proc. of IEEE ICNN'95*, pp. 588-593, Perth/ Western Australia.
- Moakes, P. A. and S. W. Beet. (1994). Non-linear speech analysis using recurrent radial basis function networks, In *Neural Networks for Signal Processing IV*, eds.: J. Vlontzos, J-N. Hwang and E. Wilson, pp 319-328. IEEE Press.
- Monnin, M., (2004). *Diagnostic et pronostic industriel en utilisant les techniques de l'intelligence artificielle. Application a la plate-forme de e-maintenance du LAB*, DEA IAP de l'Université de Franche-Comté, Besançon.
- Monnin, M., **D. Racocceanu** et N. Zerhouni (2004). Overview on diagnosis methods using artificial intelligence, *IEEE Conference on Robotics, Automation and Mechatronics - IEEE RAM 2004*, Singapour.
- Monostori, L. (2002). AI and Machine Learning Techniques for Managing Complexity, Changes and uncertainties in Manufacturing, *15^{eme} IFAC World Congress on Automatic Control*, Barcelone, Espagne.
- Moody, J. et J. Darken J. (1989). Fast Learning in networks of locally tuned processing units, *Neural Computation*, vol. 1, pp. 281-194.
- Morello-Chebel, B., I. Rasovska et N. Zerhouni (2005). Knowledge capitalization in system of equipment diagnosis and repair help. *IJCAI'2005 Workshop on Knowledge Management and Organizational Memories*, Edinburgh, Scotland.
- Mozer, M. C. (1993). Neural network architectures for temporal pattern processing, In A. S. Weigend & N. A. Gershenfeld (Eds.), *Time series prediction: Forecasting the future and understanding the past* (pp. 243-264). Redwood City, CA: Sante Fe Institute Studies in the Sciences of Complexity, Proceedings Volume XVII, Addison-Wesley Publishing.
- Murata, T. et D. Zhang (1988). A Predicate Transition net model for parallel interpretation of logic programs, *IEEE Transaction on Software Engineering*, 14(4), pp. 481-497.
- Mustawi, M.T., W. Ahmed, K.H. Chan, K.B. Faris et D.M. Hummels (1992) On the training of Radial Basis Function Classifiers, *Neural Networks*, vol. 5, pp. 595-603.
- Nauck, D. et R. Kurse (1995). NEFCLASS A Neuro-Fuzzy Approach for the Classification of data, *Proc. Of the 1995 ACM Symposium on Applied Computing*, Nashville, ACM Press.
- Nauck, D., U. Nauck et R. Kruse (1996). Generating Classification Rules with the Neuro-Fuzzy System NEFCLASS, *Biennial Conf. Of the North American Fuzzy Information Processing Society (NAFIPS'96)*, Berkeley, 1996.
- Nauck, D. et R. Kruse (1997). What are Neuro-Fuzzy Classifiers ?, *Seventh International Fuzzy Systems Association World Congress IFSA '97*, Vol. IV, pp. 228-233, Academie de Prague.
- Nauck, D. et R. Kurse (1998). A Neuro-Fuzzy Approach to Obtain Interpretable Systems for function Approximation, *IEEE International Conference on Fuzzy Systems*.
- Nikolaos, L. (1991). *Arbres de défaillance*, Ed. Hermes.

- OMG (2003). Unified Modeling Language Specification. Version 1.5.
- Ould Abdeslam, D. (2002), *Utilisation des Réseaux Neuro-Flous pour le Pronostic et le Diagnostic. Application à la classification*, Mémoire DEA, Université de Franche-Comté, Besançon.
- Ould yahia, N. S., (2001). *Diagnostic des systèmes de production en utilisant les réseaux de neurones. Introduction au neuro-flou*, Mémoire DEA, Université de Franche-Comté, Besançon.
- Pal, N. et J. Bezdek (1995). On cluster validity for fuzzy C-means model, *IEEE Trans. Fuzzy syst.*, Vol 3.
- Palade, V., R.J. Patton, F.J. Uppall, J. Quevedo et S. Daley (2002). Fault diagnosis of an industrial gas turbine using neuro-fuzzy methods, *World IFAC Congress*.
- Palluat, N. (2002). *Méthodologies Neuronales de Diagnostic des Dégradations*, Mémoire de DEA, Laboratoire d'Automatique de Besançon, France.
- Palluat, N., **D. Racocceanu** et N. Zerhouni (2004a). Diagnosis aid system using a neuro-fuzzy approach. In: *Advances in Maintenance and Modeling, Simulation and Intelligent Monitoring of Degradation*, IMS'2004, Arles, France.
- N. Palluat et **D. Racocceanu** (2004b). Conception et prototypage d'un système de surveillance dynamique embarqué et d'aide au diagnostic d'un centre d'usinage à grande vitesse, *Rapport ANVAR – aide aux jeunes pour l'innovation*, ANVAR Besançon.
- Palluat, N., **D. Racocceanu** et N. Zerhouni (2005a). Utilisation des réseaux de neurones temporels pour le pronostic et la surveillance dynamique. Etude comparative de trois réseaux de neurones récurrents, *Revue des sciences et technologies de l'information - RSTI, série Revue d'intelligence artificielle - RIA*, n°6.
- N. Palluat, **D. Racocceanu** et N. Zerhouni (2005b). An UML modelling of a neuro-fuzzy monitoring system, *16th IFAC World Congress*, Prague, République Tchèque.
- Pasquier, L. (2002). *Modélisation de raisonnement tenus en contexte. Application à la gestion d'incidents sur une ligne de métro*, Thèse de l'université de Paris 6.
- Patton, R., P. Frank et R. Clark (1989). *Fault Diagnosis in Dynamic Systems : Theory and Application*, International Series in Systems and Control Engineering, Prentice Hall International, Londres.
- Pearlmutter, B.A. (1990). Dynamic Recurrent Neural Networks, *CMU-CS-90,196*, Carnegie Mellon University, School of Computer Science.
- Pedriycz, W. et F. Gomide (1994). A generalized Fuzzy Petri net model, *IEEE Transaction on Fuzzy Systems*, 2(4), pp. 295-301.
- Pencolé, Y., (2002). *Diagnostic décentralisé de systèmes à événements discrets : application aux réseaux de télécommunications*, Thèse de doctorat, Université de Rennes 1.
- Peng, Y. et J.A. Reggia (1990). *Abductive Inference Models For Diagnostic Problem-Solving*, Springer-Verlag, New York.
- Petsche, T.A., A. Marcontonio, C. Darken, S.J. Hanson, G. Mkuh et I. Santoso (1996). *A Neural Network autoassociator for induction motor failure prediction*, Cambridge: MIT Press, Edition D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, *Advances in Neural Information Processing Systems* 8, pp. 924-930.

- Piechowiak, S. (2004). Intelligence artificielle et diagnostic, *Techniques de l'Ingénieur, S7 217, traité Informatique Industrielle*.
- Pierce, C. S. (1867). *Collected Papers*, The Belknap Press of Harvard University Press.
- Pineda, F.J. (1987). Generalization of Backpropagation to recurrent Neural Networks, *Physical Review Letters*, N° 59, pp. 2229-2232.
- Ploix, J.L., et G. Dreyfus (1997). Early fault detection in a distillation column: an industrial application of knowledge-based neural modelling, *Neural Networks: Best Practice in Europe*, B. Kappen, S. Gielen, eds, pp. 21-31, World Scientific.
- Ploix, S. et S. Gentil (2000). Causal strategy for set-membership fault diagnosis, *Conference on Decision and Control*, Sidney, Australie.
- Ploix, S. et C. Follot (2001). Fault Diagnosis reasoning for set-membership approaches and application, *IEEE International Symposium on Intelligent Control (ISIC)*, Mexico City, Mexique.
- Poddar, P. et K.P. Unnikrishnan (1991a) Nonlinear Prediction of Speech Signals Using Memory Neuron Networks, *Neural Networks for Signal Processing I*, B. H. Juang, S. Y. Kung and C. A. Kamm, Eds. IEEE Press.
- Poddar, P. et K.P. Unnikrishnan (1991b) Memory Neuron Networks : A Prolegomenon, *General Motors Research Laboratories Report GMR-7493*.
- Poulard, H., (1996). *Statistiques et réseaux de neurones pour un système de diagnostic", Application au diagnostic de pannes automobiles*, Thèse de Doctorat, LAAS, Toulouse, France.
- Pross, S., (2001). *Analyse des Défaillances des systèmes industriels : Application au Système de transfert SORMEL du Laboratoire d'Automatique de Besançon*, Mémoire de DEA, Laboratoire d'Automatique de Besançon, Université de Franche-Comté.
- Proteus - ITEA European Project (2004) - WP2 TEAM : L. Déchamp, A. Dutech, T. Montroig, X. Qian, **D. Racoceanu**, I. Rasovska, P. Brézillon, F. Charpillet, J-Y. Jaffray, N. Moine, B. Morello, S. Müller, G. Nguengang, N. Palluat, L. Pelissier - On the use of artificial intelligence for prognosis and diagnosis in the PROTEUS E-maintenance platform, *IEEE Mechatronics & Robotics*, E-Maintenance Special Session, Aachen, Allemagne.
- Racoceanu, D.**, N. Zerhouni N. et N. Addouche (2002). Modular modeling and analysis of a distributed production system with distant specialized maintenance, *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 4046-4052, Washington, États-Unis.
- Racoceanu, D.** et D., Ould Abdeslam (2003). Réseaux Neuro-Flous pour la Surveillance des Systèmes, *4^{ème} Conférence Internationale sur l'Automatisation Industrielle*, Montréal, Canada.
- Racoceanu D.** (2004). Techniques de l'Intelligence artificielle pour la surveillance industrielle. Techniques et Technologies de la e-Maintenance, section : Techniques pour les processus « Métier » de la e-Maintenance, rapport de formation du personnel de la société DCN (Division des Constructions Navales) dans le cadre du contrat DCN – MACOD/GdR MACS, Besançon.

- Rahamani, K., P. Arabshahi, T.-Y. Yan, T. Pham et S. G. Finley (1998). An Intelligent Fault Detection and Isolation Architecture For Antenna Arrays, *TDA Progress Report 42-132*.
- Rasovska, I., B. Morello-Chebel et N. Zerhouni (2005). Un système de raisonnement à partir de cas dans une plate-forme de e-maintenance. In S. Després, Eds., *actes du 13ieme atelier de raisonnement à partir de cas*, Nice, France.
- Rengaswamy, R. et V. Venkatasubramanian (1995). A Syntactic Pattern Recognition Approach for Process Monitoring and Fault Diagnosis, *Engineering Applications of Artificial Intelligence Journal*, 8(1), p. 35-51.
- Richet, D., M. Gabriel, D. Malon et G. Blaison (1996). *Maintenance basée sur la fiabilité*, Editions Masson.
- Robinson, A., et F. Fallside (1987). The Utility Driven Dynamic Error Propagation Network, *Combridge Univ. Eng. Dept., Tech. Rep. CUED/F-IN-FENG/TR.1*.
- Rocques, P. (2003), UML Modéliser un site e-commerce, *les Cahiers du Programmeur*, 152 p., Ed. Eyrolles.
- Rosenblatt, F. (1958). The Perceptron: a probabilistic model for information storage in the brain, *Psychological Review*, Vol. 65, p. 386-408.
- Rumbaugh, J., I. Jacobson et G. Booch (1999). *The unified modeling language reference manual*, Reading, Addison Wesley.
- Rummelhart, D.E, G.E. Hinton et R.J. Williams (1986). Learning Internal Representation by Error Propagation, in *Parallel Distributed Processing Explorations in the Microstructure of Cognition*, vol. 1, The MIT Press/Bradford Books, D.E. Rumelhart and J.L.McClelland, pp. 318-362.
- Rynkiewi, J. (2000). *Modèles Hybrides Intégrant des Réseaux de Neurones Artificiels à des Modèles de Chaînes de Markov Cachées : Application à la Prédiction de Séries Temporelles*, Thèse de Doctorat, SAMOS / Université de Paris I.
- Sakai, M., N. Homma et K. Abe (2002). A Statistical Approximation Learning Method for Simultaneous Recurrent Networks, *15th IFAC World Congress*, Barcelone, Espagne.
- Sampath, M., R. Sengupta, S. Lafortune, K. Sinnamohideen et D.C. Teneketzis (1996). Failure Diagnosis Using Discrete Event Models, *IEEE Transactions On Control Systems Technology*, Vol. 4, No. 2.
- Scarpelli, H. et F. Gomide (1994). High – level Fuzzy Petri nets and backward reasoning, *Proceedings IPMU*, Paris, France, pp. 1275-1280.
- Scherer, M. (2003). Vers l'e-maintenance, *Lettre hebdomadaire Industries et Technologies*, N° 28.
- Schmidhuber, J. (1992). A Fixed Storage $O(N^3)$ Time Complexity Learning Algorithm for Fully Recurrent Continually Running Networks, *Neural Computation*, Vol. 4, N°2, pp. 243-248.
- Sejnowski, T.J. et C.R. Rosenberg (1986). NetTalk: a parallel network that learns to read aloud, *Electrical engineering and Computer science technical report*, the Johns Hopkins University.
- Shing, J., et R. Jang (1993). *ANFIS : Adaptive-Network-Based Fuzzy Inference System*, *IEEE Transaction on systems, Man, and Cybernetics*, Vol. 23, NO. 3.

- Sibertin-Banc, C. (1985). High-level Petri nets with data structures, European Workshop on Application and Theory of Petri nets, Helsinki, Finland, pp. 1275-1280.
- Simpson, P.K. (1992). Fuzzy min-max neural networks – Part I : Classification, *IEEE Transaction on Neural Networks*, Vol.3, N°5, pp. 776-786.
- Simpson, P.K. (1993). Fuzzy min-max neural networks – Part II : Clustering, *IEEE Transaction on Fuzzy Systems*, Vol.1, N°1, pp. 32-45.
- Sinha, M., M.M. Gupta et P.N. Nikiforuk (2002). Hybrid Neural Models for Time Series Forecasting, *15th IFAC World Congress*, Barcelone, Espagne.
- Smyth, P. (1994) Detecting novel fault conditions with hidden Markov models and neural networks, *Pattern Recognition in Practice IV*, pp. 525-536.
- Sorel, M., et Sima J. (2000). Robust implementation of finite automata by recurrent RBF networks», *Proceedings of the SOFSEM Seminar on Current Trends in Theory and Practice of Informatics*, Milovy, Czech Republic, 431-439, Berlin: Springer-Verlag, LNCS 1963.
- Sun, G.Z., H.H. Chen et Y.C.Lee (1990). Green's Function Method for Fast on-line Learning Algorithm of Recurrent Neural Networks, *Advances Neural Information Processing System*, Vol. 4, pp. 333-340.
- Terstyanszky, G. et L. Kovacs (2002). Improving Fault Diagnosis Using Proximity and Homogeneity Measure, *15^{ème} IFAC World Congress on Automatic Control*, Barcelone, Espagne.
- Toguyeni, A.K.A., (1992). *Surveillance et diagnostic en ligne dans les ateliers flexibles de l'industrie manufacturière*, Thèse de doctorat, Université de Lille.
- Toomarian, N. et J., Barhen (1991). Adjoint-Functions and Temporal Learning Algorithms in Neural Networks, *Advances in Neural Information Processing Systems*, Vol. 3, pp. 113-120.
- Tromp, L., (2000). *Surveillance et diagnostic de systèmes industriels complexes: une approche hybride numérique/symbolique*, Thèse de doctorat, Université de Rennes 1.
- Tsang, E.C.C., J.W.T. Lee et D.S. Yeung (2002). Tuning Certainty Factor and Local Weight of Fuzzy Production Rules Using Fuzzy Neural Network, *IEEE Transactions of Systems, Man and Cybernetics*, Part B, Vol.32, No.1, pp.91-98.
- Tsoi, C.T., A.D. Back (1994). Locally Recurrent Globally Feedforward Networks : A Critical Review of Architectures, *IEEE Transaction on Neural Networks* Vol.05, pp. 229-239.
- Tyan, C. Y., P.P. Wang et D. Bahler (1994). Neural Fault Diagnosis and Fuzzy Fault Control for a Complex Linear Dynamic System, *Advances in Fuzzy Theory and Technology, Volume II*, ISBN: 0-9643456-1-7, pp. 357-375.
- Uppal, F.J., R.J. Patton et V. Palade (2002). Neuro-fuzzy based Fault Diagnosis applied to an Electro-Pneumatic Valve, *15^{ème} IFAC World Congress on Automatic Control*, Barcelone, Espagne.
- Urbani, D. (1995). *Méthodes statistiques pour la sélection d'architectures neuronales : application à la modélisation de processus dynamiques*, Thèse de Doctorat, ESPCI/ Université Pierre et Marie Curie - Paris VI.

- Valette, R. et H. Atabakhche (1987). Petri nets for sequence constraint propagation in knowledge based approaches, *Concurrency and Petri Nets*, pp. 555-569, Springer Verlag.
- Valette, R., J. Cardoso et D. Dubois (1989). Monitoring manufacturing System by means of Petri Nets with Imprecise Markings, *IEEE International Symposium on Intelligent Control*, Albany N.Y., USA.
- Valette, R., L. A. Künzle (1994). Réseaux de Petri pour la détection et le diagnostic, *Journées Nationales : Sécurité, Surveillance, Supervision, GdR Automatique*, Paris, France.
- Vemuri, A., M. Polycarpou et S. Diakourtis (1998). Neural Network Based Fault Detection and Accommodation in Robotic Manipulators, *IEEE Transactions on Robotics and Automation*, vol. 14, n° 2, pp. 342-348.
- Vemuri, A. et M. Polycarpou (1997). Neural Network Based Robust Fault Diagnosis in Robotic Systems, *IEEE Transactions on Neural Networks*, vol. 8, n° 6, pp. 1410-1420.
- Villemeur, A., (1988). *Sûreté de fonctionnement des systèmes industriels*, Edition Eyrolles, Collection DER-EDF, Volume 67.
- Waibel, A., T. Hanazawa, G. Hinton, K. Shikano et K. Lang (1989). Phoneme recognition using time delay neural network, *IEEE Trans. in Acoustics, Speech and Signal Processing*, vol. 37, n° 3, p. 328-339.
- Wang, J.-S. et G. Lee (2002). Self-Adaptative Recurrent Neuro-Fuzzy Control for an Autonomous Underwater Vehicle, *IEEE International Conference on Robotics & Automation*, Washington, DC.
- Warwick, K., G.W. Irwin, K.J. Hunt (1992). *Neural Networks for Control and Systems*, *IEE Control Engineering Series* 46, Peter Peregrinus Ltd., London, United Kingdom.
- Washio, T. et H. Hotoda (1998). Discovering admissible simultaneous equations of large scale systems, *15th National Conference on Artificial Intelligence AAAI-98*, Madison, WI, Etats-Unis, pp. 189-196.
- Weber, P., (1999). *Diagnostic de procédés par l'analyse des estimations paramétriques de modèles de représentation à temps discret*, Thèse de Doctorat, INPG, Grenoble.
- Werbos, P. (1990). Backpropagation Trough time: What it does and how to do it, *Proceedings IEEE*, Vol. 78, 1990.
- Williams, R.J. et D. Zipser (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, vol.1, pp. 270-280.
- Willsky, A.S., (1976). A Survey of Design Methods for Failure Detection in Dynamic Systems, *Automatica*, Vol. 12, pp. 601-611.
- Wu, A. et J. Meador (1994). A Measurement Selection for Parametric IC Fault Diagnosis, *Journal of Electronic Testing: Theory and Applications*, Kluwer Academic Publishers, Vol. 5, No. 1, pp. 9- 18.
- Xu, J., (2004). *Utilisation des réseaux de neurones dynamiques pour la e-maintenance. Instrumentation d'une maquette de surveillance temps réel*, DEA IAP de l'Université de Franche-Comté, Besançon.

- Yu, W.S., et Wang G.C. (2001). Adaptive Control Design Using Delayed Dynamical Neural Networks for a Class of Nonlinear Systems, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, Seoul, Corée.
- Zamai, E., (1997). Architecture de surveillance commande pour les systèmes a événements discrets complexes, Thèse de doctorat, Informatique Industrielle, Université Paul Sabatier de Toulouse, France.
- Zemouri, R., **D. Racocceanu** et N. Zerhouni (2001). The RRBF : Dynamic representation of time in radial basis function network, *Proc. of the 8th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA'2001*, vol. 2, pp.737-740, Antibes, Juan les Pins, France.
- Zemouri, R., **D. Racocceanu** et N. Zerhouni (2002a). From the spherical to an elliptic form of the dynamic RBF neural network influence field, *World Congress on Computational Intelligence, International Joint Conference on Neural Networks (IJCNN)*, USA.
- Zemouri, R., **D. Racocceanu** et N. Zerhouni (2002b). Réseaux de neurones Récurents à Fonction de base Radiales : RRFR / Application au pronostic, *Revue d'Intelligence Artificielle, RSTI série RIA*, Vol. 16, N°03.
- Zemouri, R., **D. Racocceanu** et N. Zerhouni (2002c). Application of the dynamique RBF Network in a monitoring problem of the production systems, *15^{ème} IFAC World Congress on Automatic Control*, Barcelone, Espagne.
- Zemouri, R., (2003). *Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance*, Thèse de Doctorat, Université de Franche Comté, Besançon.
- Zemouri, R., **D. Racocceanu** and N. Zerhouni (2003a) Recurrent Radial Basis Function neural network for time-series prediction, *Engineering Applications of Artificial Intelligence, The International Journal of Intelligent Real-Time Automation*, journal IFAC - the International Federation of Automatic Control, ed. Elsevier Science, vol. 16, Issue 5-6, pp.453-463.
- Zemouri, R., **D. Racocceanu** et N. Zerhouni (2003b). Réseaux de neurones récurrents à Fonctions de base Radiales. Application à la surveillance dynamique, *JESA – Journal Européen des Systèmes Automatisés*, vol. 37 – n° 1, pp. 49-81.
- Zhang, Q., (1999). *Identification et Surveillance de systèmes Dynamiques*, Habilitation à diriger les recherches, Université de Rennes1, Institut de Formation Supérieure en Informatique et en Communication.
- Zisman, M. D. (1978). Use of production systems for modeling asynchronous processes, *Pattern Directed Inference systems*, Waterman D. A., Hayes-Roth F., Academic Press, pp. 53-68.
- Zwingelstein, G., (1995). *Diagnostic des défaillances : Théorie et pratique pour les systèmes industriels*, Editions Hermès.

Index

A

ADD 5, 60, 164, 167, 172, 173, 181, 187, 198, 200, 201, 205, 206, 208, 209, 210, 215
AMDEC.....5, 60, 73, 80, 187, 198, 200, 201, 205, 206, 208, 210, 211, 215

C

chemin critique..... 168, 178, 181, 183
chien de garde 162

D

defuzzification 183
description floue 167
détection .. 160, 161, 162, 164, 166, 173, 179, 226, 232, 238, 241
diagnostic..... 160, 179, 226, 227, 235, 238, 241

E

e-Maintenance 28, 39, 51, 235, 241

F

fonction d'appartenance 5, 161, 183
fuzzyfication..... 167

G

GMAO 5, 187, 198, 200, 201, 206, 212, 215

I

implication floue 166
IMS 25, 49
Intelligence Artificielle.. 1, 11, 33, 36, 46, 50, 57, 63, 83
intersection 165, 179, 183

L

logique floue.....47, 157, 158, 183, 214, 225, 241

M

maintenance47, 170, 171, 177, 178, 181, 182, 184, 215, 235, 241

marquage flou 226, 241
mécatronique 19, 83
MIMOSA 5, 49
modus ponens généralisé 167

P

produit cartésien 183
pronostic 11, 16, 17, 23, 49, 157, 159, 233

R

RdPFM 5, 177, 181, 183, 184, 215
RdPFS.....5, 12, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 179, 180, 181, 183, 184, 214, 215
RdPTO ..5, 160, 161, 162, 163, 164, 166, 167, 171, 179, 184, 215
rejet d'ambiguïté..... 66
reprise ...5, 161, 167, 168, 170, 171, 173, 176, 177, 181, 182, 183, 184, 215, 225, 241
Réseau de référence..... 160
RRFR.....5, 12, 26, 46, 86, 105, 106, 107, 117, 119, 121, 122, 123, 124, 125, 127, 128, 129, 130, 131, 134, 135, 136, 137, 138, 140, 141, 142, 143, 144, 145, 146, 147, 149, 150, 152, 153, 154, 155, 156, 185, 205, 206, 209, 214, 239

S

SCADA 5, 200, 201, 215
supervision 178, 182, 184, 215
surveillance.....5, 47, 157, 160, 161, 163, 164, 166, 171, 173, 177, 179, 181, 182, 184, 214, 215, 226, 239, 241
système expert flou 157, 181

T

temps réel 226, 241
transition du chien de garde 162

U

union 165, 179

V

variable linguistique 162