

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار- عنابة

Thèse

**Pour l'obtention d'un diplôme de Doctorat en science de
l'université Badj Mokhtar en électronique option Automatique**

Intitulé :

**Génération des résidus par les outils de l'intelligence
artificielle pour le diagnostic des systèmes
complexes.**

**Présentée par :
KOURD Yahia**

Composition du jury:

Président:	Bensaker Bachir	Professeur : U. B. M. Annaba
Rapporteurs:	Guersi Noureddine	M.C : U. B. M. Annaba
	Lefebvre Dimitri	Professeur : Université du Havre (France)
Examineurs:	Khezzar Abdelmalek	Professeur : Université Constantine
	Lebaroud Abdesselem	M.C : Université Skikda

Année : 2011/2012

Faculté des Sciences de l'Ingénieur - Département d'Electronique
BP. 12, Sidi Amar 23000 Annaba

Remerciements

Je remercie notre Dieu, le tout puissant, de nous avoir donné, le courage, la patience, la volonté et la force nécessaire, pour affronter toutes les difficultés et les obstacles, qui se sont hissés au travers de notre chemin, durant toutes nos années d'études de la thèse.

*Je tiens à exprimer mes vifs et sincères remerciements à Monsieur **Lefebvre Dimitri**, Professeur à l'université du Havre, directeurs du laboratoire GREAH de m'avoir accueilli au sein de son équipe et de m'avoir encadré et encouragé tout au long de ces travaux. Aussi par leurs grandes compétences scientifiques et leurs remarquables qualités humaines, ils ont su rendre ces années de recherche chaleureuses et fructueuses.*

*J'adresse mes plus sincères remerciements à Monsieur **Guersi Noureddine**, Professeur à l'université d'Annaba, pour la qualité de ses interventions et de m'avoir fait confiance et pour ses conseils et orientations. Je tiens, également, à lui témoigner, ma profonde gratitude pour son engouement pour les questions scientifiques diverses relatives à l'Automatique.*

Je remercie chaleureusement le président et les membres du jury pour l'intérêt qu'ils ont bien voulu accorder à ce travail en acceptant de l'examiner.

Je remercie aussi ma femme pour son encouragement, son sacrifice et sa patience pour finaliser ce travail.

Je remercie mes amis et collègues de laboratoire de l'équipe GREAH de l'université du Havre, pour l'ambiance conviviale qu'ils ont contribué à entretenir, les bons moments passés en leur compagnie ainsi que leur sympathie.

KOURD Yahia

Sommaire

Références personnelles	X
Table des Figure	XI
Liste des tableaux	XII
Notations.....	XIII
Abréviation.....	XIV
Résumé.....	1
Introduction générale.....	4

Chapitre I: **Etat de l'art.**

I.1 Introduction	11
I.2 Terminologie et définition	11
I.3 Les méthodes de diagnostic.....	17
I.3.1 Les méthodes à base de modèle	19
I.3.2 Les méthodes à base de connaissances et de données	21
I.4 Les étapes de diagnostic	25
I.4.1 Génération de résidus :	25
I.4.2 La détection de défaut	26
I.4.3 La localisation de défaut	26
I.4.4 L'identification de défaut	28
I.5 La structuration des résidus.....	29
I.6 Conclusion.....	30

Chapitre II: **Aperçu et application des Réseaux de Neurones, Logique Floue et Neuro-Flou dans le Diagnostic.**

II.1 Réseau de neurones.....	33
II.1.1 Introduction	33
II.1.2 Neurone biologique et neurone formel	34
II.1.3 Architecture des réseaux de neurones	37
II.1.4 Apprentissage des réseaux de neurones	40
II.1.5 Algorithme d'apprentissage	41
II.1.6 La classification par les réseaux de neurone.....	46
II.1.7 Quelques points essentiels à retenir.....	46
II.2 La logique floue	47
II.2.1 Introduction	47
II.2.2 L'ensemble flou	48
II.2.3 Système flou.....	48
II.2.4 Application de la logique floue	50
II.2.5 Logique floue et système expert.....	51
II.3 Les Réseaux Neuro-Flou.....	51
II.4 Application des RN, LF, et NF dans le diagnostic	54
II.5 Conclusion.....	57

Chapitre III: **Méthode intelligentes pour la détection des défauts.**

III.1 Introduction	59
III.2 Conception d'observateurs intelligents	60
III.2.1 Modèle neuronal des comportements sains	60
III.2.2 Modèle neuro-flou des comportements sains.....	61

III.3 Détection par analyse des résidus	65
III.4 Application de la détection de défauts au Benchmark DAMADICS	66
III.4.1 Description du système.....	66
III.4.2 Les défauts et leurs caractéristiques.....	69
III.4.3 Modèle Neuronale de comportement normal	71
III.4.4 Modèle Neuro-flou de comportement normal	76
III.4.5 Génération des résidus et détection de défaut	82
III.4.6 Comparaison des résultats et discussion	89
III.5 Conclusion.....	90

Chapitre IV: **Diagnostic rapide par synthèse de modèles de dysfonctionnement.**

IV.1 Introduction	92
IV.2 Synthèse de modèles de dysfonctionnement	94
IV.3 Diagnostic par évaluation des résidus.....	95
IV.4 Diagnostic par calcul de la probabilité des modèles	96
IV.5 Structuration des résidus	98
IV.5.1 Matrice des signatures.....	98
IV.5.2 Réduction de l'ensemble des résidus	100
IV.6 Diagnostic des défauts de l'actionneur DAMADICS	101
IV.6.1 Diagnostic par analyse des modèles de référence	101
IV.6.2 Diagnostic par analyse des modèles de dysfonctionnement.....	110
IV.6.3 Structuration et discussion	134
IV.6.4 Résultat de la réduction de l'ensemble des résidus	135
IV.7 Conclusion	137
Conclusion générale.....	139
Bibliographie	140

Références personnelles

Revues avec comité de lecture:

Kour, Y., D. Lefebvre and N. Guersi (2011) "Early FDI Based on Residuals Design According to the Analysis of Models of Faults: Application to DAMADICS". *Advances in Artificial Neural Systems*. doi: 10, 1155/2011/453169.

The articles of Advances in Artificial Neural Systems are indexed/reviewed in the following databases/resources:

- Academic OneFile
- ACM Guide to Computing Literature
- Airiti Library
- ANTE: Abstracts in New Technologies and Engineering
- Biological Sciences
- Chemical Abstracts
- Computers and Applied Sciences Complete
- CSA Illustrata - Natural Sciences
- CSA Illustrata - Technology
- CSA Neurosciences Abstracts
- CSA Technology Research Database
- Directory of Open Access Journals (DOAJ)
- EBSCO Discovery Service
- EBSCO host Connection
- Expanded Academic Index
- Google Scholar
- InfoTrac Custom journals
- INSPEC
- J-Gate Portal
- ProQuest Computer Science Journals
- ProQuest SciTech Journals
- ProQuest Technology Journals
- The Informatics Portal io-port.net

03 articles soumis au journal:

- Kour, Y., D. Lefebvre and N. Guersi "FDI according to fault probability evaluation: Application to DAMADICS" articles soumis depuis 7/11/2011 au IJCAS "International Journal of Control, Automation, and Systems".
- Kour, Y., D. Lefebvre and N. Guersi "Faults Diagnosis Based on Faulty Models Design: Application to DAMADICS" articles soumis depuis 20/01/2012 au journal Synthèse de l'université d'Annaba.
- Kour, Y., D. Lefebvre and N. Guersi " Diagnosis Based on Neural Networks and Decision Trees: Application to DAMADICS" articles soumis depuis 13/03/2012 au journal IJICIC.

Congrès internationaux avec comité de lecture et actes:

- 1- Kour Y. «Proposition d'une solution multi-agents pour la commande et la coopération des robots mobiles», JNAM'07, 29 mai 2007. USTHB, Alger. Algeria.
- 2- Kour Y. Guersi N. «Diagnosis of the faults by neurons networks, application on DAMADICS Benchmark», MS'07, July, 02-04, 2007. USTHB, Alger. Algeria.
- 3- Kour Y. Guersi N. Lefebvre D. "A two stages diagnosis method with neuronal networks" proceeding ICEETD'08, November 8-10, 2008 Hammamet, Tunisie.
- 4- Kour Y. Guersi N. Lefebvre D. «Neurons Networks Approach for Fault Diagnosis», proceeding C2MSI'09, April 28-29, 2009 Souk-Ahras, Algeria.

- 5- Kourd Y. Lefebvre D. Guersi N. « Approches Neuronale et Neuro-Floue pour le diagnostic: application au DAMADICS » IEEE-CIFA 2010, 2-4 Juin 2010. Nancy France.
- 6- Kourd Y. Lefebvre D. Guersi N. « FDI WITH NEURAL AND NEURO-FUZZY PPROACHES: APPLICATION TO DAMADICS » ICINCO 2010, 15-18 Juin 2010. Madeira, Portugal
- 7- Kourd Y. Lefebvre D. Guersi N. "Neuro-fuzzy approach for default Diagnosis: application to the DAMADICS" proceeding IEEE-Dest2010, Dubai, EAU, 23-25 April 2010.
- 8--Kourd Y. Lefebvre D. Guersi N. "Faults Diagnosis Based on Faulty Models Design: Application to DAMADICS", proceeding MIC-SMD 02-05 February 2012. Istanbul, Turkey

02 articles acceptés:

- Kourd Y. Lefebvre D. Guersi N." FDI with neural network models of faulty behaviours and fault probability evaluation: Application to DAMADICS" SAFEPROCSS-2012: Fault Detection, Supervision and Safety for Technical Processes. Mexico. 27-28 August, 2012.
- Kourd Y. Lefebvre D. Guersi N. "New Technique for Online Faults Diagnosis Based on Faulty Models Design: Application to DAMADICS Actuator", IEEE- MED2012 Barcelona, July 3-6, 2012

Table des Figures

Figure 1: Procédure générale pour la Détection et le diagnostic	7
Figure I.1: Positionnement de la supervision	12
Figure I.2 : Les différents emplacements de Défauts dans le système	13
Figure I.3 : La redondance physique triple de l'information	14
Figure I.4 : Principe des méthodes basées sur la redondance analytique	15
Figure I.5 : Evolutions temporelles des différents types de défauts	17
Figure I.6 : Modules d'une procédure de surveillance	18
Figure I.7 : Les différentes méthodes de diagnostic	19
Figure I.8 : Procédure FDI	19
Figure I.9 : Diagnostic par estimation de paramètres	20
Figure I.10 : Principe général d'une observation d'état	21
Figure I.11 : Diagnostic par espace de parité	21
Figure I.12: Diagnostic basée sur la technique de la reconnaissance des formes	22
Figure I.13: Architecture générale d'un système expert de diagnostic	22
Figure I.14 : Les épisodes et la classification	24
Figure I.15 :L'étape de génération de résidus	25
Figure I.16: Détection et identification de défaut	28
Figure II.1 : Neurone Biologique	34
Figure II.2 : Une synapse	35
Figure II.3 : Mac Culloch et Pitts neurone	35
Figure II.4 : Schéma d'un neurone formel	36
Figure II.5 : Différentes formes des fonctions d'activation	36
Figure II.6 : Architectures des réseaux de neurones	37
Figure II.7 : Les Perceptrons mono couche en générale	38
Figure II.8: Architecture d'un réseau multicouche	39
Figure II.9: Différence entre l'apprentissage supervise et non-supervise du réseau de neurone	40
Figure II.9: Illustration du principe de la descente de gradient, dans le cas d'une fonction à une variable	42
Figure II.10 : Direction du gradient pour différents algorithmes	46
Figure II.11 : Fonctions d'appartenances	48
Figure II.12 : Système Flou	48
Figure II.13 : Exemple d'implication (méthode de Mamdani)	49
Figure II.14 : Système Neuro-Flou	52
Figure II.15: Architecture d'un réseau Neuro-Flou hybride	53
Figure II.16: Différents architectures d'un réseau Neuro-Flou Série/Parallèle	53
Figure II.17: Architecture d'un réseau Neuro-Flou dynamique	54
Figure II.18: Génération de résidu par le modèle neuronal	55
Figure II.19: Classification par LF	56
Figure II.20: Architecture d'un classifieur NF	57
Figure III.1: Conception du modèle a base de données	61
Figure III.2 :L'Architecture de l'ANFIS	62
Figure III.3: Exemple d'ANFIS à 2 entées avec 9 règles.	64
Figure III.4: Détection d'un défaut par analyse du résidu.	66
Figure III.5: l'usine de sucre à Lublin, Pologne.	67
Figure III.6: Emplacement de l'actionneur dans l'usine de sucre	67
Figure III.7: Actionneur DAMADICS et leur schéma de fonctionnement.	68
Figure III.8: Modèle de l'actionneur	68

Figure III.9: Les composants du modèle DABLIB.....	69
Figure III.10: L'allure d'un défaut	70
Figure III.11. Les différents types de défauts artificiels.....	70
Figure III.12: Schéma générale de l'apprentissage du modèle neuronal.....	73
Figure III.13 : L'erreur quadratique d'apprentissage du modèle netF.....	73
Figure III.14: L'erreur quadratique d'apprentissage du modèle netX.....	74
Figure III.15: La régression linéaire entre la sortie du modèle netF et la sortie réelle F.....	75
Figure III.16: La régression linéaire entre la sortie du modèle netX et la sortie réelle X.....	75
Figure III.17: (a) La sortie réelle F et la sortie du modèle netf (b): l'erreur entre ces sorties.....	75
Figure III.18: (c) La sortie X et la sortie du netX. (d) l'erreur entre X et la sortie du netX.....	76
Figure III.19: Schéma général de la modélisation neuro-flou	76
Figure III.20: Le réseau du modèle (Rx).....	77
Figure III.21: Le réseau du modèle (Rf).....	77
Figure III.22: la sortie d'apprentissage du modèle Rx.....	79
Figure III.23 : La sortie de la phase de teste.....	79
Figure III.24: La sortie des données de validation du Rx.....	80
Figure III.25: la sortie d'apprentissage (Rf).....	80
Figure III.26: La sortie de teste pour Rf.....	80
Figure III.27: La sortie de validation du Rf.....	81
Figure III.28: La sortie réelle du système et la sortie du modèle et l'erreur entre eux pour Rx.....	81
Figure III.29: La sortie du système et la sortie du modèle et l'erreur entre eux pour Rf.....	82
Figure III.30: Génération des résidus par les réseaux de neurone.....	83
Figure III.31: (a) Résidu rF, (b) Résidu rX dans le cas de f7.....	83
Figure III.32: (a) Résidu rF, (b) Résidu rX dans le cas de f10.....	84
Figure III.33: (a) Résidu rF, (b) Résidu rX dans le cas de f15.....	84
Figure III.34: (a) Résidu rF, (b) Résidu rX dans le cas de f17.....	85
Figure III.35: La génération des résidus par un modèle ANFIS.....	86
Figure III.36: Le résidu rX dans le cas ou le défaut f1 est présent dans le système.....	87
Figure III.37: Le résidu rX dans le cas de défaut f13 est présent dans le système.....	87
Figure III.38: Le résidu rF dans le cas du défaut f1 présent dans le système.....	88
Figure III.39: Les résidus rF dans le cas du défaut f13 présent dans le système.....	88
Figure III.40: Les résidus dans le cas de défaut f1 après le filtrage.....	89
Figure III.41: Les résidus dans le cas de défaut f13 après le filtrage.....	89
Figure IV.1: La représentation des trois résidus structurés.....	93
Figure IV.2: Matrices de codages différents pour trois résidus et trois défauts.....	93
Figure IV.3: Construction d'un modèle défectueux.....	94
Figure IV.4: Conception de modèle neuronal des comportements défectueux.....	95
Figure IV.5: Exemple d'une courbe COR.....	96
Figure IV.6: Deux structurations des résidus fortement isolables.....	98
Figure IV.7: Exemple de Matrice de signature.....	99
Figure IV.8: Exemple des résidus implémenter dans plan (X,Y).....	100
Figure IV.9 : Classifieur neuronal utilisé.....	102
Figure IV.10: Procédure d'élaboration d'un classifieur neuronal.....	103
Figure IV.11: L'erreur quadratique d'apprentissage du Class1.....	104
Figure IV.12: La régression linéaire entre la sortie du Class1 et la cible.....	104
Figure IV.13 : La sortie f7 du classifieur neuronal.....	104
Figure IV.14 : La sortie f10 du classifieur neuronal.....	105
Figure IV.15: La sortie f15 du classifieur neuronal.....	105
Figure IV.16 : La sortie f17 du classifieur neuronal.....	105
Figure IV.17: Classifieur flou des défauts.....	106
Figure IV.18: Choix des fonctions d'appartenances pour une variable d'entrée.....	107
Figure IV.19: La sortie f7 du classifieur flou.....	108

Figure IV.20: La sortie f10 du classifieur flou	108
Figure IV.21: La sortie f15 du classifieur flou	108
Figure IV.22: La sortie f17 du classifieur flou	109
Figure IV.23: Architecture du classifieur Neuro-Flou des défauts	109
Figure IV.24: Représentation de l'apprentissage d'Anfis	109
Figure IV.25: La sortie du classifieur Neuro-Flou dans le cas du défaut f7	110
Figure IV.26: La sortie du classifieur Neuro-Flou Dans le cas du défaut f10	110
Figure IV.27: La sortie du classifieur Neuro-Flou dans le cas du défaut f15	110
Figure IV.28: La sortie du classifieur Neuro-Flou dans le cas du défaut f17	111
Figure IV.29: Conception d'un FDI par des modèles de comportements sain et défectueux.....	111
Figure IV.30: Modèle neuronal NNFM(3) pour le cas du défaut f3.....	112
Figure IV.31: Générateur de résidu a base des réseaux de neurone.	113
Figure IV.32: Génération des résidus en cas de sans défaut.....	114
Figure IV.33: La sortie X du système et leur sortie X' du modèle neuronal	115
Figure IV.34: La sortie F du système et leur sortie F' du modèle neuronal.....	115
Figure IV.35: Génération du résidu en cas de défaut f1.....	116
Figure IV.36: Génération du résidu en cas de défaut f2.....	116
Figure IV.37: Génération du résidu en cas de défaut f3.....	117
Figure IV.38: Génération du résidu en cas de défaut f4.....	117
Figure IV.39: Génération du résidu en cas de défaut f5.....	119
Figure IV.40: Génération du résidu en cas de défaut f6.....	119
Figure IV.41: Génération du résidu en cas de défaut f7.....	119
Figure IV.42: Génération du résidu en cas de défaut f8.....	119
Figure IV.43: Génération du résidu en cas de défaut f9.....	120
Figure IV.44: Génération du résidu en cas de défaut f10.....	120
Figure IV.45: Génération du résidu en cas de défaut f11.....	121
Figure IV.46: Génération des résidus si le cas de défaut f12	121
Figure IV.47: Génération du résidu en cas de défaut f13.....	122
Figure IV.48: Génération du résidu en cas de défaut f14.....	123
Figure IV.49: Génération du résidu en cas de défaut f15.....	123
Figure IV.50: Génération du résidu en cas de défaut f16.....	123
Figure IV.51: Génération du résidu en cas de défaut f17.....	124
Figure IV.52: Génération du résidu en cas de défaut f18.....	124
Figure IV.53: Génération du résidu en cas de défaut f19.....	125
Figure IV.54: Résidus r10, r20 où le défaut f15 est simulé entre l'intervalle [444s 1000s].....	128
Figure IV.55: Diagnostic hors ligne: localisation du modèle de connaissance de défaut FM(j) dans le plan (ρ_1 , ρ_2) pour les candidats des défauts f1 à f19	128
Figure IV.56: Diagnostic en ligne: localisation du modèle de connaissance de défaut FM(j) dans le plan (ρ_1 , ρ_2) pour les défauts candidats {f1, f2, f3}.....	129
Figure IV.57: Diagnostic en ligne: localisation du modèle de connaissance de défaut FM(j) dans le plan (ρ_1 , ρ_2) pour les défauts candidats f1 à f19.....	130
Figure IV.59: Résidus pour le groupe G1 (X-label: le temps(s), Y-label: l'amplitude des résidus) pour un défaut f3 simulé à partir de t= 487s	131
Figure IV.60: L'évaluation des performances pour les FDI du défaut f3 - (X-label: le temps(s), Y-label: la probabilité et le facteur de confiance) -Gauche) Probabilité P3(t, 50,501); -Droit) facteur de confiance CF(t, 50, 501).....	132
Figure IV.61: Résidus du groupe G2 (X-label: le temps (s), Y-label: l'amplitude des résidus) lorsque f15 simulé à partir de t = 451s.....	133
Figure IV.62: Diagnostic précoce dans le cas du défaut f15 (X-label: ρ_1 , Y-label: ρ_2): Gauche) l'emplacement du modèles NNFM (j); à droite) Détails.....	130

Figure IV.63: L'évaluation des performances pour FDI du défaut f15 (X-label: le temps (s), Y-label probabilité et facteur de confiance).-Gauche) Probabilité $P_{15}(t, 50, 458)$; -Droite) facteur de confiance $CF(t, 50, 458)$ 134

Figure IV.64: Résidus du groupe G3 (X-label : time (s), Y-label : Amplitude des résidus) Où f5 est simulé entre $t = 302s$ à $1000s$ 135

Figure IV.65: L'évaluation des performances pour les FDI du défaut f5 (X-label: le temps(s), Y-label: la probabilité et le facteur de confiance) -Gauche) Probabilité P_5 ; right) -Droit) Facteur de confiance CF 135

Figure IV.66: L'histogramme pour la projection des résidus des modèles des comportements sain et défectueux (NNFM(j)) sur les deux axes X et Y dans le cas où f12 du groupe 137

Figure IV.67: L'histogramme pour la projection des résidus des modèles des comportements sain et défectueux (NNFM(j)) sur les deux axes X et Y dans le cas où f15 du groupe G2 137

Figure IV.68: L'histogramme pour la projection des résidus des modèles des comportements sain et défectueux (NNFM(j)) sur les deux axes X et Y dans le cas où f8 du groupe G3 137

Figure IV.69: Ci-dessus: Résidus des modèles des comportements sain et défectueux (NNFM(12)) du groupe G2 (X-label: p_1 , Y-label: p_2). Ci-dessous: Les résidus de NNFM(12) sont pivotés vers l'axe des abscisses 138

Liste des tableaux

Tableau I.1: Table de codage pour trois résidus pour trois défauts	30
Tableau II.1: Différentes formes des fonctions d'activation et leur commande avec le logiciel Matlab.....	36
Tableau II.2: Description des Individus	50
Tableau II.3: Représentation quantitative des Individus	51
Tableau III.1: les différentes couches d'un système ANFIS	64
Tableau III.2 : les variables de notre système et leurs caractéristiques	68
Tableau III.3: Les défauts de l'actionneur et leur type	71
Tableau III.4: Apprentissage des réseaux optimaux obtenus	72
Tableau III.5: Les meilleurs essais pour le réseau Rx obtenu	81
Tableau III.6: Les meilleurs essais pour le réseau Rf obtenu	82
Tableau IV.1: Comparaison des variances des résidus sur les axes	101
Tableau IV.2: Apprentissage de réseaux de classification	103
Tableau IV.3: Table d'inférence du défaut f_7	105
Tableau IV.4: Table d'inférence du défaut f_{10}	105
Tableau IV.4: Table d'inférence du défaut f_{10}	106
Tableau IV.5: Table d'inférence du défaut f_{15}	106
Tableau IV.6: Table d'inférence du défaut f_{17}	106
Tableau IV.7: Les signatures des 19 défauts.....	125
Tableau IV.8: Le retard à la détection en fonction de σ	125
Tableau IV.9: Diagnostic hors ligne, selon les modèles de connaissance défectueux.....	127
Tableau IV.10: Diagnostic hors ligne pour le défaut f_3 avec $T = t = 1000s$	130
Tableau IV.11: Diagnostic hors ligne pour le défaut f_{15} avec $T = t = 1000s$	132
Tableau IV.12: Comparaison des méthodes FDI (+: la décision est correcte, -: la décision est incorrecte)	135

Notations

- $r(t)$: Signal de résidu.
- $x(t)$: Signal mesuré.
- S : Seuil constant.
- t_f : Instant d'occurrence du défaut.
- f_i : Défaut i .
- $y_i(t)$: Signal de sortie i .
- $\hat{y}_i(t)$: Signal estimé i .
- $u_i(t)$: Signal d'entrée i .
- θ : Paramètre du modèle estimé.
- φ : Paramètres du système.
- M : Matrice d'incidence.
- τ_i : Retard de détection d'un défaut i .
- W : Matrices des paramètres.
- w : Poids du neurone.
- E : Erreur quadratique.
- d : Sortie désirée.
- η : Pas d'apprentissage.
- g : Gradient.
- C : Matrice symétrique.
- H : Matrice du Hessien.
- μ : Degré d'appartenance.
- A : Ensemble flou.
- CV : Variable de commande est le signal de sortie du contrôleur.
- $P1$: Valeur de la pression sur l'admission de soupape de commande.
- $P2$: Valeur de la pression sur la sortie de soupape de commande.
- $T1$: Température du milieu.
- X : Valeur du déplacement de tige.
- F : Flux moyen.
- t_{from} : Temps d'apparition du défaut.
- f_{sMax} : Amplitude du défaut.
- t_{fd} : Temps de réponse du défaut.
- t_{to} : Temps de correction du défaut.
- f_1 Bouchage de valve.
- f_2 Dépôt de prise de valve ou de siège de valve.
- f_3 Erosion de prise de valve ou de siège de valve.
- f_4 Accru de frottement de valve ou de douille.
- f_5 Fuite externe (douille, couverture, borne perméable).
- f_6 Fuite interne (étanchéité de valve).
- f_7 Evaporation moyenne ou écoulement critique.
- f_8 Tige de piston du servomoteur tordu.
- f_9 Étanchéité des bornes du servomoteur.
- f_{10} Perforation du diaphragme du servomoteur.
- f_{11} Défaut du ressort du servomoteur.
- f_{12} Défaut électropneumatique du transducteur.
- f_{13} Défaut de la sonde de déplacement de Rod.
- f_{14} Défaut de la sonde de pression.
- f_{15} Défaut de rotation du positionneur.

-
- f_{16} Chute de pression par le positionneur.
 - f_{17} Changement inattendu de pression de la valve.
 - f_{18} Clapets de dérivation entièrement ou partiellement ouverts.
 - f_{19} Défaut de la sonde de débit.
 - R_x : Modèle du système de sortie X .
 - R_f : Modèle du système de sortie F .
 - rX : Résidu entre la sortie réelle X et X' du modèle.
 - rF : Résidu entre la sortie réelle F et F' du modèle.
 - H_1 : Hypothèse simples.
 - $P(A=H)$ Probabilité d'un événement A sous l'hypothèse H .
 - R_i : Résidus additif.
 - D : Norme euclidienne.
 - ρ : Résidus cumulés.
 - P_i : Probabilité dans le cas du défaut i .
 - CF : Facteur de confiance.
 - σ : Ecart type.
 - \bar{m} : Valeur moyenne.
 - GN : Grand négatif.
 - PN : Petit négatif.
 - Z : Zéro.
 - PP : Petit positif.
 - GP : Grand positif.
 - TF : Très faible.
 - F : Faible.
 - M : Moyenne.
 - G : Grande.
 - TG : Très grande.
 - ND : Non detection.
 - FA : Fausse Alarme.

Abréviations

- DAMADICS: Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems.
- IFAC: International Federation of Automatic Control.
- COR: Caractéristique Opérationnelle de Réception.
- MONET: Network of Excellence on Model Based Systems and Qualitative Reasoning.
- GRP: Groupement pour la Recherche en Productique.
- SPSF: Systèmes de Production Sûrs de Fonctionnement.
- FDI: Fault Detection and Isolation.
- AQT: Analyse Qualitative de Tendence.
- SOS: Simplified Observer Scheme.
- DOS: Dedicated Observer Scheme.
- GOS: Generalized Observer Scheme.
- MLP: Multi Layers Perceptron.
- RNA: Réseau de Neurones Artificiels.
- PMC: Perceptron Multi-Couche.
- RBF: Radial Basis Functions.
- ART: Adaptative Resonance Theorie.
- MIT: Massachusetts Institute of Technology.
- OBS: Optimal Brain Surgeon.
- OBD: Optimal Brain Damage.
- ANFIS: Adaptive Network Based Fuzzy Inference System.
- MSE: Mean Squared Error.
- FM: Faulty Model.
- NNFM: Neuronal Networks Faulty Model.
- AIC: Akaike Information Criteria.
- FPE: Final Prediction Error.

RESUME: Le bon fonctionnement d'un système nécessite l'utilisation des techniques de diagnostic capables de détecter, de localiser, et d'identifier les défaillances. La détection et la localisation des défauts sont réalisées en vérifiant la cohérence entre les différentes mesures disponibles sur le système et les informations sur son comportement attendu, en utilisant ou non un modèle analytique du système. Cette cohérence est usuellement exprimée par des signaux spécifiques, appelés « résidus » dont les variations sont indicatrices de défauts. L'identification des défauts complète le dispositif de diagnostic en précisant les caractéristiques de la défaillance comme l'amplitude ou l'instant d'occurrence. L'objectif du travail de recherche consiste à construire et à analyser des résidus par les outils de l'intelligence artificielle. Les réseaux de neurones et les systèmes neuro-flous sont privilégiés. L'étude inclut également une comparaison entre les apports de ces outils et de leurs limites, notamment au travers de l'établissement d'indicateurs quantifiés pour évaluer leur performance (taux de fausse alarme, taux de non détection et retard à la détection).

Cette thèse présente un système de diagnostic complet pour la détection et l'isolement des défauts dans des systèmes non linéaires inconnus. Des modèles construits par les réseaux de neurones apprennent les comportements sans-défaut et défectueux des systèmes considérés. Puis les résidus sont conçus et les probabilités de défaut sont évaluées pour chaque défaut candidat. Selon le calcul d'un facteur de confiance, la méthode proposée est également appropriée pour évaluer la fiabilité de la décision de diagnostic. L'approche est appliquée pour détecter et isoler 19 défauts candidats dans le benchmark DAMADICS. Les résultats obtenus et le schéma proposé sont comparés avec les résultats obtenus selon la méthode de seuillage habituelle.

Mots-clés: Détection de défaut, Diagnostic de défaut, Modèle des comportements sains, Modèle des comportements défectueux, Réseaux de Neurones, Probabilité, Facteur de Confiance.

ABSTRACT: The best operation of a system requires the use diagnosis techniques in order to detect, isolate, and identify the faults. Detection and isolation of faults are realized by checking the consistency between the various available measurements and the information about its expected behavior. This coherence is usually expressed by specific signals, called "residuals" for which changes are indicators of faults. The identification of faults completes the diagnosis device by specifying the characteristics of the fault as the amplitude and the time of occurrence. The objective of the research is to construct and analyze residuals by tools from artificial intelligence. Neural networks and neuro-fuzzy systems are preferred. The study also includes a comparison between the contributions of these tools and their limitations, particularly through the establishment of quantitative indicators to assess their performance (false alarm rates, non-detection rates and the detection delay).

This thesis presents a complete FDI scheme for the fault detection and isolation of unknown nonlinear systems. Neural networks models are designed that learn the fault-free and also the faulty behaviors of the considered systems. Then residuals are designed and fault probabilities are introduced for each fault candidate. According to the computation of a confidence factor, the proposed method is also suitable to evaluate the reliability of the FDI decision. The approach is applied to detect and isolate 19 fault candidates in the DAMADICS benchmark. The results obtained with the proposed scheme are compared with the results obtained according to a usual thresholding method.

Keywords: Fault detection, Fault diagnosis, Model of fault-free behaviors, Model of faulty behaviors, Neuronal networks, Probability, Confidence factor.

الملخص: الأداء السليم للنظام يتطلب استخدام تقنيات التشخيص بإمكانها اكتشاف، التعرف و تحديد موقع الأخطاء. إجراء الكشف ومكان الخطأ يتم عن طريق التحقق من الاتساق بين مختلف التدابير المتاحة للنظام وعلى معلومات حول السلوك المتوقع منه أو عدم استخدام نموذج تحليلي للنظام. عادة ما يتم التعبير عن هذا التماسك من خلال اشارات معينة، والمسماة "البقايا" و تغييراتها التي تدل على الأخطاء. تحديد العيوب تكمل طريقة التشخيص عن طريق تحديد خصائص الخطأ كما مثل اتساعه أو زمن وقوعه. الهدف من هذا البحث بناء وتحليل البقايا باستعمال أدوات الذكاء الاصطناعي. بفضل الشبكات العصبية ونظم الأعصاب غامضة. وتشمل الدراسة أيضا مقارنة بين المساهمات من هذه الأدوات ومحدداتها، ولا سيما من خلال وضع مؤشرات كمية لتقييم أدائها (معدل الانذار الزائف، معدل عدم الاكتشاف و التأخير في الكشف).

تقدم الأطروحة هذه نظام تشخيص كامل لكشف وعزل الخطأ في أنظمة الغير خطية الغير معروفة. وذلك ببناء نماذج من الشبكات العصبية تتعلم سلوكيات الأنظمة الصحيحة و المخطئة. ثم يتم تصميم البقايا مع تقييم الاحتمالات الافتراضية لكل مرشح خطأ. و وفقا لحساب عامل الثقة، والطريقة المقترحة هو أيضا مناسبة لتقييم مدى موثوقية القرار التشخيص. يتم تطبيق هذه التقنية لكشف وعزل 19 خطأ مرشحا في القياسي DAMADICS. النتائج التي حصلنا عليها و المخطط المقترح تتم مقارنتهم مع النتائج التي حصل عليها الأسلوب المعتاد باستعمال تقنية العتبة.

كلمات البحث: اكتشاف الخطأ، تشخيص الخطأ، نموذج ذو سلوكيات خال من الخطأ ، نموذج ذو سلوكيات الخاطئة، الشبكات العصبية، الاحتمال عامل الثقة.

Introduction générale

L'automatisation de procédés dans l'industrie et dans la vie quotidienne a atteint un niveau élevé. Les contraintes imposées aux installations industrielles en termes de productivité, de sécurité ou de qualité de production, rendent le diagnostic des défauts en ligne particulièrement important. Le diagnostic est la détection et l'identification des défauts dans les systèmes dynamiques. Ce sujet de recherche est un enjeu important de l'automatique moderne. En effet dans beaucoup d'applications il s'agira souvent, au delà de considérations purement économiques, d'assurer la sécurité des personnes et de préserver leur environnement. Les systèmes de surveillance et de supervision jouent un rôle majeur pour la sécurité des installations industrielles et la disponibilité des équipements. La phase de diagnostic et de décision est primordiale dans certains types d'installations tels que les procédés nucléaires, l'industrie spatiale, chimique, pétrochimique...etc. Ces procédés présentent une grande complexité pour la commande, surveillance et la supervision en raison de leur caractère hybride (aspects continus et discrets étroitement liés), du nombre de variables mises en jeu et de la complexité de leurs relations. Cette complexité est accentuée par la nécessité de nombreux changements de modes opératoires qui conduisent à de nombreuses phases transitoires. La mauvaise gestion de ces phases transitoires, durant lesquelles l'opérateur est fortement impliqué, peut avoir des conséquences dramatiques pour les produits et les équipements. Des procédures de conduite très strictes doivent être suivies. L'opérateur doit enchaîner un certain nombre d'actions en respectant les contraintes temporelles imposées.

On dit qu'il y a un défaut dans un système dynamique lorsque la différence entre le comportement du système et son modèle nominal est importante. Cependant, dans la plupart des applications, les défauts ne sont pas pris en compte, et souvent un petit défaut peut avoir un grand impact sur l'évolution d'un système. La demande croissante de disponibilité, d'efficacité, de qualité, de fiabilité et de sûreté, des approches de traitement et de prise en compte de défauts sont devenues une question cruciale.

Effectuer le diagnostic de fonctionnement d'un système consiste à détecter et localiser les défauts internes (affectant le processus lui-même), les défauts externes (affectant les actionneurs, les capteurs), les fonctionnements anormaux, puis à estimer les caractéristiques principales des défauts mis en évidence (leurs amplitudes). Dans la plupart des situations pratiques, le diagnostic de fonctionnement doit être effectué en présence de bruits de mesure, de perturbations sur le processus et d'erreurs de modélisation. Les performances de la procédure de diagnostic en termes de détection et localisation de défauts dépendent donc directement de l'exactitude de ces relations [Fra, 00]. De façon générale, un système de surveillance et supervision réalise les fonctions suivantes :

- La détection de défauts : la détection permet de déterminer si le système physique fonctionne normalement et a pour objectif de signaler la présence d'un défaut en comparant le comportement courant du système avec celui donné pour référence.
- La localisation de défauts : la localisation a pour objet la localisation d'un défaut détecté survenant sur les actionneurs, les capteurs d'instrumentation, la commande ou le système commandé en indiquant quel organe ou composant est affecté par celui-ci.

- L'identification de défauts : l'identification a pour but de caractériser le défaut en durée et en amplitude afin de le classer par types et degrés de sévérité. Le suivi de son évolution est fort utile dans le cas d'un changement de comportement lent dû au vieillissement et à l'usure. De plus, l'identification peut comprendre une procédure visant à déterminer la cause du défaut, c'est-à-dire son origine.

L'évolution technologique a contribué au développement de nouvelles procédures de surveillance et de supervision qui permettent la détection, la localisation et le diagnostic des éventuels défauts. Ces procédures ou techniques de diagnostic comprennent le plus souvent une étape de génération d'indicateurs de défauts ou résidus. Ces indicateurs caractérisent un écart par rapport aux conditions de fonctionnement attendues (plusieurs modes de fonctionnement peuvent être répertoriés). Une autre étape d'évaluation de ces résidus conduit ensuite à prendre une décision. Ou, pour générer ces résidus, on utilise des informations issues d'un modèle qualitatif ou quantitatif du système, afin de les comparer à celles fournies par les instruments de mesure. Une phase préliminaire d'analyse est donc nécessaire: c'est la structuration de la connaissance et la modélisation du système. L'intérêt d'utiliser conjointement la modélisation des comportements sains et défectueux du système réside dans le fait qu'en plus de l'estimation de l'état du système, les modèles défectueux procurent la probabilité d'occurrence ou d'activation de chaque mode de dysfonctionnement. Ces probabilités peuvent être utilisées pour le diagnostic.

Les réseaux de neurones et la logique floue peuvent être utilisés pour décrire le fonctionnement d'un système en termes de règles avec l'avantage de ne pas exiger la disponibilité d'information précise. Quand la connaissance experte n'est pas disponible l'applicabilité des systèmes flous peut être limitée. Les valeurs des paramètres des fonctions d'appartenance et le nombre précis de règles sont parfois difficiles à déterminer. Les réseaux de neurones peuvent alors être employés pour ajuster les fonctions d'appartenance des systèmes flous.

Le sujet proposé concerne le diagnostic des systèmes complexes (par exemple les systèmes non linéaires et commutés) en utilisant des méthodes de l'intelligence artificielle, et en particulier les réseaux de neurones pour construire des résidus en absence de modèle de référence. La problématique de génération automatique de résidus est importante car une telle construction, lorsqu'elle existe, permet de s'affranchir de toute connaissance analytique sur le système surveillé. Il s'agit de recourir systématiquement aux données collectées et archivées par les systèmes de supervision qui accompagnent généralement d'instrumentation des processus industriels.

Afin de rendre les contributions utiles et efficaces pour des systèmes réels, les méthodes proposées doivent dépendre d'un nombre minimal de paramètres. En effet les paramètres utilisés avec les méthodes usuelles de diagnostic, en particulier les seuils de détection et de classification, sont souvent difficiles à calibrer.

Pour ces raisons, l'étude proposée tente de s'affranchir de ces paramètres en incluant une analyse statistique des résidus et en utilisant une métrique dans l'espace d'observatio

Les méthodes développées sont testées et utilisées avec le banc d'essais DAMADICS

Plusieurs techniques de diagnostic développées pour le *DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems)*. La question principale à laquelle ce travail tente de répondre consiste à élaborer des résidus structurés (en

vue de la détection et de la localisation des défauts) à partir de données collectées sur les systèmes étudiés. Nous proposons comme hypothèse de répondre à cette question en exploitant les données avec des outils de l'intelligence artificielle pour la synthèse des résidus et leur analyse. Les difficultés liées à la mise en œuvre de méthodes d'apprentissage seront étudiées. L'étude inclut également une comparaison entre les apports de ces outils et leurs limites, notamment au travers de l'établissement d'indicateurs quantifiés pour évaluer leur performance (taux de fausse alarme, tau de non détection et retard à la détection).

L'étude comprend les phases suivantes :

- 1) Une recherche bibliographique sur les méthodes de diagnostic avec et sans modèle de référence. Une attention particulière sera portée aux méthodes basées sur les réseaux de neurones.
- 2) Le développement d'une méthode de synthèse et analyse de résidus par apprentissage en utilisant les mesures disponibles sur le système. La méthode pourra être particularisée pour répondre à des problèmes spécifiques comme ceux présentés par les systèmes commutés.
- 3) La validation de la méthode sur une plate forme d'expérimentation : le banc d'essais DAMADICS (*Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*).

L'analyse bibliographique effectuée a démontré que les RNA sont largement utilisés dans le domaine du diagnostic, en raison de leurs avantages et de leur simplicité.

A l'issue de l'étude bibliographique, nous avons proposé d'étudier des systèmes industriels et de développer un système de diagnostic à partir de ces étapes:

La première étape est basée sur la création d'un système de surveillance à base de modèle. Elle consiste à générer des indicateurs de défauts (résidus). Ils contiennent des informations sur les anomalies ou dysfonctionnements du système à surveiller. Le principe est de mesurer l'écart entre les mesures des signaux du procédé, capteurs ou actionneurs, et la valeur théorique fournie par le modèle créé par des approches d'intelligence artificielle comme les réseaux de neurone et les réseaux neuro-flou. La génération de résidus est un problème crucial pour les systèmes de diagnostic. En effet, de la structure du système de résidus engendré dépendra la robustesse de la localisation. Les techniques dérivées de cette modélisation sont variées mais pas toujours efficaces. En effet, les procédés physiques sont bien souvent des systèmes dynamiques complexes, présentant de fortes non-linéarités dans leur fonctionnement. Des simplifications s'imposent pour formuler un modèle exploitable à base des techniques de l'intelligence artificielle. De même, d'autres problèmes subsistent avec certains paramètres du modèle mesurables ou variant dans le temps.

La deuxième étape consiste à étudier des défauts susceptibles de survenir au niveau d'un actionneur électro-pneumatique. Cet actionneur est un élément de commande, dans l'usine de production de sucre à Lublin, Pologne. Il constitue un Benchmark mis à la disposition de différents groupes de recherche à travers le monde, pour tester différentes techniques de diagnostic (DAMADICS : *Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*).

Ce mémoire de thèse est décomposé en quatre chapitres et organisé de la façon suivante:

Chapitre 1 : Etat de l'art

Ce chapitre propose un tour d'horizon sur la supervision et les différentes approches de détection et de diagnostic de défaillance développées par les différentes communautés de recherche. Nous insisterons tout particulièrement sur les systèmes de diagnostic à base de modèle, dont le principe repose sur la comparaison entre les mesures générées et leur estimation. La problématique sera abordée selon les trois principes fondamentaux du diagnostic que sont la génération d'indicateurs de défauts, la détection et la classification.

Chapitre 2 : Aperçu et application des Réseaux de Neurones, Logique Floue et Neuro-Flou dans le Diagnostic.

Ce chapitre est composé en deux grandes parties :

- La première partie présente des notions générales sur les réseaux de neurones, la logique floue et le neuro-flou.
- La deuxième partie résume brièvement leurs applications dans le cas de notre travail.

Chapitre 3: Méthode intelligentes pour la détection des défauts

Dans ce chapitre, deux modèles seront présentés en détail pour la génération des résidus et la détection, l'un est un modèle neuronal et l'autre un modèle Neuro-Flou. Ces deux modèles simulent bien les comportements sains de l'actionneur industriel. Nous pouvons utiliser ces deux modèles pour la génération des résidus et la détection des défauts de l'actionneur.

Chapitre 4: Diagnostic rapide par synthèse de modèles de dysfonctionnement.

Ce chapitre expose notre contribution concernant le développement d'une technique capable de faire un diagnostic rapide en ligne. Dans cette étape nous allons utiliser les réseaux de neurones comme outil pour créer des modèles des comportements sain et défectueux et appliqué au benchmark DAMADICS pour détecter et isoler les 19 défauts existants dans cet actionneur.

Finalement, la conclusion récapitule le travail développé, commente les résultats trouvés et indique les perspectives de recherche ouvertes par cette étude.

Chapitre I

Etat de L'art

Sommaire

I.1	Introduction	11
I.2	Terminologie et définition	11
I.3	Les méthodes de diagnostic	17
I.3.1	Les méthodes à base de modèle	19
I.3.2	Les méthodes à base de connaissances et de données	22
I.4	Les étapes de diagnostic	25
I.4.1	Génération de résidus	25
I.4.2	La détection de défaut	26
I.4.3	La localisation de défaut	27
I.4.4	L'identification de défaut	28
I.5	La structuration des résidus	29
I.6	Conclusion	31

I.1 Introduction

L'automatisation des processus technologiques fait appel à des méthodes qui deviennent de plus en plus sophistiquées et complexes afin d'augmenter les performances, la fiabilité, la disponibilité et la sûreté de fonctionnement de ces processus. Le besoin de sûreté de fonctionnement et de fiabilité est encore plus crucial lorsqu'il s'agit de systèmes sensibles pour lesquels une fausse manœuvre peut être coûteuse aussi bien humainement que financièrement. Ainsi, afin d'atteindre ces objectifs de performance, de sécurité et de disponibilité des processus technologiques, on leur associe des modules de diagnostic servant à détecter tout écart de comportement par rapport au comportement souhaité et même dans certaines situations à reconfigurer le fonctionnement du système.

Un module de diagnostic procède en trois étapes fondamentales. Dans un premier temps, il s'agit de déterminer si le processus considéré se comporte ou opère normalement, c'est-à-dire s'il remplit le cahier des charges que l'ingénieur aura fixé lors de la conception du système en question. C'est l'étape dite de détection. Arrive ensuite la seconde étape qui a pour objet la recherche de certaines caractéristiques du défaut comme son instant d'apparition, son amplitude, sa gravité. Enfin, la dernière étape, qui découle des deux étapes précédentes, permet de décider de l'action à entreprendre sur le système. Il en résultera une décision soit de maintenir le système dans le même mode opératoire, de corriger son fonctionnement ou encore de l'arrêter complètement.

En consultant la très abondante littérature existante dans le domaine du diagnostic, on se rend assez vite compte du non unicité de la terminologie dans ce domaine. Le comité technique IFAC [1] (International Federation of Automatic Control) SAFEPROCESS a tenté de normaliser certaines définitions généralement acceptées par l'ensemble de la communauté de l'Automatique. Nous rappellerons dans ce qui suit les définitions données à certains principaux termes propres au domaine du diagnostic.

I.2 Terminologie et définition

Durant ces dernières années, un effort a été réalisé par les communautés liées à la supervision et au diagnostic de façon à établir un vocabulaire et une terminologie communs. Nous avons retenu, dans ce mémoire de thèse, des définitions issues de sources bibliographiques diverses. La plupart d'entre elles ont été discutées au sein du SAFEPROCESS (*Technical Committee of International Federation of Automatic Control*) [2] et condensées sous la forme d'un glossaire dans le cadre du projet du groupe de recherche européen Bridge (*BridgeTask Group*) issu du réseau d'excellence MONET (*Network of Excellence on Model Based Systems and Qualitative Reasoning*) [3]. D'autres ont été discutées au sein des groupes GRP (*Groupement pour la Recherche en Productique*) et SPSF (*Systèmes de Production Sûrs de Fonctionnement*) [4]. Nous complétons cette terminologie avec quelques définitions provenant de la norme ISA 88 [ANSI/ISA-S88.01, (1995)] qui définit la terminologie et les modèles de base des principes de contrôle des procédés batch.

Définitions générales

- *Défaut*: une déviation non permise d'au moins une propriété ou d'un paramètre caractéristique du système par rapport aux conditions normales de fonctionnement.
- *Détection de défauts* : détermination de la présence de défauts et de l'instant d'occurrence de ces défauts.
- *Diagnostic* : détermination du type, de la taille, de la localisation et de l'instant d'occurrence d'un défaut. Fonction qui suit la détection et inclut les fonctions de localisation et d'identification.
- *Dysfonctionnement*: exécution d'une fonction du système au cours de laquelle le service rendu n'est pas délivré ou est délivré de manière incomplète.

- *Identification de défauts* : détermination de la taille et du comportement temporel d'un défaut. Fonction qui suit la localisation.
- *Localisation de défauts* : détermination du type, de la localisation et de la date d'un défaut détecté. Consiste à remonter d'un ensemble de symptômes à un ensemble de composants défaillants.
- *Panne ou défaillance*: interruption permanente de la capacité du système à remplir une fonction requise dans des conditions d'opérations spécifiées.
- *Perturbation* : signal d'entrée non contrôlé dont la présence est non souhaitable mais considérée comme normale.
- *Résidu* : indicateur de défaut basé sur la déviation entre les mesures et les calculs basés sur un modèle.
- *Suivi* : fonction maintenant en permanence un historique des traitements effectués par le système de commande/supervision et une trace des événements que perçoit le système.
- *Supervision* : représente la surveillance d'un système physique et la prise de décisions appropriés en vue de maintenir son opération face à des défaillances.

Une deuxième définition, peut être donnée à la supervision qu'on peut définir comme étant la surveillance et le contrôle de l'exécution d'une opération accomplie par un homme ou par une machine. Le dispositif de supervision assure la reconfiguration du système en fonction des résultats de surveillance. Suite à la détection d'une défaillance, le dispositif de supervision réagit pour l'accommodation de la défaillance, à l'aide des informations fournies par l'étude de sûreté et par le dispositif de surveillance. Les interventions peuvent être la reconfiguration de la commande (Figure I.1).

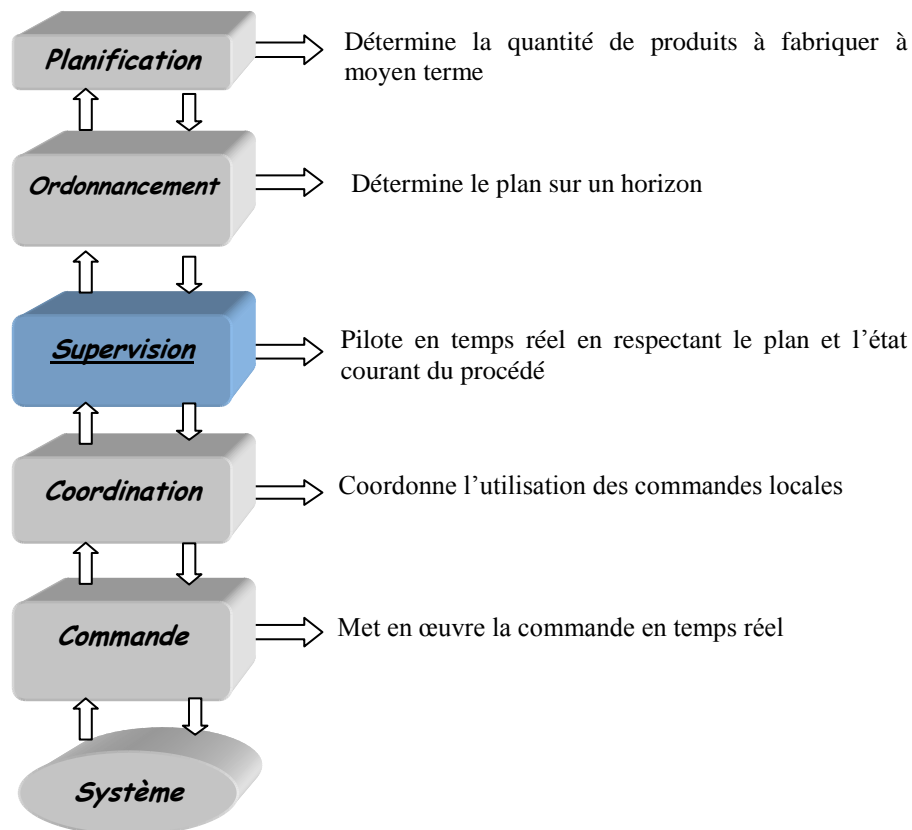


Figure I.1: Positionnement de la supervision

- *Surveillance* : ensemble de fonctions s'exécutant en temps réel ayant pour but de reconstituer l'état réel du procédé au sein de modèles utilisés par le système de conduite du

procédé. Consiste en l'enregistrement de l'information ainsi qu'en la reconnaissance et l'indication de comportements anormaux.

- *Symptôme* : changement d'une quantité observable par rapport à un comportement normal.
- *Trajectoire* : évolution temporelle d'une variable.

Pour la définition du diagnostic de défaut, une définition légèrement différente existe également dans la littérature. Cette définition peut être trouvée par exemple dans [Ger, 91] et indique que le diagnostic de défaut inclut également la détection de défaut.

Diagnostic

Étymologie : Nouveau latin, du grec *diagnOsis*, « à distinguer de ».

Date : 1655

- 1- **a** : l'art ou l'acte d'identifier une maladie à partir de ses signes et symptômes, **b** : la décision obtenue par diagnostic,
- 2- **a** : recherche ou analyse de la cause ou nature d'une condition, d'une situation, ou d'un problème < *diagnostic d'un défaut de moteur* >, **b** : un rapport ou une conclusion d'une telle analyse.

La première question que l'on se pose lorsqu'on conçoit un système de diagnostic, est de savoir ce que l'on veut détecter, c'est à dire de définir le type de dysfonctionnement que l'on veut diagnostiquer et donc les défauts susceptibles d'altérer le bon fonctionnement du système. Ainsi on pourra être amené à détecter des biais, des dérives et/ou des valeurs aberrantes. Qu'il s'agisse de défauts inhérents aux organes de mesure (capteurs), aux organes de commande (actionneurs) ou aux composants du processus, ils se manifestent par une altération des signaux associés (Figure I.2).

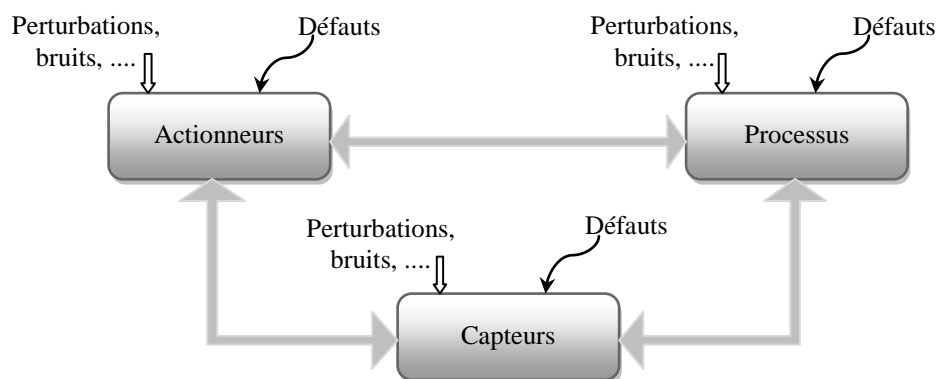


Figure I.2 : Les différents emplacements de Défauts dans le système

Dans une vision multi niveaux d'un processus complexe, il sera possible de définir au niveau des interfaces des sous processus de nouveaux éléments d'action ou de mesure et donc de définir plus finement les défaillances. Un biais correspond à un saut brutal du signal alors qu'une dérive se manifeste par une évolution anormale lente et continue du signal, et donc un éloignement progressif de sa valeur nominale.

Les phénomènes de dérive sont plus longs à détecter du fait de leur faible amplitude et de leur lente évolution. Les valeurs aberrantes sont des défauts dits fugitifs, affectent le système de manière instantanée. Ces défauts sont souvent dus à un parasite, par exemple une perturbation électromagnétique et correspondent à un écart important et sporadique par rapport à la valeur nominale du signal.

Le concept de la supervision s'applique donc dans un cadre hiérarchisé à deux niveaux au moins. Superviser le fonctionnement d'installation industrielle est devenu aujourd'hui le rôle principal des opérateurs humains, avec l'aide des systèmes automatiques assurant localement

le contrôle et la commande. De nombreuses tâches telles que la prédiction, le diagnostic, le pronostic ou le conseil d'action relèvent de la supervision.

Le diagnostic devra être aussi robuste que possible vis-à-vis des perturbations et aussi sensible que possible aux défauts. La notion de « normalité » qui distingue défauts et perturbations est assez subjective : elle dépend de l'application et de la criticité des écarts de comportement.

Pour arriver à déterminer et à caractériser les anomalies d'un système, il est, nécessaire d'avoir des informations sur son comportement et/ou sur son fonctionnement. Ces informations peuvent être classées en deux catégories : les connaissances qui ne dépendent pas de l'environnement du système qui sont fournies par un modèle et les connaissances qui dépendent de l'environnement des systèmes (changement de consigne, perturbation...) qui sont fournies par des observations.

Un système de diagnostic peut être vu comme un capteur « intelligent » qui traite des observations (en particulier les mesures) en vue de fournir des observations ciblées sur les anomalies. Par conséquent, le traitement réalisant le diagnostic consiste à extraire les signes et les symptômes les plus pertinents des observations disponibles.

Domaine de la connaissance

- *Connaissance comportementale*: description de la relation d'entrée-sortie des composants du système. Le modèle de comportement du système décrit les trajectoires du système.
- *Connaissance topologique (structurelle)*: description de l'interconnexion des composants du procédé (modèle structurel).
- *Redondance d'informations* : Le concept de base des systèmes de diagnostic est la redondance d'informations. Cette redondance de connaissances sur le système fournit plusieurs informations différentes sur une même variable du système. Il est ainsi possible de vérifier la cohérence de cette information par des tests de cohérence que nous allons exposer. Celle-ci se divise en deux : la redondance physique et la redondance analytique.
- *Redondance physique* : Le moyen le plus facile d'obtenir plusieurs informations sur une même variable est de disposer de plusieurs capteurs mesurant la grandeur de cette variable: c'est le principe de la redondance physique. Par exemple, pour obtenir deux mesures d'une même température, on doublera les capteurs de température comme indiquée par le schéma de la Figure I.3. La redondance matérielle possède plusieurs avantages tels que la fiabilité et les bonnes performances. En revanche, celle-ci impose de nombreuses contraintes de coûts, de poids et d'encombrements, du fait des composants additionnels. Par ailleurs, tous les composants d'un système ne peuvent être dupliqués.

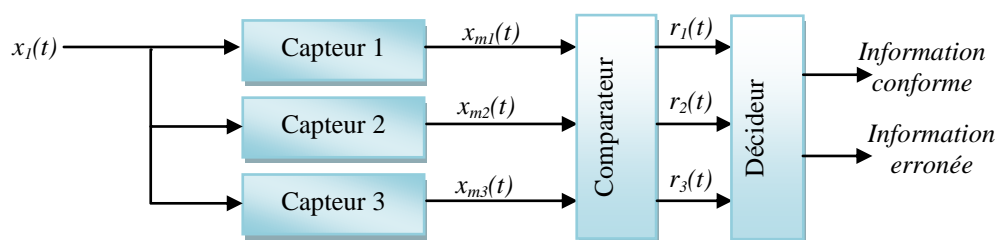


Figure I.3 : La redondance physique triple de l'information

Le *Comparateur* compare les trois signaux $x_{m1}(t)$, $x_{m2}(t)$ et $x_{m3}(t)$ mesurés respectivement par les trois capteurs et délivre respectivement trois signaux appelés "résidus" $r_1(t)$, $r_2(t)$ et $r_3(t)$ donnés par l'équation suivante:

$$\begin{cases} r_1(t) = x_{m1}(t) - x_{m2}(t) \\ r_2(t) = x_{m1}(t) - x_{m3}(t) \\ r_3(t) = x_{m2}(t) - x_{m3}(t) \end{cases} \quad (\text{I.1})$$

Le *Décideur* fournit suivant les résidus une information conforme ou erronée (capteur en panne).

L'approche redondance matérielle est très efficace et possède plusieurs avantages tels que la détection et la localisation des capteurs défaillants, aussi, elle permet de multiplier les chaînes de mesures [6]. Cependant cette dernière ne couvre pas les pannes de mode commun (panne d'alimentation électrique, panne de masse etc...) et présente des limites par rapport au coût et à l'encombrement. Ainsi l'inconvénient majeur réside dans le fait que le champ d'application de cette dernière est strictement limité aux pannes de capteurs.

– *Redondance analytique* : La redondance analytique présente une alternative à la redondance matérielle. La redondance dans ce cas est assurée par un modèle formel réalisant les mêmes performances que le système étudié, à la place du matériel additionnel.

La mise en œuvre d'une telle approche nécessite de générer des indicateurs de défauts ou symptômes (les résidus), puis à interpréter correctement ces symptômes afin de déterminer l'origine du défaut, c'est-à-dire l'élément présentant un fonctionnement anormal et enfin à prendre une décision pour un retour à un fonctionnement normal de l'installation.

Elle constitue l'élément fondamental de tout système de diagnostic à base de modèle. Elle consiste à utiliser des informations supplémentaires issues de modèles générant des grandeurs homogènes à celles provenant de capteurs. Elle exploite la redondance informationnelle contenue implicitement dans un ensemble de mesures. L'utilisation des techniques de redondance analytique repose sur la disponibilité d'un modèle de connaissance ou de représentation;

La redondance analytique dont l'utilisation permet la détection et la reconnaissance de défauts de fonctionnement de façon à effectuer des actions correctives appropriées, repose sur des relations de cause à effet, statiques ou dynamiques, qui existent entre les entrées et les sorties observées d'un système. Son champ d'application ne se limite donc pas aux pannes de capteurs mais s'étend aux pannes des actionneurs ou à celles du procédé lui-même.

La redondance analytique est basée sur un ensemble d'équations mathématiques reliant l'état du système aux variables d'entrées et de sorties. Un schéma général concernant son utilisation est présenté sur la (Figure I.4) L'idée de base de la redondance analytique est de comparer le comportement actuel du système avec son modèle pour l'identification. Une différence identifiée comme résidu peut être utilisée pour la détection des défaillances. Les résidus devront être proches du zéro lorsqu'il n'y a pas de défaillance. La génération de tel résidu est basée sur un modèle dérivé analytiquement [Vil, 88].

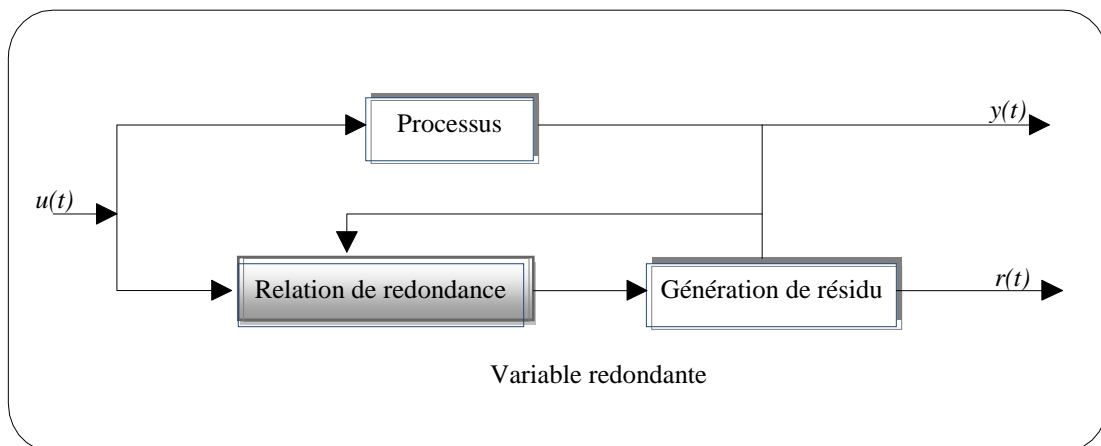


Figure I.4 : Principe des méthodes basées sur la redondance analytique

Les méthodes basées sur la redondance analytique donnent des résidus qui sont insensibles aux incertitudes et le contraire pour les défaillances. Une des techniques bien connues pour réaliser cette indépendance est la séparation des perturbations. Dans cette approche toutes les incertitudes sont traitées comme des perturbations et des filtres sont conçus pour séparer l'effet des défaillances et des perturbations pour permettre leur distinction [Vil, 88].

L'approche utilisant la redondance analytique se décompose généralement en deux phases distinctes. La première concerne la *génération de résidus* caractéristiques de la panne. Ils représentent les écarts entre le comportement observé du système et le comportement de référence attendu en fonctionnement "normal". Ces résidus sont généralement à moyenne nulle et ont une variance déterminée en l'absence de défauts de fonctionnement. Le problème qui se pose au concepteur du système de diagnostic est de sélectionner les résidus satisfaisant le compromis : sensibilité maximum aux pannes que l'on cherche à détecter-sensibilité minimum aux erreurs de modélisation et aux bruits de mesure [7].

Caractéristiques du modèle

- *Exactitude*: liée à la distance entre la solution du modèle et le comportement observé du système réel. L'opération *d'abstraction* conserve l'exactitude du modèle.
- *Granularité* : se rapporte au nombre de variables utilisé dans un modèle. L'élimination ou l'agrégation de certaines variables du modèle correspond à un processus *d'abstraction*.
- *Incertitude*: décrit le niveau de confiance que l'on peut accorder à la solution du modèle. Il y a l'incertitude liée à la valeur de paramètres, celle liée aux relations fonctionnelles entre les variables et celle correspondant aux perturbations non prises en compte par le modèle, que peut subir le système.
- *Précision* : reflète le niveau de détail mis pour exprimer les distinctions entre les valeurs des variables décrivant le système. Les *modèles analytiques* sont utilisés pour une représentation plus précise tandis que les *modèles qualitatives et heuristiques* demandent une connaissance moins précise.

Critères liés au diagnostic

- *Fiabilité* : consiste en un diagnostic précis de tous les défauts y compris les défauts non prévus.
- *Pouvoir d'explication*: capacité du système de diagnostic à fournir des explications sur comment un défaut s'est produit et s'est propagé jusqu'à la détection, et sur sa propagation au sein du procédé en justifiant la proposition des hypothèses.
- *Précision* : un diagnostic précis est bien celui qui contient un défaut correct dans l'ensemble de candidats (défauts).
- *Résolution* : correspond à la faculté du diagnostic à établir un ensemble minimal de candidats (le cas idéal un) en s'appuyant sur l'information issue du procédé.
- *Robustesse* : capacité du système à fournir un diagnostic précis malgré les incertitudes de modélisation, les perturbations ou les bruits de mesure.
- *Sensibilité* : capacité à détecter de faibles défauts qui proviennent de faibles changements de trajectoires des variables.

Norme ISA 88 [ANSI/ISA88 1995] [5] définie dans le cadre de procédés batch (hors ligne):

- *Instruction* : commande élémentaire orientée équipement adressée à un élément ou à un composant afin qu'il effectue un traitement donné.
- *Opération* : ensemble de phases exécutées en séquence ou en parallèle sur une unité de production. Une opération conduit le produit d'un état à un autre, induisant une modification des propriétés physiques et/ou chimiques de celui-ci.

- *Pas* : séquence temporelle ou événementielle d'instructions à exécuter dans un module (ensemble) d'équipements en vue d'accomplir une tâche spécifique.
- *Phase* : succession de pas permettant de réaliser une fonction élémentaire.
- *Procédure* : ensemble d'opérations à exécuter dans une ligne de production, pour fabriquer un produit particulier.
- *Recette* : utilisé pour spécifier le produit à réaliser, le type d'équipements requis, la procédure de traitement et les ingrédients, quantités, paramètres et temps de fabrication, ainsi que la formule assurant la fabrication d'une quantité désirée de produit.

Types de défauts: Selon Isermann [Ise, 02], les défauts peuvent être différenciés selon leur forme (systématique ou aléatoire), selon l'extension du défaut (local ou global) ou selon leur comportement dans le temps (Figure I.5). La notation t_f représente l'instant d'occurrence du défaut.

1- *Défaut abrupt*: Ce défaut a un comportement temporel discontinu, donné par la formule suivante:

$$f(t - t_f) = \begin{cases} \delta & t \geq t_f \\ 0 & t < t_f \end{cases} \quad (I.2)$$

Où $f(t - t_f)$ représente le comportement temporel du défaut et δ un seuil constant. Ce type de défauts est caractérisé par une discontinuité dans l'évolution temporelle de la variable. Cette évolution est caractéristique d'une panne brutale de l'élément en question: arrêt total ou partiel ou déconnexion, si elle ne correspond pas aux évolutions dynamiques normales attendues pour la variable.

2- *Défaut intermittent*: Ce défaut est un cas particulier du défaut abrupt avec la propriété particulière que le signal revient de façon aléatoire à sa valeur normale. Ce type de défauts caractérise les faux contacts.

3- *Défaut graduel ou naissant*: Ce défaut a un comportement temporel lent, donné par la formule suivante:

$$f(t - t_f) = \begin{cases} \delta(1 - e^{-\alpha(t-t_f)}) & t \geq t_f \\ 0 & t < t_f \end{cases} \quad (I.3)$$

Où α et δ sont des constantes positives. Il est très difficile à détecter car son évolution temporelle a la même signature que celle d'une modification paramétrique lente représentant un non stationnarité du procédé. Ce type de défauts est caractéristique d'un encrassement ou d'une usure de pièce.

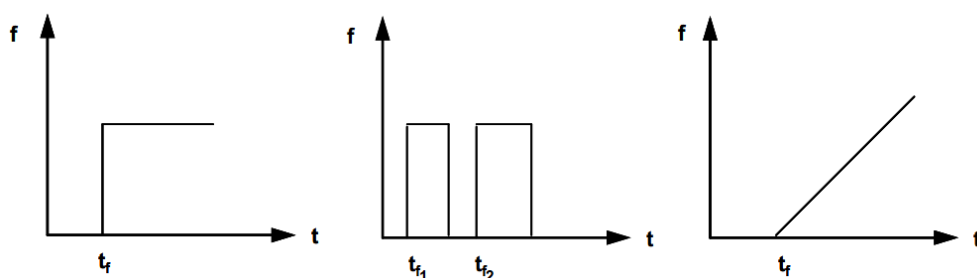


Figure I.5 : Evolutions temporelles des différents types de défauts

Notre travail sera principalement focalisé sur les défauts *abrupts* et *naissants*.

I.3 Les méthodes de diagnostic

Les méthodes de diagnostic des défaillances utilisées dans les différents secteurs industriels sont très variées. Leur principe général consiste à confronter les données relevées au cours du fonctionnement réel du système avec la connaissance que l'on a de son fonctionnement nominal ou de ses fonctionnements défaillants. La forme, sous laquelle se

présente la connaissance sur le système, conditionne les différentes méthodes utilisées pour le diagnostic.

Les techniques de diagnostic sont connues sous le nom de FDI (Fault Detection and Isolation) l'application de cette approche dans le cas des systèmes linéaires a été très largement étudiée. Les principaux résultats peuvent être trouvés dans [Zha, 00]; [Che, 99]; [Ger, 98]; [Ise, 97b]; [Pat, 97]; [Fra, 96]; [Mas, 89]; [Wil, 76a]. Par contre, la FDI pour les systèmes non linéaires est encore un champ de recherche important. Pour quelques systèmes non linéaires particuliers, des auteurs [Sre, 93] ont employé la linéarisation autour des points de fonctionnement afin d'appliquer des méthodes linéaires de FDI. En général, cela n'est pas toujours réaliste et cela est du aux fortes non linéarités (par exemple les effets de saturation ou comportement non analytique) ou a l'inefficacité de la linéarisation. Par conséquent, plusieurs approches de FDI ont été améliorées pour manipuler également des systèmes non linéaires plus généraux, par exemple l'approche basée sur les observateurs, l'approche de l'espace de parité et l'estimation paramétrique. Les observateurs flous et les réseaux de neurones artificiels ont été également considérés comme des outils potentiellement pertinents dans le cas des systèmes non linéaires. Récemment, de nouvelles approches ont été proposées pour la classe de systèmes non linéaires a entrée affine. Plus particulièrement une approche de FDI basée sur les observateurs pour les systèmes non linéaires a été étudiée et développée [Åst, 00]; [Ise, 94]; [Ven, 03a]. Nous trouvons également les observateurs a modes glissants, les observateurs à grand gain [Edw, 94]; [Xin, 05]; [Ham, 99] et les observateurs adaptatifs [Gar, 97]; [Che, 05]; [Li, 06]; [Zha, 00]. Suivant notre recherche bibliographique, on constate que la procédure de diagnostic et de surveillance des défaillances et des dégradations susceptibles d'affecter les différentes entités d'un processus industriel s'articule autour des étapes suivantes (Figure I.6):

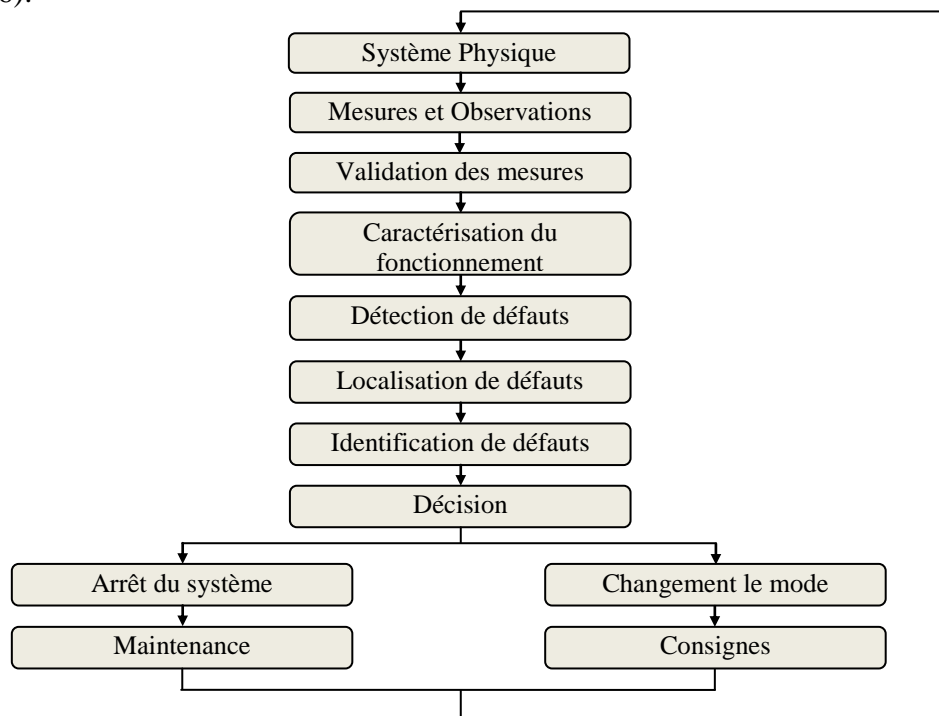


Figure I.6 : Modules d'une procédure de surveillance

A partir de cette figure, on peut noter que :

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

- L'élaboration des caractéristiques et signatures associées à des symptômes révélateurs de défaillances et de dégradations en vue de la détection d'un dysfonctionnement.
- La détection d'un dysfonctionnement par comparaison avec des signatures associées à des états de fonctionnement normaux et la définition d'indicateurs de confiance dans la détection.
- La mise en œuvre d'une méthode de diagnostic de la défaillance ou de la dégradation à partir de l'utilisation des connaissances sur les relations de cause à effet.
- La prise de décision en fonction des futures conséquences des défaillances et des dégradations. Cette prise de décision peut conduire à un arrêt de l'installation si les conséquences de la défaillance sont importantes pour la sécurité des personnes et des biens ou à une reconfiguration du fonctionnement du procédé pour éviter une perte de production en attendant le prochain arrêt de production le plus propice aux opérations de maintenance corrective.

La classification des méthodes de diagnostic la plus appropriée à un système industriel donnée ne peut se faire qu'après un recensement de besoin des connaissances disponibles, pour cela de nombreuses approches ont été développées pour contribuer au diagnostic des défauts (Figure I.7) Dans ce qui suit, nous allons présenter ces différentes approches [Zwi, 95]. Elles peuvent être groupées en trois grandes familles : les méthodes de diagnostic par modélisation fonctionnelle et matérielle ou méthodes basées sur le mode de raisonnement, les méthodes de diagnostic par modélisation physique et enfin les méthodes de diagnostic par analyse des signatures externes.

Les approches à base de modèles, reposent sur les principes physiques fondamentaux et sur des modèles quantitatifs ou qualitatifs incluant la structure du système et les liens entre composants [Pat, 89], [Ger, 91].

Les approches à base de connaissance, s'appuient sur les connaissances et les raisonnements experts [Cho, 80], [Ema, 88], [Fra, 89], [Adj, 93], [Beb, 07].

Les approches à base de données réalisent une extraction des caractéristiques du système à partir de données sauvegardées [Din, 91], [Wal, 79], [Wil, 76a], [Wil, 76b].

L'originalité de l'étude réside dans la construction directe des résidus sans construire de modèle de comportement du système étudié.

Dans ce contexte, cette étude s'attache à compléter les méthodes de diagnostic. Notre travail privilégie les techniques de l'intelligence artificielle (réseaux de neurones, logique floue et systèmes neuro-flous) afin d'étudier leur efficacité et d'évaluer leurs performances [Nau, 94], [Nau, 95], [Moh, 05].

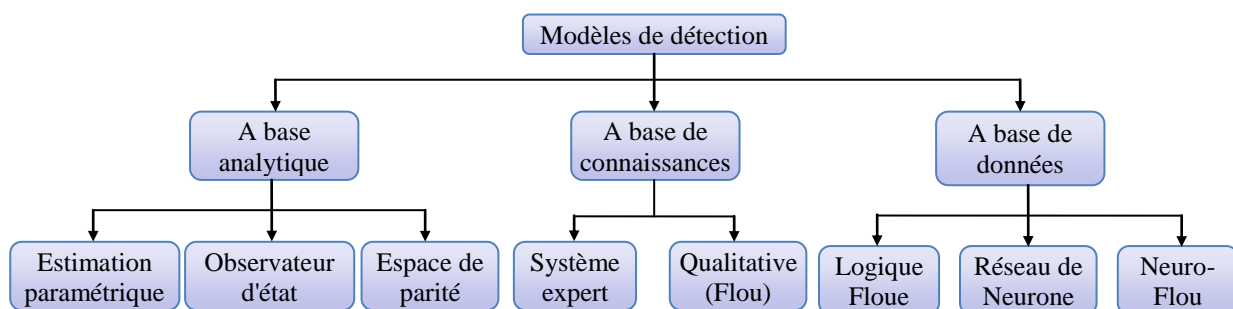


Figure I.7 : Les différentes méthodes de diagnostic

I.3.1 Les méthodes à base de modèle

Les méthodes de diagnostic à base de modèles occupent une place importante dans la littérature. Leur utilisation, notamment dans le cadre d'applications critiques (systèmes énergétiques, systèmes de transport, industrie lourde), s'est considérablement développée. L'objet de ce chapitre est donc de faire un état de l'art sur les méthodes de diagnostic à base

de modèle. Le diagnostic à base de modèle a été développé dès les années soixante-dix. On peut citer par exemple sur le plan méthodologique [Cla, 89], [Wil, 76a] [Ise, 84] [Cho, 84] [Pat, 89], [Fra, 90] [Ise, 97a] [Maq, 97] [Ger, 98] [Che, 99], alors que [Ise, 96] a fait le point sur les applications industrielles de cette approche. Cette approche FDI, comme on l'a vu précédemment, fait intervenir les techniques de génération de résidus, de détection et de localisation des défauts. Cette approche est schématisée sur la Figure I.8:

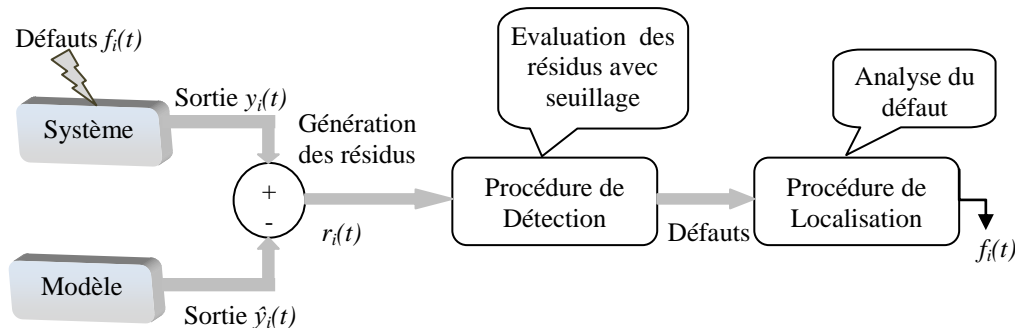


Figure I.8 : Procédure FDI

La première démarche dans la mise en œuvre d'un système de diagnostic à base de modèle consiste à générer des indicateurs de défauts. Ils contiennent des informations sur les anomalies ou dysfonctionnements du système à surveiller. Le principe est de mesurer l'écart entre les mesures des signaux du procédé, capteurs ou actionneurs, et la valeur théorique fournie par le modèle dans des conditions de fonctionnement nominal. La qualité de la génération de résidus est un élément essentiel pour garantir les performances d'un système de diagnostic. En effet, la robustesse de la détection et de la localisation dépendra de la structure (composition et précision) du système de résidus engendré.

a) *Estimation paramétrique:*

Les méthodes par estimation paramétrique ont pour principe d'estimer les paramètres du modèle. On entend par *paramètre* les constantes physiques du système (masse, coefficient de viscosité,...) ou une agrégation de plusieurs paramètres physiques. Pour ce dernier cas, une correspondance unique doit exister entre les paramètres du modèle et les paramètres du système. Alors le diagnostic par estimation paramétrique consiste à surveiller les changements de la valeur de certains paramètres caractéristiques. Des modèles paramétriques précis sont nécessaires, généralement exprimés sous formes d'équation différentielle ordinaires ou partielles Dans le cas de systèmes continus, ou d'équations récurrentes dans le cas numérique (Figure I.9). La procédure consiste à obtenir le modèle du processus en se basant seulement sur les entrées et les sorties mesurées :

$$y(t) = f(u(t), \theta) \quad (I.4)$$

Les paramètres du modèle θ sont estimés à l'aide des valeurs de $y(t)$ et $u(t)$. θ est relié aux paramètres réels du processus par $\theta = g(\varphi)$. Le changement dans les paramètres φ donné par $\Delta\varphi$ provoque une variation $\Delta\theta$ de θ [Mou, 05].

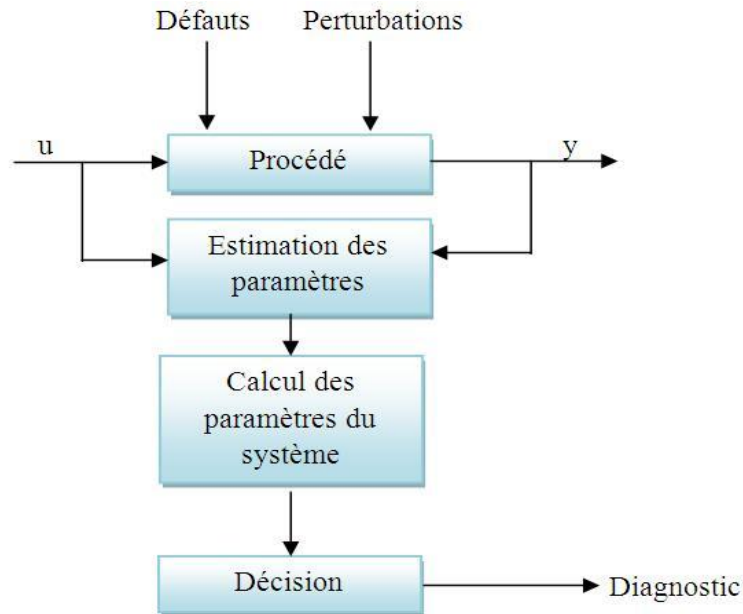


Figure I.9 : Diagnostic par estimation de paramètres

b) Observation d'état

Les méthodes d'estimation d'état ont pour principe de reconstruire les états et les sorties du système à partir des entrées et des sorties. Ce sont des méthodes de génération de résidus indirectes en ce sens qu'elles calculent l'erreur d'estimation de la sortie. L'estimation de l'état peut être effectuée à l'aide d'observateurs [Adj, 93] dans le cas déterministe ou de filtres dans le cas stochastique (filtre de Kalman [Wil, 76b] ou filtres détecteurs de défauts [Mas, 86]). Les deux méthodes présentent des analogies dans leurs formulations et peuvent être synthétisées par la Figure I.10, les différences entre elles, proviennent du mode de calcul des paramètres du reconstituteur en fonction du contexte choisi (cas continu ou discret, déterministe ou stochastique).

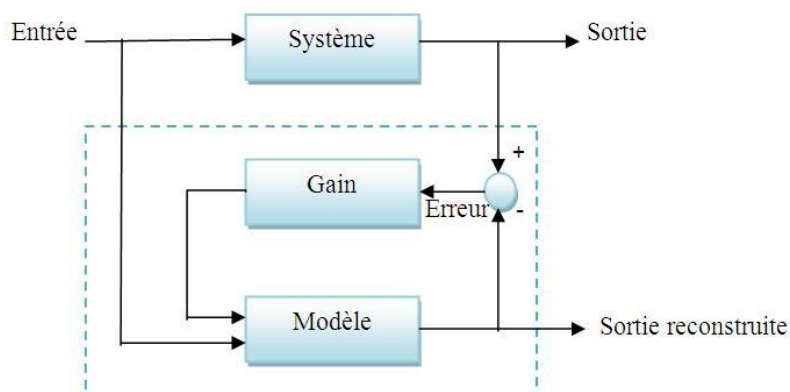


Figure I.10 : Principe général d'une observation d'état

c) Espace de parité

Les équations du modèle sont projetées dans un espace particulier appelé espace de parité, permettant ainsi d'éliminer les inconnues à l'aide de redondances [Cho, 80]. Les équations projetées ne font intervenir que des variables mesurables (les entrées et les sorties du système) sur une fenêtre d'estimation. Ces équations s'appellent des relations de redondance analytique

(RRA). L'idée est de tester la cohérence des mesures par rapport à leurs estimations données par le modèle (on parle de consistance des mesures, de leur parité) Figure I.11.

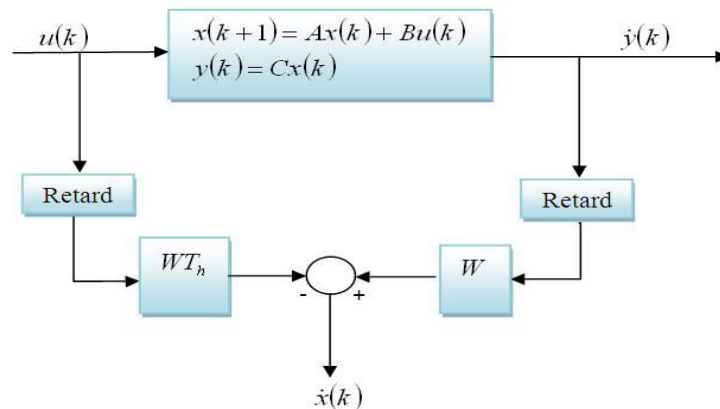


Figure I.11 : Diagnostic par espace de parité

I.3.2 Les méthodes à base de connaissances et de données

Ces méthodes s'appliquent dans les situations où la modélisation des mécanismes reliant les causes des défaillances n'est pas techniquement possible. Elles sont basées sur le retour d'expérience. Parmi ces méthodes on trouve la reconnaissance des formes, les réseaux de neurones, la logique floue et les systèmes experts ou à base de connaissances.

a) Reconnaissance des formes

Parmi toutes les approches du diagnostic, la reconnaissance des formes est une technique de définition d'algorithmes permettant de classer des objets dont l'aspect a varié par rapport à un objet type. Il s'agit de définir à quelle forme type une forme observée ressemble le plus.

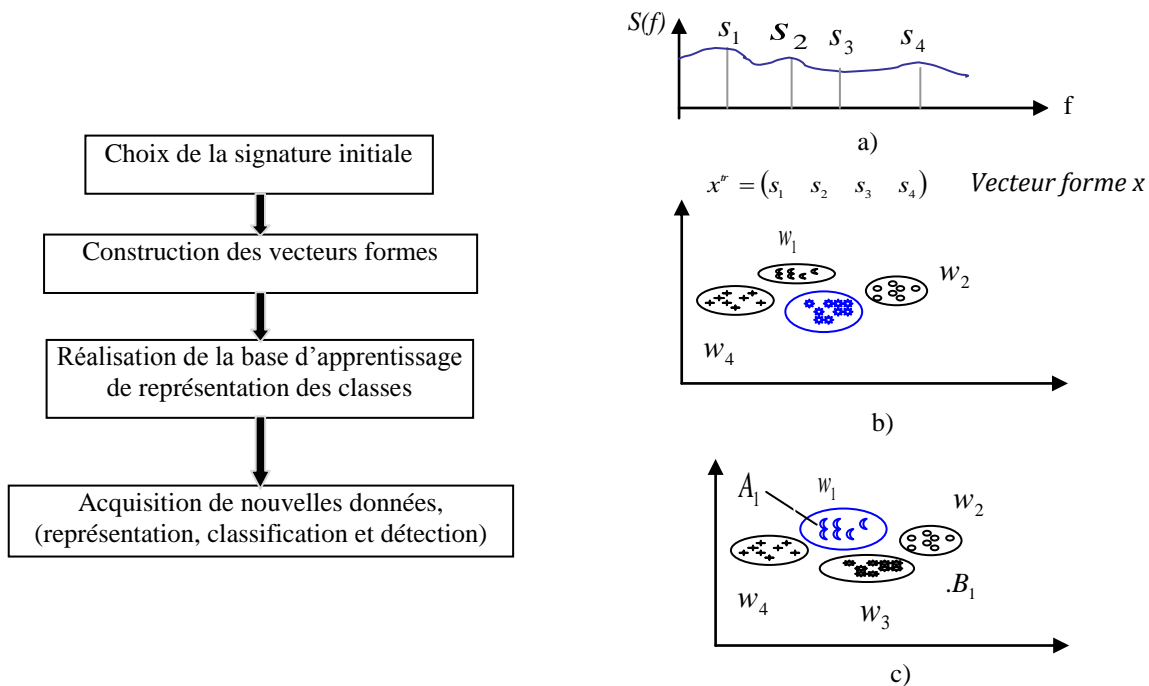


Figure I.12: Diagnostic basée sur la technique de la reconnaissance des formes

Dans ce cas, la forme est définie par un ensemble de 'd' paramètres appelés caractères qui sont les composants du vecteur forme x; les prototypes sont les points représentatifs du

vecteur forme dans l'espace à ' d ' dimensions. Dans un problème de reconnaissance des formes on suppose généralement que les formes x à classer appartiennent à M catégories bien déterminées appelées classes et notées w_1, w_2, \dots, w_m . Compte tenu du bruit de mesure les vecteurs appartenant à une classe donnée w_i forment une zone particulière dans l'espace de dimension d . Le problème de la reconnaissance des formes consiste à déterminer les formes séparant les M classes dans le but de pouvoir ultérieurement classer une nouvelle forme x lors de l'opération de classification ou de discrimination [Ven, 03b].

La procédure de diagnostic basée sur les techniques de reconnaissance des formes met en œuvre quatre étapes principales qui sont résumées sur la (figure I.12).

Dans ce cas, la dernière phase correspond évidemment à l'exploitation du système de diagnostic.

D'autre part, après l'acquisition d'une nouvelle signature, on calcule la forme associée dans l'espace de présentation et on procède à sa classification, chose qui se réalise sans difficulté (par exemple, il faudra prendre la décision de rejeter le point B ou de le caser dans une nouvelle classe).

b) *Système expert*

Un système expert doit fournir des réponses associées à une situation donnée sachant que la complexité du domaine étudié ne permet pas d'établir a priori toutes les configurations possibles des situations ; ainsi, un spécialiste du dépannage réalise son diagnostic à l'aide d'une description partielle de la situation. A partir de l'ensemble de symptômes dont il dispose, il peut en déduire toutes les conclusions possibles, élaborer de nouvelles hypothèses et approfondir son diagnostic en exploitant des informations supplémentaires collectées sur le système à diagnostiquer [Zwi 95]. Un système expert est constitué de plusieurs éléments principaux comme indiqué dans la Figure I.13.

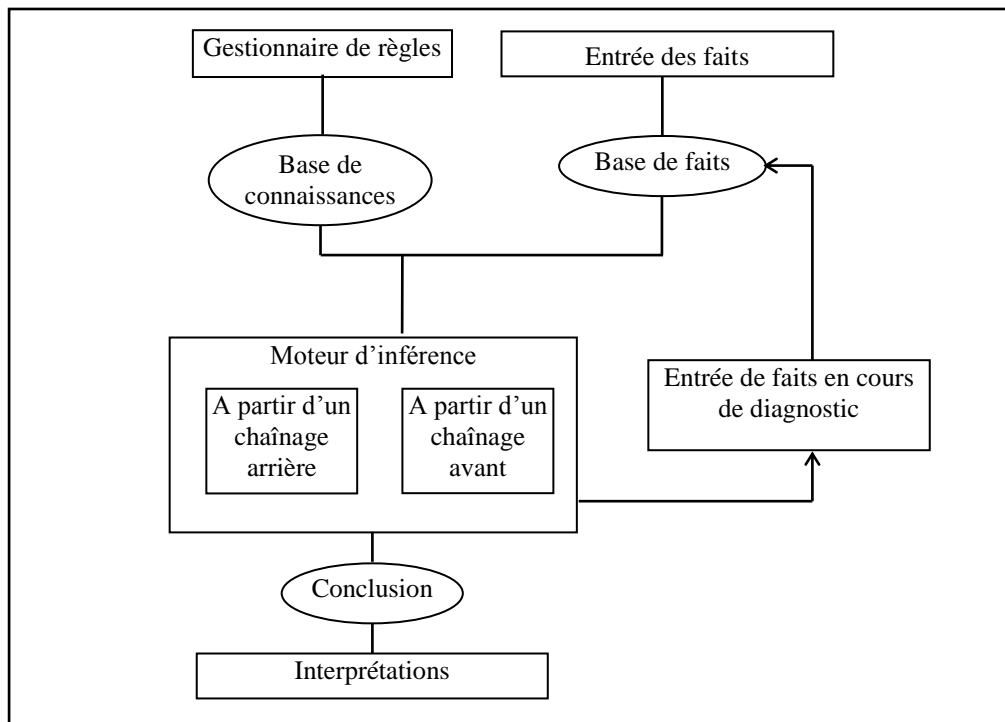


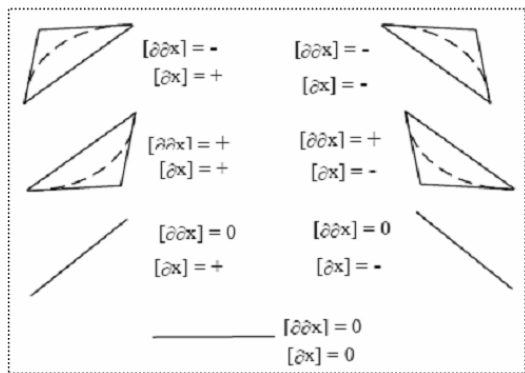
Figure I.13: Architecture générale d'un système expert de diagnostic.

- *La base de connaissances* : est souvent la partie stable du système, elle est immédiatement exécutée dès l'appel du système et c'est elle qui rassemble les connaissances nécessaires à la résolution des problèmes du domaine.
- *La base de faits* : Les faits sont les objets élémentaires de l'application ; la base de faits qui au départ contient les informations initiales relatives au domaine s'enrichit progressivement de conclusions ou de nouveaux faits déduits par le moteur d'inférence.
- *Le moteur d'inférence* : exploite de façon indépendante les informations contenues dans la base de faits et la base de connaissances pour fournir une solution au problème donné. Par ailleurs, le moteur d'inférence produit dans son mécanisme de raisonnement de nouveaux faits qui sont mémorisés dans la base de faits.

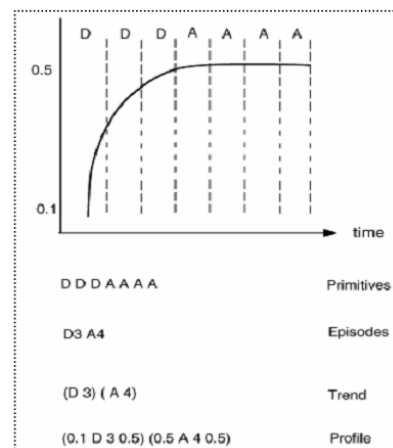
c) Les approches qualitatives

La représentation et l'analyse qualitative (symbolique) des signaux à travers l'information des dérivés du premier et second ordre représentent l'une des principales techniques pour le diagnostic et la supervision basées sur des données et sur l'abstraction de la dynamique continue. L'Analyse Qualitative de Tendances (AQT) des signaux est basée, en général, sur la segmentation temporelle des signaux en unités, appelées épisodes, en fonction de leur pente et de leur convexité et d'une classification de ces épisodes. [Mel, 01] énumère les besoins des systèmes de supervision qui motivent l'utilisation des méthodes d'analyse qualitative de tendances :

- L'incertitude, l'incomplétude et l'hétérogénéité (différentes échelles de temps par exemple) des données demandent un raisonnement qualitatif.
- Un raisonnement temporel est nécessaire non seulement sur des informations instantanées mais aussi sur le comportement historique.
- Face au volume de données présent dans les procédés complexes, il est préférable d'adopter une représentation symbolique et adaptée au système de surveillance.
- La représentation de la connaissance doit être adaptée à l'interprétation de l'opérateur humain.



(a) Ensemble des épisodes triangulaires [Che, 90]



(b) Classification selon [Jan, 91]

Figure I.14 : Les épisodes et la classification.

Le concept d'*épisode* a été introduit par [Wil, 86], [Che, 90]. Ils formalisent le concept de *tendance* comme une séquence d'épisodes caractérisés par le signe de la dérivée de premier et second ordre. Des *épisodes triangulaires* sont considérés comme *primitives* géométriques utilisées pour la modélisation des tendances comme le montre la Figure 14(a) : ce sont des

intervalles de temps pendant lesquels les dérivées du premier et second ordre ne changent pas. Des *épisodes trapézoïdaux* sont définis en fonction de la constance de la valeur qualitative de la dérivée du second ordre. [Jan, 91] proposent la description qualitative des signaux à travers une classification hiérarchique illustrée sur la Figure I.14(b): primitives, épisodes, tendances et profils. (En anglais : *primitives, episodes, trends and profiles*).

Relativement simple conceptuellement, l'AQT se trouve alourdie par certaines contraintes. L'observation des dérivées des signaux s'avère délicate. L'AQT implique la mise en œuvre d'un étage de filtrage de bruits et de traitement des discontinuités ou même la suppression d'informations d'ordre supérieur (dérivées). Pour le suivi et le diagnostic, le contexte temporel des tendances est fondamental. Le synchronisme ou l'alignement temporel entre le dictionnaire (signature) de tendances et les observations n'est pas facile à obtenir et des écarts faibles peuvent induire des erreurs d'interprétation et d'inférence au niveau du diagnostic. Des techniques DTW (*Dynamic Time Warping*) sont souvent utilisées pour résoudre ce problème [Kas, 98], [Col, 02].

D'autres travaux fondés sur l'analyse qualitative de tendances (AQT) sont proposés pour le suivi, le diagnostic et la supervision des procédés complexes. La classification de tendances pour le diagnostic de procédés chimiques à travers une inférence floue est proposée par [Das, 03]. Une matrice de similarité des primitives quantifie la mesure de distance (indices de similarité) (SI_j) entre l'observation et la signature à travers une inférence floue qui permet une robustesse de l'identification de tendance. L'indice de similarité SI_j considère non seulement la séquence des primitives mais aussi l'intervalle de temps ($\Delta t_i = t_{ui+1} - t_{ui}$). Un module de connaissance à base de règles s'appuie sur les indices des variables pour raisonner sur les hypothèses des défauts et les classer en fonction de l'indice de confiance globale (CI_j).

I.4 Les étapes de diagnostic

La performance et la fiabilité de l'ensemble des moyens de commande et de contrôle sont liées à la qualité des systèmes de mesures. Toute défaillance de l'instrumentation conduit à la génération d'informations erronées. La validation de mesures qui permet de s'assurer de la cohérence des informations acquises constitue donc une étape essentielle qui doit précéder toute tentative de conduite rationnelle. Alors l'image de la cohérence entre les signaux mesurés du système et ceux du modèle est donnée par les caractéristiques statistiques (méthodes basées sur un modèle du système) ou spectrales (méthodes basées sur un modèle du signal) du résidu. Pour qu'un signal généré à partir des entrées et sorties d'un système soit un résidu, il faut qu'il soit affecté par un sous-ensemble de défauts.

L'une des techniques permettant de s'assurer de la crédibilité d'une mesure consiste à créer une redondance d'informations ; celle-ci peut être obtenue en utilisant, par exemple, les relations structurellement exactes de bilan matière ou de bilan énergie. Ce type de redondance est qualifié d'*analytique* ou *fonctionnelle*, contrairement à la redondance *matérielle* obtenue en multipliant les capteurs mesurant une même grandeur.

I.4.1 Génération de résidus :

La première étape d'un système de surveillance à base de modèles consiste à générer des indicateurs de défauts (Figure I.15). Ils contiennent des informations sur les anomalies ou dysfonctionnements du système à surveiller. Le principe est de mesurer l'écart entre les mesures des signaux du procédé, capteurs ou actionneurs, et la valeur théorique fournie par le modèle dans des conditions de fonctionnement nominal. La génération de résidus est un problème crucial pour les systèmes de diagnostic. En effet, de la structure du système de résidus engendré dépendra la robustesse de la localisation.

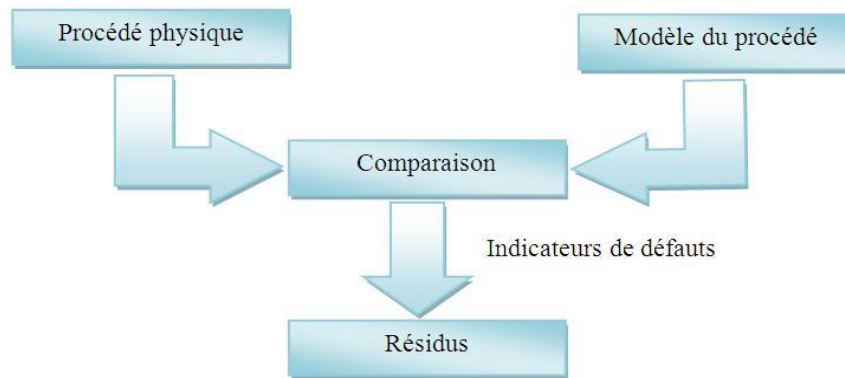


Figure I.15 : L'étape de génération de résidus

Un générateur de résidu $Q(t)$ est un système qui filtre les entrées $U(t)$ et sorties $Y(t)$ du système à surveiller et génère un signal appelé résidu [Nyb, 97].

$$R(t) = Q(t) \begin{pmatrix} U(t) \\ Y(t) \end{pmatrix} \quad (\text{I.5})$$

C'est sur l'étude de ce signal particulier que reposent les procédures de diagnostic [Jon, 73]; [Fra, 90]; [Pat, 91]. Généralement, lorsqu'un modèle est utilisé, seulement deux caractéristiques statistiques sont prises en compte pour caractériser le résidu : sa moyenne et son écart type. En pratique, on génère des résidus ayant une moyenne nulle en fonctionnement normal et différents de zéro en fonctionnement défaillant. De façon plus générale, on cherche toujours à calculer un vecteur de résidus $r(t)$ ayant les propriétés suivantes :

- $r(t) = 0$ quand $f(t) = 0$.
- $r(t) \neq 0$ quand $f(t) \neq 0$ pour la détection du défaut.
- $r_i(t) \neq 0$ et $r_j(t) = 0$ pour $j \neq i$ quand $f_i(t) \neq 0$ et $f_j(t) = 0$ pour la localisation du défaut $f_i(t)$.
- $\lim_{t \rightarrow \infty} (f_i(t) - r_i(t)) = 0$ pour l'identification du défaut $f_i(t)$.

I.4.2 La détection de défaut

Etant la deuxième étape de la fonction de diagnostic, la détection permet de déterminer la présence ou non d'un défaut sur le procédé. Elle est aussi appelée alarme globale. Nous nous intéresserons aux méthodes de détection à base de résidus. Ces derniers ont une valeur théorique nulle pour un système idéal en l'absence de défaut (pas d'incertitude de modèle, ni de bruits de mesure), et non nul dans le cas contraire. La principale difficulté de la détection réside dans le calcul du seuil des résidus. Un seuil trop grand risque d'engendrer une non-détection. Alors qu'un seuil trop petit entraînera des fausses alarmes. La problématique est donc de trouver un seuil optimal qui constituera le compromis idéal entre le taux de fausses alarmes et le taux de non-détection. L'évaluation des résidus consiste donc à optimiser le problème de seuillage des résidus. L'approche classique consiste à établir des fonctions de décision selon la nature des résidus :

- Si l'on cherche à déterminer la présence de bruits, on choisira l'analyse par tests statistiques.
- Si l'on considère la sensibilité des résidus aux défauts, l'évaluation des résidus r deviendra un problème de seuillage, i.e. la détermination d'un vecteur de seuils ε tel que :

$$\exists i : r_i(t) \geq \varepsilon_i \text{ pour un système en défaut,}$$

$$\forall i : r_i(t) < \varepsilon_i \text{ pour un système sans défaut.}$$

Concernant les tests statistiques, on peut citer :

- Le test du maximum de vraisemblance ou test GLR (Generalized Likelihood Ratio) introduit par Willsky et Jones [Wil, 76b] qui prend en compte les perturbations stochastiques (test de blancheur de bruit),
- Le test de Page-Hinkley qui teste la valeur moyenne du résidu sur une fenêtre de détection par rapport à un seuil prédéfini,
- Le filtre de décorrélation afin de supprimer l'influence des incertitudes de modèle sur le résidu [Bor, 90].

Concernant les problèmes de seuillage, les premiers travaux ont porté sur l'élaboration de seuils fixes, indépendants du temps et des entrées du système. Par exemple, Walker et Gai [Wal, 79] l'on définit à l'aide de la théorie de Markov.

Remarquons que l'effet des perturbations extérieures peut rendre inadaptée la détection avec un seuil de valeur constante prédéfinie. De même, une incertitude sur la mesure peut entraîner un résidu au-delà de la valeur limite fixée et donc déclencher une fausse alarme.

Ainsi, Emami-Naeni et al. [Ema, 88] ont défini la notion de seuil adaptatif, robuste contre les incertitudes de modèles. De même, Patton et Clark ont proposés d'adapter les seuils de décision en utilisant des fonctions déterministes : ils ont défini un seuil de détection adaptatif en fonction des entrées du système [Pat, 89]. L'idée consiste à définir des bornes inférieures et supérieures pour le résidu en fonction des bornes des incertitudes des paramètres et des bruits de mesure. Les seuils sont ainsi fonctions des modes opératoires. On recense d'autres contributions dans ce domaine [Din, 91]; [Wei, 88]; [Sau, 96].

I.4.3 La localisation de défaut

La troisième étape de la fonction de diagnostic est celle de la localisation. Elle a pour but de remonter à l'origine du défaut détecté. Les méthodes de génération de résidus sont nombreuses et leur application aux systèmes de surveillance dépend du type de modèle considéré (temporel, fréquentiel, linéaire ou non linéaire, dynamique ou statique...) et des informations disponibles sur le système. Néanmoins, quelle que soit la méthode utilisée (estimation paramétrique ou d'état, espace de parité...), les résidus générés sont généralement associés à des propriétés structurelles différentes. Etant donné que la localisation s'appuie sur la structure et la connectivité du système étudié, ainsi, c'est à partir de cette considération que l'on peut aborder la problématique de la localisation. Ceci conduit à introduire les notions de matrice d'incidence et de signatures de défauts.

La signature de la panne est associée à la structure du système de résidus par l'intermédiaire de la matrice d'incidence. Soit R un ensemble des relations de redondance et E l'ensemble des variables mesurées du procédé étudié. La matrice d'incidence M représente la relation suivante :

$$M: R * E \rightarrow \{0, 1\}^{n*m}$$

$$(r_i, e_j) \rightarrow M(i, j)$$

Avec :

$$M(i, j) = 1 \text{ si } r_j \text{ apparaît dans la relation de redondance analytique } e_i,$$

$$M(i, j) = 0 \text{ sinon.}$$

La matrice d'incidence est donc binaire, chaque colonne correspond à une variable et chaque ligne correspond à une composante du vecteur de résidu :

- un 1 dans une position donnée signifie que le résidu associé est directement influencé par la variable *correspondante*,
- un 0 signifient que la variable donnée n'entre pas dans le calcul du résidu.

Ainsi, pour des conditions idéales (pas de dispersion des valeurs des paramètres ou de modification de la structure du système...), chaque colonne de la matrice d'incidence correspondra à la signature d'une défaillance particulière. Afin de rendre, respectivement, chaque défaut détectable et unique, toutes les colonnes doivent contenir au moins un 1 et les

signatures associées doivent être uniques. Lorsque le problème de modélisation s'apparente à une estimation d'état, les résidus peuvent être soit structurés, soit directionnels. Pour améliorer la localisation des défauts, on peut utiliser des sous-ensembles de résidus. Chaque sous-ensemble sera sensible à un défaut ou un ensemble de défauts et insensible aux autres défauts. Clark [Cla, 75] a proposé deux schémas de localisation :

- Le schéma SOS (Simplified Observer Scheme), détecte parfaitement des défauts simultanés mais ne permet pas la localisation de chacun des défauts,
- Le schéma DOS (Dedicated Observer Scheme), dont la structure permet la détection de défauts simultanés mais la localisation n'est effective que pour des défauts non simultanés.

Frank [Fra, 87] a proposé un schéma de détection et de localisation de défauts simultanés: le schéma GOS (Generalized Observer Scheme). Ces schémas ont par la suite été complétés par la prise en compte d'un résidu supplémentaire découplant parfaitement les incertitudes structurées. On parlera alors de structures DOS ou GOS augmentées dont les différentes aptitudes sont discutées dans les travaux de Courtine [Cou, 97]. Van Schrik [Van, 96] a élaboré une comparaison des différentes structures en indiquant leurs conditions de mise en œuvre.

Néanmoins, les différents schémas cités plus haut ne peuvent être élaborés dans tous les cas et notamment lorsqu'on n'aborde pas le diagnostic par le biais de l'estimation d'état. Les dimensions de la matrice d'incidence sont déterminées à partir du nombre de capteurs et du nombre de résidus engendrés par la méthode choisie. Cette matrice est donc imposée par le système et ses instruments d'observation. Pour reprendre les termes utilisés par Gertler, on qualifie cette matrice de :

- *Non-localisante* si deux signatures de pannes sont identiques,
- *Déterministiquement localisant* (traduction littérale de l'anglais *deterministically isolable*) si toutes les signatures de pannes sont différentes, mais lorsqu'on modifie une signature en changeant un 1 par un 0, on retrouve une autre signature déjà existante,
- *Statistiquement localisante* (traduction littérale de l'anglais *statistically isolable*) si toutes les signatures sont différentes et ne peuvent être identiques même en changeant un 1 par un 0.

Dans la littérature, on parlera également d'isolation faible dans le cas de structures *déterministiquement localisantes*. En effet, la présence de perturbations pouvant fausser l'étape de détection, l'algorithme de localisation devient plus robuste pour des signatures de défauts différentes, même lorsque ces dernières sont dégradées (modification d'un 1 par un 0). Dans le cas de structures *statistiquement localisante*, on parlera d'isolation forte.

Il est à noter que certains auteurs se sont penchés sur l'étude du positionnement des capteurs dans le but d'optimiser la structure de la matrice d'incidence, i.e. de tendre vers une structure *statistiquement localisante*. Des exemples d'algorithmes peuvent être trouvés grâce aux travaux de Ragot [Rag, 92] et de Carpentier et Litwak [Car, 96].

I.4.4 L'identification de défaut

Dans le cas d'un changement de comportement lent dû au vieillissement et à l'usure du système, l'identification permet de surveiller la dégradation de ce dernier. De plus elle comprend une procédure visant à déterminer la cause d'un défaut, c'est à dire son origine. Ainsi, une fois le fonctionnement incorrect du système constaté, il est primordial d'agir de façon à maintenir les performances souhaitées ou à limiter les dégradations sur le système réel. Cette prise de décision permet le choix entre plusieurs possibilités telles qu'arrêter le système pour faire de la maintenance ou accepter un fonctionnement dégradé. Il peut encore s'agir, quand cela est possible, de reconfigurer ou de réorganiser le système [Bea, 71]. Le rôle de la reconfiguration est de s'affranchir des conséquences du défaut pour conserver les performances initiales lorsque cela est possible ou encore d'assurer un fonctionnement dégradé du système si celui-ci est tolérable. Il est important que le défaut soit identifié avec

exactitude (étendue, amplitude, type, cause) afin de permettre sa compensation éventuelle. La reconfiguration peut porter sur le système de régulation (ou une partie de celui-ci), sur la structure de la loi de commande ou sur le processus physique (en présence de redondance matérielle, on peut basculer sur les éléments ayant un fonctionnement correct).

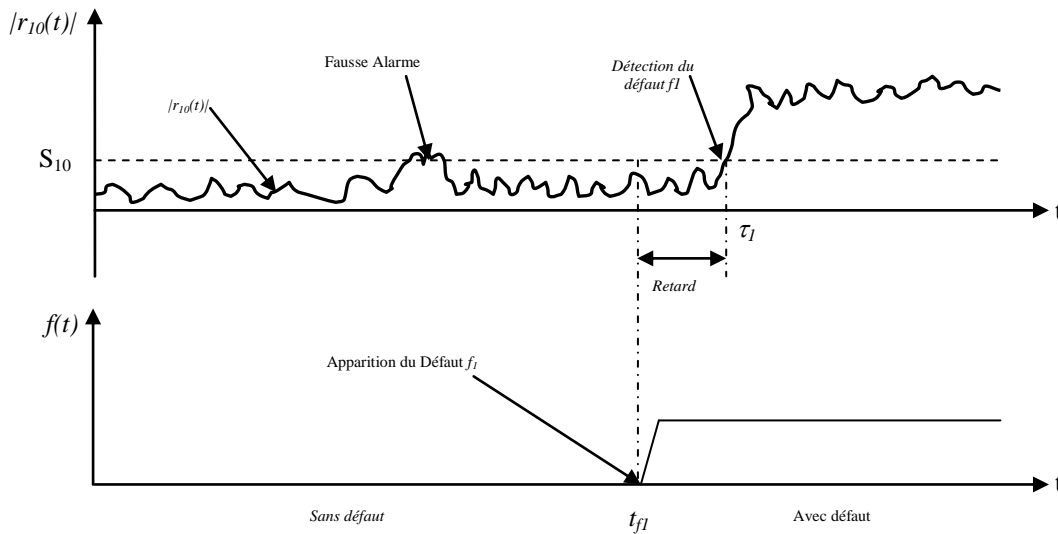


Figure I.16: Détection et identification de défaut.

La figure I.16 illustre les principales étapes de détection et d'identification d'un défaut. Le résidu $r_{10}(t)$ est fourni à partir de l'une des méthodes utilisées pour la génération des résidus comme vue précédemment, c'est une source d'information sur les défauts pour un traitement ultérieur. La détection de défauts est basée sur l'évaluation de l'ampleur des résidus. Chaque résidu $r_{k0}(t)$, $k=1, \dots, N$, (avec k le nombre de défaut) devrait normalement être proche de zéro dans le cas sans défaut, et loin de zéro dans le cas avec défaut. Ainsi, les défauts sont détectés en fonction du seuil S_{k0} admis des résidus (un résidu unique et une seule panne sont considérés pour la simplicité). L'analyse de résidu $r_{10}(t)$ fournit également une estimation de l'instant d'apparition τ_1 du défaut f_1 avec un certain *Retard*. Lorsqu'on utilise plusieurs signaux de résidus, l'estimation de cet instant est donnée par:

$$\tau = \min \{ \tau_k, k = 1, \dots, n \}. \quad (\text{I.6})$$

Les défauts sont détectés lorsque la valeur absolue du résidu $|r_{k0}(t)|$ dépasse le seuil S_{k0} :

$$\begin{cases} |r_{k0}(t)| \leq S_{k0} : \text{aucun défaut détecté à l'instant } t \\ |r_{k0}(t)| > S_{k0} : \text{un défaut est détecté à l'instant } t. \end{cases} \quad (\text{I.7})$$

La difficulté principale de cette évaluation est que la mesure des sorties du système $y_i(t)$ est généralement bruitée par des perturbations (par exemple, le bruit de mesure). Dans la pratique, en raison des incertitudes de modélisation et de perturbations, il est nécessaire d'affecter un seuil élevé S_{10} afin d'éviter les fausses alarmes. Cette augmentation du seuil peut engendrer la non-détection du défaut f_1 . Pour cette raison, il est nécessaire de choisir une technique adaptée pour fixer le seuil.

I.5 La structuration des résidus

La conception de tels résidus comporte deux étapes. Tout d'abord, il convient de définir les sensibilités (et insensibilités) désirées des résidus par rapport aux différents défauts à (ou à ne

pas) détecter. Puis, en fonction de ces contraintes, il faut définir le générateur de résidus adéquat.

Définition : La structure d'un résidu r_i par rapport à un ensemble de défauts F de dimension n_f est le mot binaire M_{r_i} composé de n_f bits $(M_{i,j})$ positionnés de la manière suivante :

– $M_{i,j} = 1$ si le résidu r_i est affecté par le $j^{\text{ème}}$ élément de $\{F\}$

– $M_{i,j} = 0$ si le résidu r_i n'est pas affecté par le $j^{\text{ème}}$ élément de $\{F\}$

Les résidus structurés sont conçus de manière à être sensibles à un sous-ensemble de défauts et robustes par rapport aux défauts restants. Lorsqu'un défaut apparaît, seul un sous-ensemble de résidus réagit.

Les informations de sensibilité et de robustesse souhaitées pour les résidus sont représentées dans une table binaire, appelée *table des signatures* théoriques ou encore *table de codage*. Celle-ci est construite de la façon suivante : lorsque le $i^{\text{ème}}$ résidu est sensible (respectivement robuste) au $j^{\text{ème}}$ défaut, alors la valeur binaire 1 (respectivement 0) est affectée à la ligne et à la colonne correspondante. L'ensemble des résidus structurés est construit en respectant la procédure suivante :

1. Fixer la table des signatures théoriques que l'on souhaite obtenir.
2. Construire un vecteur de résidus ayant les propriétés désirées.
3. Si l'obtention d'un vecteur de résidus ayant les propriétés imposées par la table des signatures théoriques n'est pas possible, alors de nouvelles spécifications de sensibilité et robustesse doivent être proposées.

Lorsque la table des signatures théoriques est construite, une procédure de détection, appliquée à chaque résidu, permet d'obtenir la *signature réelle* des résidus à un instant donné. Si cette signature est nulle, alors aucun résidu ne permet de détecter le défaut. Le système sera donc déclaré sain. Si un défaut survient, au moins un résidu le détectera, la signature réelle deviendra non nulle. La procédure de localisation consistera alors à comparer la signature booléenne du résidu aux signatures de la table des signatures théoriques.

Définition : Un défaut est structurellement localisable si toutes les colonnes de la table de signatures théoriques sont différentes.

La table de signatures théoriques doit être proposée d'une façon à ce que le vecteur de résidus structurés correspondant soit calculable et que les propriétés d'isolabilité soient les plus intéressantes possibles. La qualité de l'isolabilité est définie comme suit.

Définition : Un défaut est localisable d'ordre k si sa distance de Hamming par rapport à la signature du défaut le plus proche est de k .

Nous allons illustrer ce propos à travers un exemple simple.

Exemple: Supposons que nous avons généré trois résidus r_1 , r_2 et r_3 à partir d'un système soumis à trois défauts f_1 , f_2 et f_3 . Le Tableau 1 représente quatre ensembles de signatures théoriques ayant des propriétés d'isolabilité différentes.

Tableau I.1: Table de codage pour trois résidus pour trois défauts

a)	f_1	f_2	f_3	b)	f_1	f_2	f_3	c)	f_1	f_2	f_3	d)	f_1	f_2	f_3
r_1	1	1	0	r_1	1	1	0	r_1	1	1	0	r_1	1	0	0
r_2	1	1	1	r_2	1	0	1	r_2	1	0	1	r_2	0	1	0
r_3	0	0	1	r_3	0	0	1	r_3	0	1	1	r_3	0	0	1

- La table (a) illustre le cas de deux défauts non isolables (f_1 et f_2 possèdent la même signature).
- La table (b) illustre l'exemple où les défauts sont isolables d'ordre 1 (un seul bit est différent entre la signature de f_1 et celle de f_2).
- Les tables (c) et (d) montrent des défauts isolables d'ordre 2.

Malgré un indice d'isolabilité identique entre les tables (c) et (d), la table (d) sera la plus simple à traiter. D'une manière générale, lorsque deux tables de signatures théoriques ont le même ordre d'isolabilité, la table retenue est celle qui contient le plus de zéros.

I.6 Conclusion

A travers ce chapitre nous avons présentés de façon générale et succincte les différentes méthodes de diagnostic rencontrées dans la littérature. Les méthodes de diagnostic de défaillances et de dégradation utilisées dans les différents secteurs industriels sont très variées et tiennent compte de la spécificité du matériel utilisé dans les procédés industriels. Diagnostiquer un système ou un processus de production, n'est pas une opération simple, pour cela, on a recours à des outils d'aide au diagnostic complexe, qui exigent plusieurs informations.

Dans le cadre de notre thèse nous nous sommes intéressés aux méthodes de diagnostic basées sur des modèles obtenus par des techniques de l'intelligence artificielle. Le diagnostic de défauts se fait en deux étapes. Dans la première étape le comportement dynamique du système réel en fonctionnement sain (sans défauts) aussi complexe qu'il soit, linéaire ou non linéaire, peut être émulé. Dans la deuxième étape les résidus générés lors de l'occurrence d'une défaillance dans le système réel, sont analysés et traités par différentes techniques dans le but de reconnaître et de déterminer quel type de défaut est survenu. Les capacités offertes par les systèmes à base de connaissances (*BC*) ne peuvent que confirmer l'intérêt de leur utilisation en particulier dans les domaines touchant au secteur industriel.

Le chapitre suivant sera consacré à l'introduction de quelques techniques de l'intelligence artificielle, ainsi que leur utilisation dans le domaine du diagnostic de défauts. Nous nous intéresserons particulièrement aux réseaux de neurones et la logique floue qui seront par suite utilisés dans notre contribution au diagnostic.

Chapitre II

Aperçu et application des Réseaux de Neurones, Logique Floue et Neuro-Flou dans le Diagnostic

Sommaire

II.1	Réseau de neurones	33
II.1.1	Introduction.....	33
II.1.2	Neurone biologique et neurone formel	34
II.1.3	Architecture des réseaux de neurones	37
II.1.4	Apprentissage des réseaux de neurones	40
II.1.5	Algorithme d'apprentissage	41
II.1.6	La classification par les réseaux de neurone	46
II.1.7	Quelques points essentiels à retenir	46
II.2	La logique floue	47
II.2.1	Introduction.....	47
II.2.2	L'ensemble flou.....	48
II.2.3	Système flou.....	48
II.2.4	Application de la logique floue.....	50
II.2.5	Logique floue et système expert	51
II.3	Les Réseaux Neuro-Flou	51
II.4	Application des RN, LF, et NF dans le diagnostic.....	54
II.5	Conclusion.....	57

II.1 Réseau de neurones

II.1.1 Introduction:

L'objectif principal de ce chapitre est de présenter des développements concernant l'application des réseaux de neurone et la logique floue au diagnostic de défauts.

Nous allons adopter la méthode de diagnostic à base de modèle de type boîte noire (réseaux de neurones MLP avec apprentissage supervisé).

Notre travail se divise en deux étapes :

- Elaboration d'un modèle neuronal pour la génération des résidus,
- Conception de classifieurs pour classer les fautes.

Malgré la constante augmentation de la puissance des calculateurs, malgré les approches théoriques de plus en plus sophistiquées, un certain nombre de tâches résistent encore aux algorithmes et aux méthodes classiques de traitement des signaux et des données. Ces tâches relèvent typiquement du traitement, en temps réel, de très grands flots de données souvent multidimensionnelles et arrivant à des cadences élevées. Le grand nombre de données, leur variabilité, le fait qu'elles ne répondent pas à des modèles physiques connus nous laissent souvent démunis devant des tâches de caractérisation, de reconnaissance et de prise de décision.

L'objectif de ce chapitre est de rappeler les définitions de base relatives aux réseaux de neurones ainsi que les propriétés mathématiques de certains d'entre eux. Ensuite, Nous parlerons de leur apprentissage. Un réseau de neurones est une fonction paramétrée qui est la composition d'opérateurs mathématiques simples appelés neurones formels (ou plus simplement neurones). Afin de préciser ces notions, nous commencerons par présenter les définitions relatives aux neurones avant de détailler différentes architectures de réseaux de neurones.

Les premiers à proposer un modèle neuronal sont deux bio-physiciens de Chicago, McCulloch et Pitts, qui inventent en 1943 le premier neurone formel qui portera leurs noms (neurone de McCulloch-Pitts) [Mac, 43]. Quelques années plus tard, en 1949, Hebb [Heb, 49] propose une formulation du mécanisme d'apprentissage, sous la forme d'une règle de modification des connexions synaptiques (règle de Hebb). Cette règle, basée sur des données biologiques, modélise le fait que si des neurones, de part et d'autre d'une synapse, sont activés de façon synchrone et répétée, la force de la connexion synaptique s'accroît.

Le premier réseau de neurones artificiels apparaît en 1958, grâce aux travaux de Rosenblatt qui conçoit le Perceptron [Ros, 58b]. Le Perceptron est inspiré du système visuel (en terme d'architecture neuro-biologique) et possède une couche de neurones d'entrée ("perceptive") ainsi qu'une couche de neurones de sortie ("décisionnelle"). Ce réseau parvient à identifier des formes simples et à calculer certaines fonctions logiques. Il constitue le premier système artificiel présentant une faculté jusque là réservée aux êtres vivants : la capacité d'apprendre par l'expérience.

Malgré tout l'enthousiasme que soulève le travail de Rosenblatt dans le début des années 60 [Ros, 62], la fin de cette décennie sera marqué en 1969, par une critique violente du Perceptron par Minsky et Papert [Min, 69]. Ils montrent dans un livre (« Perceptrons ») toutes les limites de ce modèle, et soulèvent particulièrement l'incapacité du Perceptron à résoudre les problèmes non linéairement séparables, tels que le problème du XOR (OU exclusif). Il s'en suivra alors, face à la déception, une période noire d'une quinzaine d'années dans le domaine des réseaux de neurones artificiels.

Il faudra attendre le début des années 80 et le génie de Hopfield pour que l'intérêt pour ce domaine soit de nouveau présent [Hop, 80]. En effet, Hopfield démontre en 1982 tout l'intérêt d'utiliser des réseaux récurrents (dits "feed-back") pour la compréhension et la modélisation des processus mnésiques [Hop, 82]. Les réseaux récurrents constituent alors la deuxième

grande classe de réseaux de neurones, avec les réseaux type perceptron (dits "feed-forward"). En parallèle des travaux de Hopfield [Hop, 84], Werbos [Wer, 89b] conçoit son algorithme de rétro-propagation de l'erreur, qui offre un mécanisme d'apprentissage pour les réseaux multicouches de type Perceptron (appelés MLP pour Multi-layer Perceptron), fournissant ainsi un moyen simple d'entraîner les neurones des couches cachées. Cet algorithme de "back-propagation" ne sera pourtant popularisé qu'en 1986 par Rumelhart [Rum, 86a].

II.1.2 Neurone biologique et neurone formel:

II.1.2.1 Structure d'un neurone biologique: On pense que le système nerveux compte plus de 1000 milliards de neurones interconnectés. Bien que les neurones ne soient pas tous identiques, leur forme et certaines caractéristiques permettent de les répartir en quelques grandes classes. En effet, il est aussi important de savoir, que les neurones n'ont pas tous un comportement similaire en fonction de leur position dans le cerveau. Avant de rentrer plus en avant dans les détails, examinons un neurone. La Figure II.1 représente un neurone biologique:

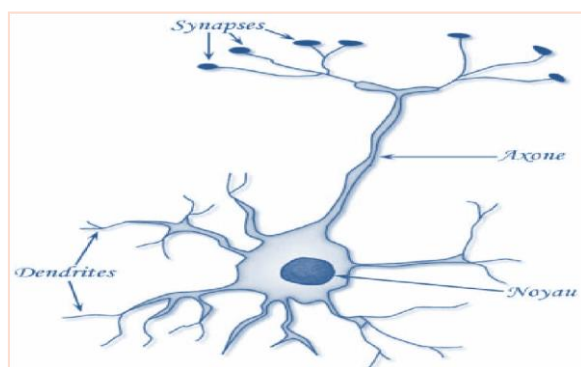


Figure II.1: Neurone Biologique.

On peut le décomposer en trois régions principales:

- *Le corps cellulaire:* Il contient le noyau du neurone ainsi que la machine biochimique nécessaire à la synthèse des enzymes. Ce corps cellulaire de forme sphérique ou pyramidale contient aussi les autres molécules essentielles à la vie de la cellule. Sa taille est de quelques microns de diamètre.
- *Les dendrites:* Ce sont de fines extensions tubulaires qui se ramifient autour du neurone et forment une sorte de vaste arborescence. Les signaux envoyés au neurone sont captés par les dendrites. Leur taille est de quelques dizaines de microns de longueur.
- *L'axone:* C'est le long de l'axone que les signaux partent du neurone. Contrairement aux dendrites qui se ramifient autour du neurone, l'axone est plus long et se ramifie à son extrémité ou il se connecte aux dendrites des autres neurones. Sa taille peut varier entre quelques millimètres à plusieurs mètres.
- *Synapse:* Une synapse est une jonction entre deux neurones, et généralement entre l'axone d'un neurone et une dendrite d'un autre neurone (mais il existe aussi des synapses axo-axonales par exemple), Figure II.2.

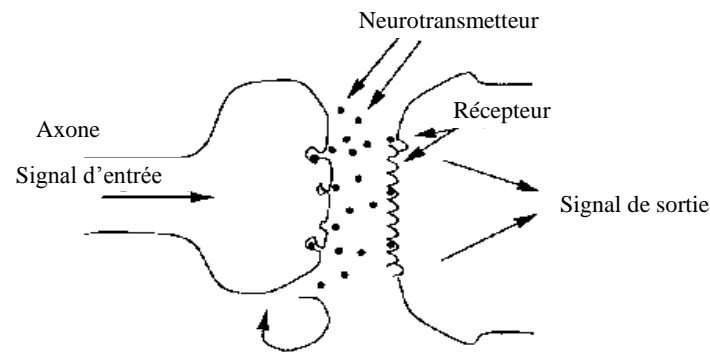


Figure II.2 : Une synapse

- *Fonctionnement:* Au point de vue fonctionnel, il faut considérer le neurone comme une entité polarisée, c'est-à-dire que l'information ne se transmet que dans un seul sens : des dendrites vers l'axone. Le neurone va donc recevoir des informations, venant d'autres neurones, grâce à ses dendrites. Il va ensuite y avoir sommation, au niveau du corps cellulaire, de toutes ces informations et via un potentiel d'action (un signal électrique) le résultat de l'analyse va transiter le long de l'axone jusqu'aux terminaisons synaptiques. A cet endroit, lors de l'arrivée du signal, des vésicules synaptiques vont venir fusionner avec la membrane cellulaire, ce qui va permettre la libération des neurotransmetteurs (médiateurs chimiques) dans la fente synaptique. Le signal électrique ne pouvant pas passer la synapse (dans le cas d'une synapse chimique), les neurotransmetteurs permettent donc le passage des informations, d'un neurone à un autre. Les neurotransmetteurs excitent (neurotransmetteurs excitateurs) ou inhibent (neurotransmetteurs inhibiteurs) le neurone suivant et peuvent ainsi générer ou interdire la propagation d'un nouvel influx nerveux. En effet, au niveau post-synaptique, sur la membrane dendritique, se trouvent des récepteurs pour les neurotransmetteurs. Suivant le type de neurotransmetteur et le type des récepteurs, l'excitabilité du neurone suivant va augmenter ou diminuer, ce qui fera se propager ou non l'information.

Les synapses possèdent une sorte de «mémoire» qui leur permet d'ajuster leur fonctionnement. En fonction de leur «histoire», c'est-à-dire de leur activation répétée ou non entre deux neurones, les connexions synaptiques vont donc se modifier. Ainsi, la synapse va faciliter ou non le passage des influx nerveux. Cette plasticité est à l'origine des mécanismes d'apprentissage.

II.1.2.2 Structure d'un Neurone formel: Le premier neurone formel est apparu en 1943 [Mac, 43]. On le doit à Mac Culloch et Pitts [Min, 56]. Voici un schéma de leur modèle de neurone formel :

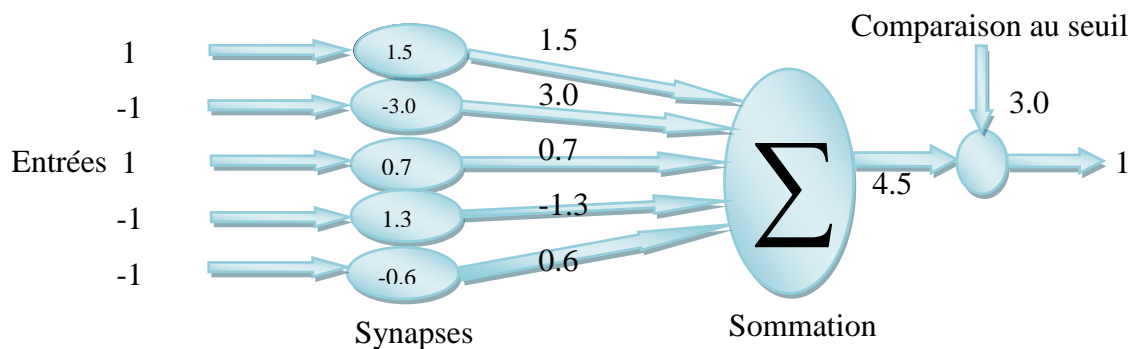


Figure II.3: Neurone de Mac Culloch et Pitts.

Le neurone formel est donc une modélisation mathématique qui reprend les principes du fonctionnement du neurone biologique, en particulier la sommation des entrées. Sachant qu'au niveau biologique, les synapses n'ont pas toutes la même «valeur» (les connexions entre les neurones étant plus ou moins fortes), les auteurs ont donc créé un algorithme qui pondère la somme de ses entrées par des poids synaptiques (coefficients de pondération). De plus, les 1 et les -1 en entrée sont là pour figurer une synapse excitatrice ou inhibitrice.

- *Interprétation mathématique:* D'un point de vue mathématique, le neurone formel peut être représenté de la manière suivante la Figure II.4.

Le neurone formel va calculer la somme de ses entrées (x_1, \dots, x_n), pondérées par les poids synaptiques (w_1, \dots, w_n), et la comparer à son seuil ou biais b . Si le résultat est supérieur au seuil, alors la valeur renvoyée est 1, sinon la valeur renvoyée est 0. D'où la formule (avec $f =$ fonction seuil):

$$y = f(n) = f\left(\sum_{j=1}^n w_j x_j - b\right) \tag{II.1}$$

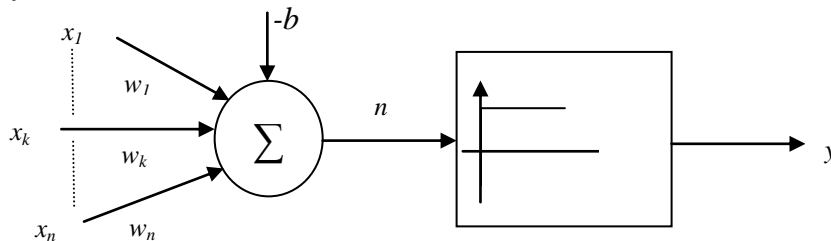
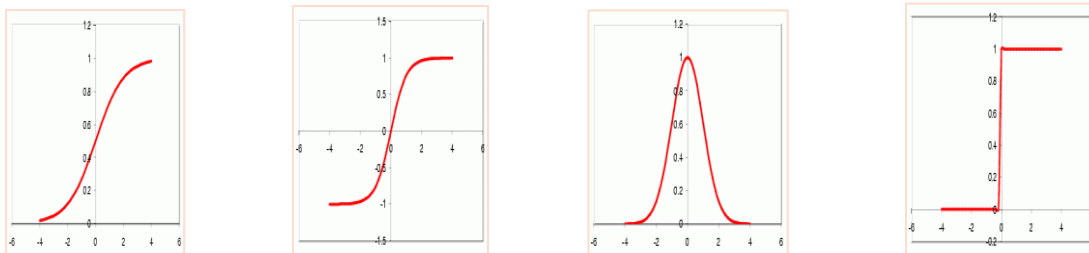


Figure II.4: Schéma d'un neurone formel.

- *Fonctions d'activation:* Dans sa première version, le neurone formel était donc implémenté avec une fonction à seuil, mais de nombreuses versions existent. Ainsi le neurone de McCulloch et Pitts a été généralisé de différentes manières, en choisissant d'autres fonctions d'activation, comme les fonctions linéaires, gaussienne ou des sigmoïdes par exemple (Figure II.5).



Sigmoïde standard

Tangente hyperbolique

La fonction Gaussienne

La fonction à seuil

Figure II.5 : Différentes formes des fonctions d'activation.

Tableau II.1: Différentes formes des fonctions d'activation et leur commande avec le logiciel Matlab.

Nom de la fonction	Relation d'entrée: n /sortie: a	Commande
Seuil	$\begin{cases} a = 0 & \text{si } n < 0 \\ a = 1 & \text{si } n \geq 0 \end{cases}$	<i>hardlim</i>
Seuil symétrique	$\begin{cases} a = -1 & \text{si } n < 0 \\ a = 1 & \text{si } n \geq 0 \end{cases}$	<i>hardlims</i>
Linéaire	$a = n$	<i>purelin</i>

Linéaire saturée	$\begin{cases} a = 0 & \text{si } n < 0 \\ a = n & \text{si } 0 \leq n \leq 1 \\ a = 1 & \text{si } n > 1 \end{cases}$	<i>satlin</i>
Linéaire saturée symétrique	$\begin{cases} a = -1 & \text{si } n < -1 \\ a = n & \text{si } -1 \leq n \leq 1 \\ a = 1 & \text{si } n > 1 \end{cases}$	<i>satlins</i>
Linéaire positive	$\begin{cases} a = 0 & \text{si } n < 0 \\ a = n & \text{si } n \geq 0 \end{cases}$	<i>poslin</i>
Sigmoïde	$a = \frac{1}{1 + \exp^{-n}}$	<i>logsig</i>
Tangente hyperbolique	$a = \frac{\exp^n - \exp^{-n}}{\exp^n + \exp^{-n}}$	<i>tansig</i>
Compétitive	$\begin{cases} a=1 & \text{si } n \text{ maximum} \\ a=0 & \text{autrement} \end{cases}$	<i>compet</i>

II.1.3 Architecture des réseaux de neurones:

Un RNA (Réseau de Neurones Artificiels) est un ensemble de neurones formels (d'unités de calcul simples, de nœuds processeurs) associés en couches (ou sous-groupes) et fonctionnant en parallèle. Dans un réseau, chaque sous-groupe fait un traitement indépendant des autres et transmet le résultat de son analyse au sous-groupe suivant. L'information donnée au réseau va donc se propager couche par couche, de la couche d'entrée à la couche de sortie, en passant soit par aucune, une ou plusieurs couches intermédiaires (dites couches cachées). Il est à noter qu'en fonction de l'algorithme d'apprentissage, il est aussi possible d'avoir une propagation de l'information à reculons ("back propagation"). Habituellement (excepté pour les couches d'entrée et de sortie), chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante. Les RNA ont la capacité de stocker de la connaissance empirique et de la rendre disponible à l'usage. Les habiletés de traitement (et donc la connaissance) du réseau vont être stockées dans les poids synaptiques, obtenus par des processus d'adaptation ou d'apprentissage. En ce sens, les RNA ressemblent donc au cerveau car non seulement, la connaissance est acquise au travers d'un apprentissage mais de plus, cette connaissance est stockée dans les connexions entre les entités, soit dans les poids synaptiques.

Ils existent différentes architectures de réseaux de neurones, leur classement est représenté par la figure II.6.

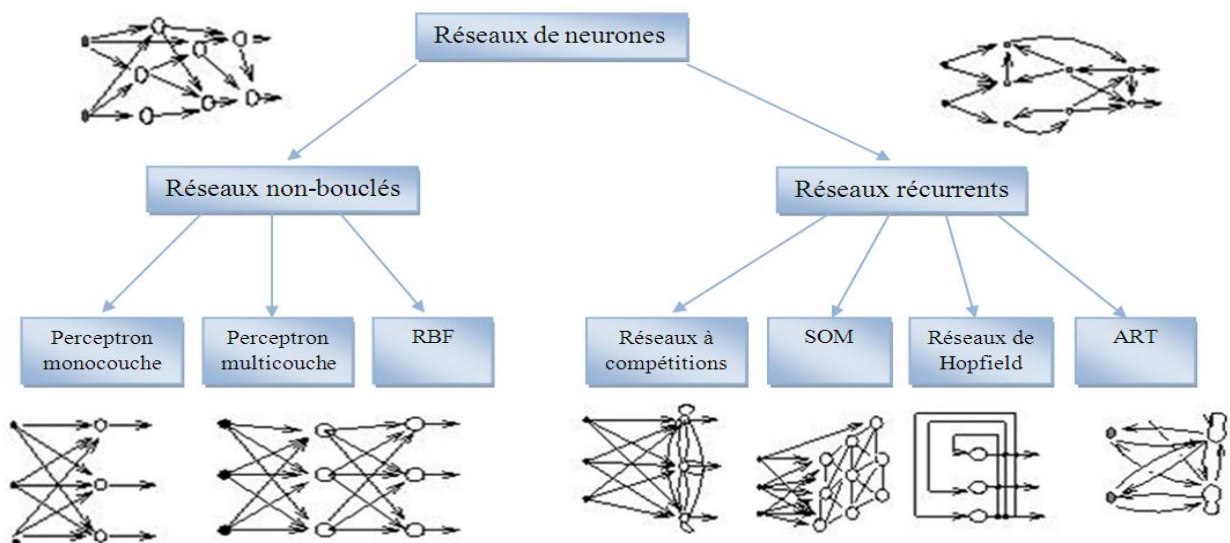


Figure II.6: Architectures des réseaux de neurones

- Les réseaux "FEED-FORWARD": Appelés aussi "réseaux de type Perceptron", sont des réseaux dans lesquels l'information se propage de couche en couche sans retour en arrière possible.

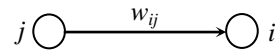
- Les *Perceptrons monocouche*: C'est historiquement le premier RNA, proposé par Rosenblatt. C'est un réseau simple, puisque il ne se compose que d'une couche d'entrée et d'une couche de sortie. Il est calqué, à la base, sur le système visuel et de ce fait, il a été conçu dans un but premier de reconnaissance des formes. Cependant, il peut aussi être utilisé pour faire de la classification et pour résoudre des opérations logiques simples (telle "ET" ou "OU"). Sa principale limite est qu'il ne peut résoudre que des problèmes linéairement séparables. Il suit généralement un apprentissage supervisé selon la règle de correction de l'erreur (ou selon la règle de Hebb).

La règle de Hebb: Cette règle permet de modifier la valeur des poids synaptiques en fonction de l'activité des unités qui les relient [Heb, 49]. Si deux unités connectées sont actives simultanément, le poids de leur connexion est augmenté ou diminué. R est une constante positive qui représente la force d'apprentissage (learning rate). $\Delta w_{ij} = R a_i a_j$

$$a_i = -1 \quad a_i = 1$$

$$a_j = -1 \quad \Delta W_{ij} = R \quad \Delta W_{ij} = -R$$

$$a_j = 1 \quad \Delta W_{ij} = -R \quad \Delta W_{ij} = R$$



La Figure II.7 illustre un réseau monocouche composé de k neurones organisés en sortie et connectés à des entrées pondérées par la matrices des paramètres $W = [w_{1k} \ w_{2k} \ \dots \ w_{Rk}]^T$, avec w_{Rk} le poids du neurone de l'entrée x_R .

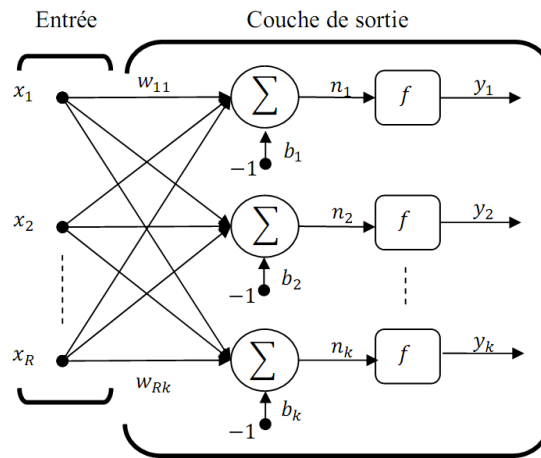


Figure II.7: Perceptrons mono couche.

Les entrées du neurone k correspondent au vecteur $X = [x_1 \ x_2 \ \dots \ x_R]^T$, alors que la sortie n_k de l'intégrateur est donnée par l'équation suivante :

$$n_k = \sum_{i=1}^R w_{ik} x_i - b_k \tag{II.2}$$

Que l'on peut aussi écrire sous forme matricielle :

$$n = w^T X - b \tag{II.3}$$

En remplaçant w^T par une matrice $W = w^T$, l'équation (II.2) devient :

$$n = WX - b \tag{II.4}$$

La sortie n_k correspond à une somme pondérée des entrées moins ce qu'on nomme le biais b_k du neurone. Le résultat n_k de la somme pondérée s'appelle le niveau d'activation du neurone.

Le biais b_k s'appelle aussi le seuil d'activation du neurone. La fonction d'activation f permet de définir l'état interne du neurone en fonction de son entrée totale, comme exemple on a la fonction seuil définie comme nous avons vue dans le Tableau I.1 :

$$y_n = f(n_k) = \begin{cases} = 0 & \text{si } n_k < 0 \\ = 1 & \text{si } n_k \geq 0 \end{cases} \quad (\text{II.5})$$

Considérons le cas le plus simple, à savoir lorsque $R=2$, c'est-à-dire lorsque les neurones ne sont reliés que de deux entrées. Dans ce cas, nous aurons $X = [x_1 \ x_2]^T$, $W = [w_{1k} \ w_{2k}]^T$, $b = [b_k]$, ou:

$$y_n = f(n_k) = \begin{cases} = 0 & \text{si } w_{1k}x_1 + w_{2k}x_2 - b_k < 0 \\ = 1 & \text{si } w_{1k}x_1 + w_{2k}x_2 - b_k \geq 0 \end{cases} \quad (\text{II.6})$$

Cette dernière équation nous indique clairement que la sortie du neurone peut prendre seulement deux valeurs distinctes selon le niveau d'activation du neurone : 0 lorsque ce dernier est strictement inférieur à 0 et +1 dans le cas contraire.

- *Le Perceptron Multicouche*: C'est une extension du précédent, avec une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante (excepté pour les couches d'entrée et de sortie) et il n'y a pas de connexions entre les cellules d'une même couche. Les fonctions d'activation utilisées dans ce type de réseaux sont principalement les fonctions à seuil ou sigmoïdes. Il peut résoudre des problèmes non linéairement séparables et des problèmes logiques plus compliqués, et notamment le problème du XOR. Il suit aussi un apprentissage supervisé selon la règle de correction de l'erreur.

Les réseaux de ce type possèdent trois sortes de couches:

- Des couches d'entrée avec R_i des unités d'entrée ;
- Des couches des neurones "C" cachées avec R_c , l unités. Les couches cachées démultiplient la puissance de calcul et permettent d'aborder des problèmes plus difficiles;
- Des couches des neurones "S" de sortie avec R_o unités de sortie.

On peut voir la structure d'un tel réseau multicouche suivant la Figure II.8, [Cha, 93].

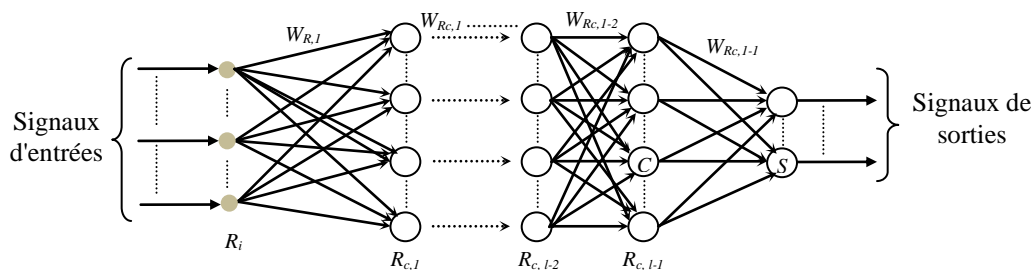


Figure II.8: Architecture d'un réseau multicouche.

- *Les réseaux à fonction radiale*: Ce sont les réseaux que l'on nomme aussi RBF ("Radial Basis Functions"). L'architecture est la même que pour les PMC (*Perceptron MultiCouche*) cependant, les fonctions de base utilisées ici sont des fonctions Gaussiennes. Les RBF seront donc employés dans les mêmes types de problèmes que les PMC à savoir, en classification et en approximation de fonctions. L'apprentissage le plus utilisé pour les RBF est le mode hybride et les règles sont soit, la règle de correction de l'erreur soit, la règle d'apprentissage par compétition.

- *Les réseaux "FEED-BACK"*: Appelés aussi "réseaux récurrents", ce sont des réseaux dans lesquels il y a retour en arrière de l'information.

- *Les cartes auto-organisatrices de Kohonen*: Ce sont des réseaux à apprentissage non-supervisé qui établissent une carte discrète, ordonnée typologiquement, en fonction de patterns d'entrée. Le réseau forme ainsi une sorte de treillis dont chaque nœud est un neurone associé à un vecteur de poids. La correspondance entre chaque vecteur de poids est calculée pour chaque entrée. Par la suite, le vecteur de poids ayant la meilleure corrélation, ainsi que certains de ses voisins, vont être modifiés afin d'augmenter encore cette corrélation.
- *Les réseaux de Hopfield*: sont des réseaux récurrents et entièrement connectés. Dans ce type de réseau, chaque neurone est connecté à chaque autre neurone et il n'y a aucune différenciation entre les neurones d'entrée et de sortie. Ils fonctionnent comme une mémoire associative non-linéaire et sont capables de trouver un objet stocké en fonction de représentations partielles ou bruitées. L'application principale des réseaux de Hopfield est l'entrepôt de connaissances mais aussi la résolution de problèmes d'optimisation. Le mode d'apprentissage utilisé ici est le mode non-supervisé.
- *Les ART*: Les réseaux ART ("Adaptative Resonance Theorie") sont des réseaux à apprentissage par compétition. Le problème majeur qui se pose dans ce type de réseaux est le dilemme «stabilité/plasticité». En effet, dans un apprentissage par compétition, rien ne garantit que les catégories formées restent stables. La seule possibilité, pour assurer la stabilité, serait que le coefficient d'apprentissage tende vers zéro, mais le réseau perdrait alors sa plasticité. Les ART ont été conçus spécifiquement pour contourner ce problème. Dans ce genre de réseau, les vecteurs de poids ne seront adaptés que si l'entrée fournie est suffisamment proche, d'un prototype déjà connu par le réseau. On parlera alors de résonance. A l'inverse, si l'entrée s'éloigne trop des prototypes existants, une nouvelle catégorie va alors se créer, avec pour prototype, l'entrée qui a engendrée sa création. Il est à noter qu'il existe deux principaux types de réseaux ART : les ART-1 pour des entrées binaires et les ART-2 pour des entrées continues. Le mode d'apprentissage des ART peut être supervisé ou non.

II.1.4 Apprentissage des réseaux de neurones:

On appelle « apprentissage » des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui lui est affectée [Dre, 98]. Cette phase est vraisemblablement la propriété la plus intéressante des réseaux de neurone. Il peut être considéré comme une action de mise à jour des poids des connexions synaptiques jusqu'à l'obtention du comportement désiré afin de résoudre le problème considéré. Dans la littérature on trouve deux principales modes d'apprentissage: supervisé et non supervisé selon les schémas de la Figure II.9 [Par, 04].

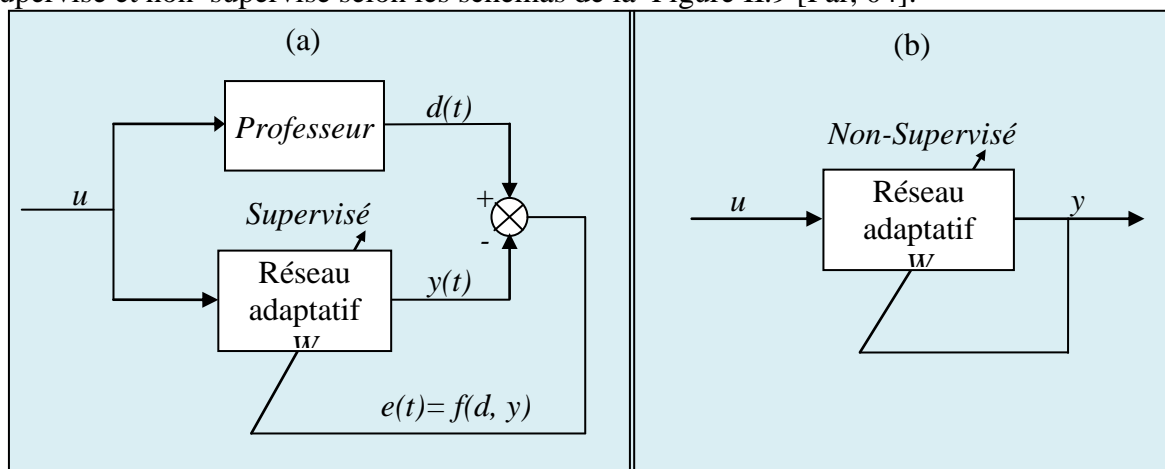


Figure II.9: Différence entre l'apprentissage supervisé et non-supervisé du réseau de neurone.

- *Mode supervisé*: L'apprentissage supervisé est illustré d'une manière conceptuelle par la Figure II.9(a). Le professeur produit une sortie désirée d pour ce stimulus, grâce à ses connaissances intrinsèques. On suppose que cette réponse est optimale. Elle est ensuite comparée (par soustraction) avec la sortie du réseau pour produire un signal d'erreur $e(t) = f(d, y)$ qui est réinjecté dans le réseau pour modifier son comportement via une procédure itérative qui, éventuellement, lui permet de simuler la réponse du professeur. Autrement dit, la connaissance de l'environnement par le professeur est graduellement transférée vers le réseau jusqu'à l'atteinte d'un certain critère d'arrêt. Par la suite, on peut éliminer le professeur et laisser le réseau fonctionner de façon autonome.

En résumé, les paramètres W du réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées u fournies, et la réponse attendue en sortie y . Ainsi, le réseau se modifie jusqu'à obtenir la meilleure sortie, par rapport à celle désirée d , c.a.d jusqu'à l'obtention d'une erreur $e(t) = f(d, y)$ minimale [Par, 04].

Ce mode d'apprentissage dispose d'un signal d'erreur qui non seulement permet de calculer un indice de satisfaction (par exemple l'erreur quadratique moyenne), mais permet aussi d'estimer le gradient local qui indique une direction pour l'adaptation des poids synaptiques. C'est cette information fournie par le professeur qui fait toute la différence.

- *Le mode non-supervisé (ou auto-organisationnel)*: ce mode ne nécessite pas une connaissance a priori de la sortie du système, il est basé sur des probabilités. Le réseau se modifie en fonction des régularités statistiques de l'entrée et établit des catégories, en attribuant et en optimisant une valeur de qualité, par rapport aux catégories reconnues.

Normalisation des entrées

- *La normalisation* : Avant l'opération d'apprentissage, il est indisponible de normaliser et de centrer toutes les variables d'entrées, en effet si ces entrées ont des grandeurs très différents, celle qui sont petites n'ont pas d'influence sur l'apprentissage.

Donc pour chaque vecteur d'entée x_i on effectue le changement de variable x'_i .

$$x'_i = \frac{x_i}{\max(x_i) - \min(x_i)} \quad (\text{II.7})$$

Avec \max et le \min sont les valeurs maximum et minimum établies sur des données d'apprentissage.

II.1.5 Algorithme d'apprentissage:

- *Principe de la rétro-propagation*: appelée back-propagation permet de confronter le réseau à des exemples déjà classés (phase de propagation avant : l'information se propage de la couche d'entrée vers la couche de sortie). Lorsque le réseau a obtenu un résultat, l'erreur E de classification est calculée et ensuite rétro-propagée dans le réseau (phase de rétro-propagation (back-propagation)) pour que celui-ci modifie les poids de connexion W . Le mode d'apprentissage est supervisé.

Le principe est de minimiser l'erreur quadratique entre les sorties calculées et celles souhaitées suivant l'équation (II.8). Il s'agit ensuite de calculer la contribution de cette erreur sur la modification de chacun des poids synaptiques (l'erreur calculée en sortie est transmise en sens inverse vers l'entrée). L'erreur quadratique est définie par :

$$E = \frac{1}{2} \sum_{j=1}^J \sum_{k=1}^K (d_{j,k} - y_{j,k})^2 \quad (\text{II.8})$$

Où :

J : le nombre d'exemples et K le nombre de sorties.

- *L'algorithme de descente du gradient*: L'algorithme d'optimisation le plus utilisé est la descente du gradient, dont le principe est de partir d'un point aléatoire puis de se déplacer dans la direction de la plus forte pente.

Cet algorithme invite à modifier les matrices des paramètres de poids W du réseau d'une quantité Δw_{ij} . Avec chaque pas ces paramètres sont actualisés par la règle de mise à jour:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \quad (\text{II.9})$$

Avec :

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \quad (\text{II.10})$$

Où :

η : Pas d'apprentissage $0 < \eta < 1$.

En appliquant un certain nombre d'itérations, l'algorithme doit converger vers une solution qui est un minimum globale de la fonction coût f à optimiser (figure II.9):

Si le pas du gradient est trop petit, il risque d'être attrapé dans un minimum local.

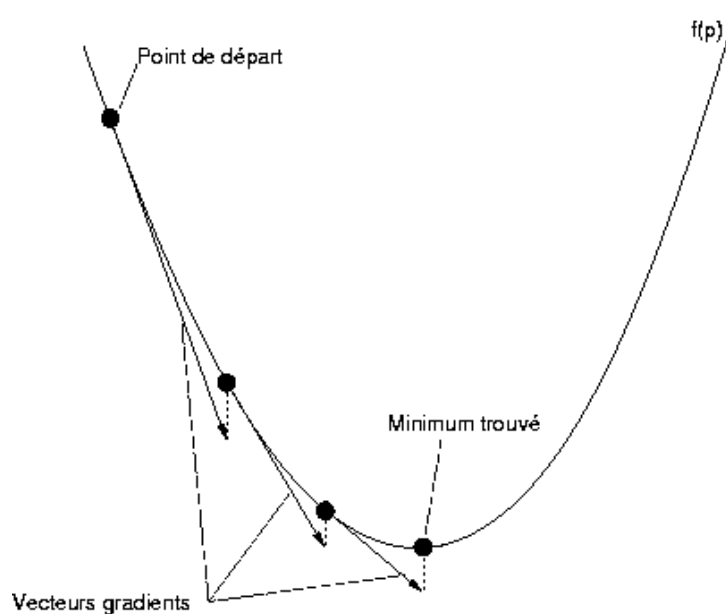


Figure II.9: Illustration du principe de la descente de gradient, dans le cas d'une fonction à une variable

D'une manière générale, on commence donc par choisir un vecteur P_0 de manière aléatoire. Puis pour l'itération numéro i on calcule le gradient de f au point P_{i-1} :

$$g_i = \nabla f(P_{i-1}) \quad (\text{II.11})$$

Le nouveau vecteur de paramètres calculé est :

$$P_i = P_{i-1} - \eta g_i \quad (\text{II.12})$$

Où η est une constante qui ajuste la vitesse de convergence de l'algorithme. Une fois le nouveau vecteur P_i calculé on passe à l'itération suivante. Si η est trop grand, on obtient une convergence rapide avec un risque d'instabilité, et si η est trop petit, on se confronte à la lenteur de la convergence, mais la probabilité de convergence vers un minimum local est plus grande. Plusieurs critères peuvent être définis pour arrêter l'algorithme : le nombre d'itérations, un seuil atteint par $f(P_i)$ ou encore lorsque le vecteur évolue peu, c'est à dire quand la valeur suivante atteint un seuil minimal :

$$\frac{\|g_i\|}{\|P_i\|} \quad (\text{II.13})$$

Ce dernier critère peut présenter le défaut d'arrêter l'algorithme trop tôt si la fonction présente des plateaux. Le choix du meilleur critère ainsi que le seuil à fixer sont généralement trouvés de manière empirique. Il est également possible de prendre une combinaison de ces différents critères. Le choix du coefficient η peut être délicat dans certains cas. Par exemple si f possède par endroits de grands plateaux, il faudrait avoir un coefficient η grand pour pouvoir s'en affranchir avec peu d'itérations. Si en d'autres endroits f évolue au contraire très rapidement, il faut qu'il soit faible pour que l'algorithme soit stable. Une variante, la descente de gradient adaptative, peut être utile dans ce cas.

Dans une descente de gradient adaptative, le coefficient η est également ajusté à chaque itération, suivant l'évolution de la valeur de $f(P_i)$. Si $f(P_i)$ diminue, il est probable que l'on pourrait aller plus vite en augmentant légèrement η , et au contraire si $f(P_i)$ augmente, cela veut dire que le coefficient η est trop grand et qu'il faut le diminuer. Donc on décide d'augmenter η (de 10% par exemple) si $f(P_i)$ diminue, et de le réduire (en le divisant par 2 par exemple) si $f(P_i)$ augmente. Cette approche permet généralement de réduire le nombre d'itérations requis, et s'est révélée efficace avec tous les réseaux de neurones que nous avons testés.

Les inconvénients connus de cette méthode sont :

- le choix de η est empirique.
- Si η est trop petit, le nombre d'itérations peut être très élevé, et l'algorithme risque de converger vers un minimum local.
- Si η est trop grand, il y'a un risque d'oscillation autour du minimum sans l'atteindre,
- Rien ne garantit que le minimum trouvé soit un minimum global.

- Résumé de l'algorithme de rétro-propagation du gradient

- ✓ Initialisation aléatoire des poids W du réseau entre et application d'un exemple du vecteur d'entrée X ,
- ✓ Calculer l'erreur en sortie et la rétro-propager de couche en couche jusqu'à la couche d'entrée.
- ✓ Mise à jour des poids (équation (II.9)).
- ✓ Répéter ce processus jusqu'à l'arrêt de l'apprentissage.

L'algorithme de la descente du gradient standard, étant un algorithme d'optimisation du premier ordre, il est connu par sa lenteur de convergence et le risque d'être attrapé dans un minimum local. En introduisant le Hessien dans le calcul de la descente du gradient, Gauss, a donné naissance à un algorithme du second ordre qui converge plus rapidement vers un minimum global. Le pas du gradient de Gauss étant trop grand, il y'a un risque d'oscillation quant on est proche du minimum, pour cette raison une deuxième modification a été apportée par Levenberg-Marquardt [Lev, 44], [Mar, 63], pour moduler le pas (Grant au début de l'apprentissage et plus petit qu'on se rapproche du minimum). Cet algorithme est performant, et très rapide, son inconvénient est qu'il nécessite un grand espace mémoire.

La convergence du réseau par rétro-propagation est un problème crucial car il requiert de nombreuses itérations. Pour pallier à ce problème, il existe des techniques plus ou moins rapides, performantes et ne pas un. Il apparaît que la technique de Levenberg-Marquardt est un algorithme très rapide.

- L'algorithme de Levenberg-Marquardt: est une méthode d'apprentissage avec un pas adaptatif. C'est une autre manière de diminuer le nombre d'itérations d'un algorithme

d'optimisation et d'utiliser les dérivées secondes de f . En effet le gradient donne une direction vers laquelle se déplacer pour trouver le minimum, mais ne donne pas le pas. Dans la descente de gradient classique ce pas est un coefficient fixe, et dans la variante adaptative il peut varier à chaque itération. Mais la dérivée seconde de f est liée au rayon de courbure de la fonction, et permet donc de déterminer ce pas de manière plus fine (Figure II.10).

En effet si l'on suppose que f est une fonction quadratique :

$$f(p) = a + b^T p + p^T C p \quad (\text{II.14})$$

Où p^T est la transposée du vecteur p et C est une matrice symétrique, on peut trouver l'extremum de la fonction f , il s'agit du point où la dérivée de f s'annule :

$$\nabla f = 0 \Leftrightarrow b + 2Cp = 0 \quad (\text{II.15})$$

Soit :

$$p = -2C^{-1}b \quad (\text{II.16})$$

A condition que C soit inversible. Pour une fonction f quelconque, il est possible de l'approcher localement en un point p_i par une fonction quadratique, en utilisant ses dérivées première et seconde, et avec l'équation (II.16) déterminer le vecteur pour l'itération suivante d'un algorithme d'optimisation plus évolué que la descente de gradient. Mais le calcul des dérivées secondes peut être très long, tout d'abord parce que le nombre de dérivées secondes est le carré de celui des dérivées premières, et également parce que la dérivée seconde de f peut être assez complexe. De nombreux algorithmes, peuvent-être abusivement appelés algorithmes d'ordre 2, ils utilisent en fait une approximation des dérivées secondes calculées à partir de dérivées premières. Cependant ils gardent l'avantage de nécessiter beaucoup moins d'itérations qu'une descente de gradient. L'algorithme de Levenberg Marquardt [Mar, 63] fait partie de ces algorithmes, et s'applique au cas particulier où f est une erreur quadratique moyenne. On peut donc l'exprimer sous la forme :

$$f(p) = \langle (\theta(x, p) - y)^2 \rangle \quad (\text{II.17})$$

où θ désigne une fonction de deux vecteurs x et p et $\langle \cdot \rangle$ désigne la moyenne calculée sur un ensemble de couples (x, y) . On se place dans le cas où θ est une fonction scalaire afin de simplifier la notation, mais la même démarche peut être faite si f est une fonction vectorielle. Dans la suite de cette section toutes les dérivées sont en fonction du vecteur p . C'est en effet uniquement ce vecteur que l'on fait varier afin de trouver le minimum de f .

On suppose, comme pour la descente de gradient, que l'on cherche à calculer un nouveau vecteur p_i en fonction de p_{i-1} , tel que $f(p_i)$ se rapproche plus d'un minimum local de f . Pour cela on calcule une approximation quadratique \hat{f} de f à partir d'une approximation linéaire $\hat{\theta}$ de θ autour du point p_{i-1} . En déterminant le point p auquel le gradient de \hat{f} s'annule, on obtient :

$$p = p_{i-1} - H^{-1}d \quad (\text{II.18})$$

Avec :

$$d = \langle (\theta(x, p_{i-1}) - y) \nabla \theta(x, p_{i-1}) \rangle \quad (\text{II.19})$$

$$H = \langle \nabla \theta(x, p_{i-1}) \nabla \theta(x, p_{i-1})^T \rangle \quad (\text{II.20})$$

A condition que H soit inversible. La matrice H est une approximation du Hessien de f , calculée à partir du gradient de θ . L'équation précédente pourrait servir dans un algorithme d'optimisation, qui permet de calculer p_i à partir de p_{i-1} au cours de l'itération. Mais ceci n'est

efficace en pratique que si θ est effectivement proche d'une droite autour du point p_{i-1} . Dans le cas contraire cet algorithme donne de très mauvais résultats.

L'idée de Levenberg est donc d'utiliser cette approche quadratique dans les zones où θ est quasi-linéaire, et une descente de gradient dans les autres cas. Le pas d'une itération de cet algorithme est calculé de la manière suivante :

$$p_i = p_{i-1} - (H + \lambda I)^{-1} d \quad (\text{II.21})$$

Lorsque λ est faible, cette équation est équivalente à (9), et le nouveau vecteur de paramètres est déterminé avec l'approximation quadratique de f . Lorsque λ est grand, cette équation est équivalente à :

$$\begin{aligned} p_i &= p_{i-1} - \frac{1}{\lambda} d \\ &= p_{i-1} - \frac{1}{\lambda} \langle (\theta(x, p_{i-1}) - y) \nabla \theta(x, p_{i-1}) \rangle \\ &= p_{i-1} - \frac{1}{2\lambda} \nabla f(x, p_{i-1}) \end{aligned} \quad (\text{II.22})$$

Ce qui correspond bien à une descente de gradient. Pour des valeurs intermédiaires de λ l'algorithme est un mélange entre la descente de gradient et l'approche quadratique basée sur l'approximation linéaire de θ . Ce coefficient λ est modifié à chaque itération, comme pour la descente de gradient adaptative. Si $f(p_i)$ diminue au cours de l'itération, on diminue λ (en le divisant par 10 par exemple), et l'on se rapproche ainsi de la méthode quadratique. Au contraire si $f(p_i)$ augmente, cela signifie que nous nous trouvons dans une région dans laquelle θ n'est pas très linéaire, et donc on augmente λ (en le multipliant par 10 par exemple) afin de se rapprocher de la descente de gradient.

Cet algorithme a ensuite été amélioré par Marquardt, le pas de l'itération étant défini cette fois par :

$$p_i = p_{i-1} - (H + \lambda \text{diag}(H))^{-1} d \quad (\text{II.23})$$

La matrice identité a été remplacée par la diagonale de H . Le but est de modifier le comportement de l'algorithme dans les cas où λ est grand, c'est à dire lorsque l'on est proche d'une descente de gradient. Avec cette modification l'on se déplace plus vite dans les directions vers lesquelles le gradient est plus faible, afin d'éviter de passer de nombreuses itérations sur un plateau. Ceci est appelé *l'algorithme de Levenberg Marquardt*.

- *Résumé de cet algorithme:*

- ✓ Présentez toutes les entrées au réseau et calculez les sorties et les erreurs correspondantes de réseau
- ✓ Calculez la matrice H
- ✓ Mise à jour des poids
- ✓ Recalculer l'erreur après la mise à jour des poids. Si cette erreur est inférieure à celle calculée à l'étape 1 le paramètre d'apprentissage λ est réduit par λ^- , sinon le paramètre d'apprentissage λ est augmenté par λ^+ . Les paramètres λ^- , λ^+ sont prédéfinis par l'utilisateur, généralement λ^+ est fixé à 10 et λ^- est fixé à 0.1.
- ✓ L'algorithme converge quand la norme du gradient est inférieure à une certaine valeur prédéterminée, ou quand l'erreur a été réduite à un certain but d'erreur.

En pratique cet algorithme, en particulier dans le cas des réseaux de neurones, permet de converger avec beaucoup moins d'itérations. Mais chaque itération demande plus de calculs, en particulier pour l'inversion de la matrice H , et son utilisation se limite donc aux cas où le

nombre de paramètres à optimiser n'est pas très élevé. En effet le nombre d'opérations nécessaires à l'inversion d'une matrice est proportionnel à N^3 , N étant la taille de la matrice, et ici également la taille du vecteur p .

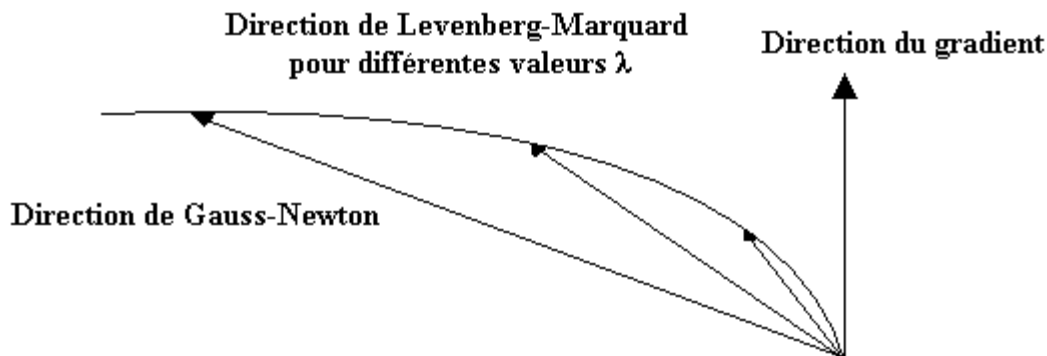


Figure II.10 : Direction du gradient pour différents algorithmes

La méthode d'apprentissage que nous avons employée, est celle du gradient total. Le vecteur g_i est calculé avec tous les exemples de la base d'apprentissage à chaque itération, et le nouveau vecteur des paramètres est déterminé après avoir parcouru toute la base. Pour cette méthode il est nécessaire d'avoir la base complète avant de commencer la première itération. C'est une méthode d'apprentissage dite Batch (ou hors ligne).

Dans la méthode d'apprentissage, dite du gradient stochastique, le vecteur g_i est calculé avec chaque exemple présenté, et le vecteur de paramètres est recalculé entre chaque exemple. Cette dernière méthode est particulièrement adaptée aux systèmes dits On-line (en ligne ou séquentiel), pour lesquels les exemples sont communiqués l'un après l'autre pendant l'optimisation.

II.1.6 La classification par les réseaux de neurone:

La classification est l'une des fonctions réalisées par les réseaux de neurones. Cette fonction peut se réaliser selon que les ensembles sont linéairement séparables (classification linéaire) ou non linéairement séparables (classification non-linéaire). La classification par les réseaux de neurones fait partie des méthodes de classification non-linéaire (perceptrons multicouches (MLP) et les réseaux de neurones RBF) et modélise la fonction non linéaire de séparation en choisissant un nombre de neurones de la couche cachée selon la complexité du problème [Ben, 04].

Supposons que l'on désire classer des formes en deux ensembles, A ou B , en fonction de certaines caractéristiques de ces formes ; on peut définir une fonction Φ qui vaut $+1$ pour toutes les formes de la classe A et -1 pour toutes les formes de la classe B . Les réseaux de neurones sont de bons candidats pour réaliser une approximation de cette fonction Φ , et l'on peut démontrer que cette approximation constitue une estimation de la probabilité d'appartenance de la forme inconnue à la classe A . Les réseaux de neurones fournissent donc une information très riche, qui est loin d'être une simple réponse binaire. Cette propriété remarquable (que les réseaux de neurones partagent avec d'autres classifieurs) n'est malheureusement pas mise à profit dans la plupart des applications.

II.1.7 Quelques points essentiels à retenir:

Nous avons exposé les éléments essentiels qui permettent de comprendre pourquoi, et dans quels cas, il est avantageux d'utiliser des réseaux de neurones. En présentant quelques applications typiques, nous avons montré de manière concrète ce que l'ingénieur peut attendre de cette technique. S'ils sont bien compris et convenablement utilisés, les réseaux de neurones

peuvent rendre des services considérables dans des domaines très variés. Voici quelques points fondamentaux à retenir lors de la procédure de mise en œuvre des réseaux de neurones:

- Les réseaux de neurones sont des outils statistiques, qui permettent d'ajuster des fonctions non linéaires très générales à des ensembles de points; l'utilisation de réseaux de neurones nécessite que l'on dispose de données suffisamment nombreuses et représentatives ;
- Les réseaux de neurones sont des estimateurs parcimonieux ;
- Les réseaux de neurones permettent de modéliser des phénomènes statiques (réseaux non bouclés) et dynamiques (réseaux bouclés) ;
- Il est toujours souhaitable, et souvent possible, d'utiliser, pour la conception du réseau, les connaissances mathématiques dont on dispose sur le phénomène à modéliser : les réseaux de neurones ne sont pas nécessairement des "boîtes noires".

II.2 La logique floue

II.2.1 Introduction:

La logique floue dans la plupart des applications actuelles, permet de prendre en compte toutes sortes de connaissances qualitatives de concepteurs et d'opérateurs dans l'automatisation des systèmes. Les nombreuses applications dans l'électroménager et l'électronique grand public réalisées notamment au Japon en ont été l'élément déclenchant. Des programmes de machines à laver, de caméscopes avec stabilisateur d'images,...etc, ont été réalisés à l'aide de la logique floue. Dans le domaine des processus de production, continus et par lots, et dans les automatismes, les applications se sont également multipliées. La logique floue s'y développe car il s'agit d'une approche essentiellement pragmatique, efficaces, et générique. On dit parfois qu'elle permet de systématiser ce qui est du domaine de l'empirique, et donc difficile à maîtriser. La théorie des ensembles flous fournit une méthode adéquate et facilement réalisable dans des applications temps réel, où elle permet de transcrire et de rendre dynamiques les connaissances des concepteurs et des opérateurs. Cet aspect adaptable et universel de la logique floue permet de s'attaquer à l'automatisation de procédures telle que la mise en route, le réglage de paramètres pour lesquels peu d'approches existaient auparavant.

- *Bref historique:*

-1965: Concept introduit par Pr. Lotfi Zadeh "Fuzzy set theory". Définition des ensembles flous et opérateurs associés.

-1970: Premières applications : système expert, aide à la décision en médecine, commerce,...etc.

-1974: Premières applications industrielle. Régulation floue d'une chaudière à vapeur réalisée par Mamdani.

-1985: Les premiers, les Japonais introduisent des produits grand public "Fuzzy Logic Inside".

-1990: Généralisation d'utilisation de cette technique.

- *Définition et principe:* La logique floue est née de constatation que la plupart des phénomènes ne peuvent pas être représentés à l'aide des variables booléennes qui ne peuvent prendre que deux valeurs (0 ou 1). Peut on considérer une eau à 18c° comment étant chaude ou froide? N'est elle pas ni vraiment chaude, ni vraiment froide mais tout simplement tiède ? Pour reprendre à ce type de question, la logique floue considère la notion d'appartenance d'un objet à un ensemble non plus comme une fonction booléenne mais comme une fonction qui peut prendre toutes les valeurs entre 0 et 1.

Dans ce contexte les 2 concepts de base de la logique floue sont:

- Les ensembles et variables flous et opérateurs associés.
- Prise de décision à partir d'une base de règles *SI...ALORS...*, c'est l'inférence floue.

II.2.2 L'ensemble flou:

Les ensembles flous sont des extensions de la théorie des ensembles classiques utilisées dans la logique floue qui ont été formalisées par le professeur Lotfi Zadeh à l'université de Californie en 1965.

Un ensemble flou est caractérisé par sa fonction d'appartenance, qui projette les variables linguistiques de l'univers de discours dans l'intervalle $[0,1]$. Pour un univers de discours X donné le degré d'appartenance est $\mu : X \rightarrow [0,1]$, et l'ensemble flou A est défini comme ceci :

$$A = \{(x, \mu(x)) \mid x \in X\}$$

L'exemple suivant illustre cette notion :

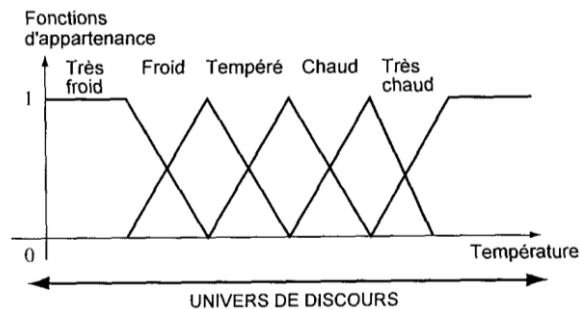


Figure II.11 : Fonctions d'appartenances

Univers du discours : Gamme de température de 0°C à 200°C.

Variable linguistique : La température.

Valeurs linguistiques : «Très froid», «Froid», «Tempéré», «Chaud», et «Très Chaud».

II.2.3 Système flou:

Le principe de fonctionnement d'un système flou est simple, celui-ci se décompose en trois étapes distinctes (Figure II.12).

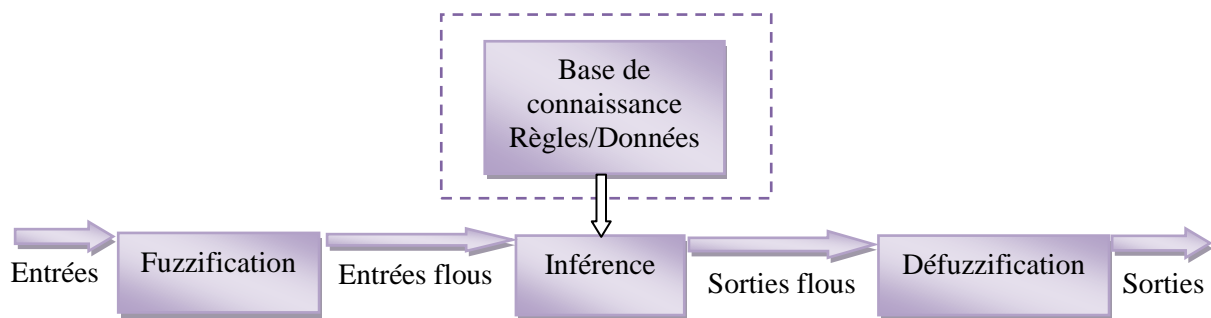


Figure II.12 : Système Flou

- *Fuzzification*: Cette première étape consiste à déterminer le degré d'appartenance de chaque variable d'entrée à chaque état. Ceci est déterminé à l'aide des fonctions d'appartenance définies dans le système. Ces fonctions d'appartenance, qui utilisent entre trois et sept états flous pour chaque variable, possèdent différentes forme : triangle, cloche, trapèze, etc...

- *Inférence floue*: Si (x est A) Alors (y est B)

- La variable floue x appartient à la classe floue A avec un degré de validité $\mu(x_0)$.

- La variable floue y appartient à la classe floue B avec un degré qui dépend du degré de validité $\mu(x_0)$ de la prémisse.

Donc plus la condition sur les entrées est vraie. Plus l'action préconisée pour les sorties doit être respectée. [Yag, 94]

Méthode d'implication:

Il y a 2 méthodes principales d'implication floue:

- Méthode de Mamdani

$$\mu'_{conclusion}(y) = \underset{y}{MIN}(\mu_{prémisse}(x_0), \mu_{conclusion}(y))$$

- Méthode de Larsen

$$\mu'_{conclusion}(y) = \mu_{prémisse}(x_0) \times \mu_{conclusion}(y)$$

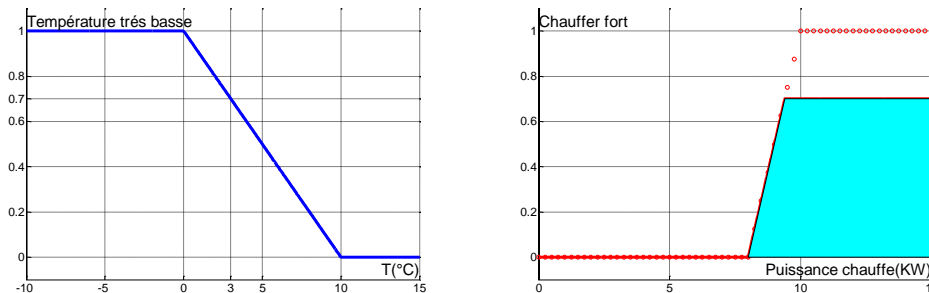
Exemple (Mamdani) :

Règle : SI la température est très basse ALORS Chauffer fort

La conclusion d'une règle floue est l'appartenance d'une variable floue de sortie « Chauffer » à une classe floue « fort ».

Cette appartenance dépend de :

1. La classe floue de sortie considérée.
2. du degré de validité de la prémisse $\mu_{prémisse}(x_0)$.
3. de la méthode d'implication choisie.



Selon la règle considérée, si $T=3^{\circ}\text{C}$ alors Puissance de chauffe = 12KW

Figure II.13 : Exemple d'implication (méthode de Mamdani)

- *Défuzzification*: En sortie le système flou ne peut pas communiquer des valeurs floues qui ne peuvent être exploitées. Il lui est donc nécessaire de fournir des valeurs précises, c'est le rôle de la défuzzification. Cette étape s'effectue toujours à l'aide des fonctions d'appartenance. A partir des degrés d'appartenance, on obtient autant de valeurs qu'il y'a d'états. Pour déterminer les valeurs précises, deux principales méthodes de défuzzification sont proposées :

Méthodes de centre de gravité (COG): C'est l'abscisse du centre de gravité de la surface sous la courbe de résultat :

$$sortie = \frac{\int_U y \cdot \mu(y) \cdot dy}{\int_U \mu(y) \cdot dy} \quad (\text{II.24})$$

U = Univers du discours

= Toutes les valeurs de sorties considérées

Méthode moyenne des maximums (MM): C'est la moyenne des valeurs de sortie les plus vraisemblables :

$$sortie = \frac{\int_S y \cdot dy}{\int_S dy} \quad (II.25)$$

où $S = \left\{ y_0 \in U / \mu(y_0) = \sup_{y \in U} (\mu(y)) \right\}$

En commande floue la défuzzification (COG) est presque toujours utilisée. Elle prend en compte l'influence de l'absence des valeurs proposées par la solution floue. La défuzzification (MM) est plutôt utilisée lorsque il s'agit de discriminer une valeur de sortie (Ex : Reconnaissance des formes).

II.2.4 Application de la logique floue:

La logique floue, qui possède les mêmes capacités de représentation que les réseaux de neurones est surtout utilisée pour des applications où l'on dispose d'une expertise humaine pour résoudre un problème (Contrairement aux réseaux de neurones qui nécessitent essentiellement des données). Elle est ainsi utilisée en commande de processus, dans le domaine du diagnostic des défauts (détection et classification), et dans plusieurs autres domaines.

- *Commande de processus:* De nombreux systèmes de contrôle pourraient être pilotés par un expert humain. Prenons l'exemple de la régulation d'un système d'arrosage de jardin. L'expert serait capable d'expliquer sa stratégie de contrôle en termes de règles floues. Par exemple : s'il fait relativement chaud dans l'après midi et s'il fait beaucoup de vent, alors il faut arroser abondamment. La logique floue possède donc sa place dans les systèmes de contrôles.

- *Diagnostic des défauts:* L'utilisation de la logique floue dans le domaine du diagnostic des défauts possède plusieurs avantages parmi les quelles :

- Une représentation évidente des sous systèmes étudiés.
- Une interprétation linguistique dans une forme d'ensembles flous.
- La capacité de combiner les connaissances quantitatives et qualitatives.
- Les règles extraites des données peuvent être validées par l'expert.
- La capacité de représenter certains genres d'incertitudes présentes dans le processus réel.

En plus la prise en compte de la validation du résidu permet d'appréhender la dynamique du défaut et d'anticiper la décision. Par exemple (Figure II.11), si le résidu est grand et augmente vite (dynamique rapide du défaut), alors la méthode considère que la situation est plus grave que s'il est grand et reste constant ou diminue.

Pour illustrer le traitement de données qualitatives de manière facilement interprétable et pour analyser la génération de règles dans des arbres de décision, on peut donner l'exemple comme suite: On consiste à classer 8 individus en deux groupes (classes), ceux qui risquent le coup de soleil et ceux qui ne le risquent pas. Chaque individu est caractérisé par quatre attributs de type qualitatif, deux représentent les possibles facteurs du coup de soleil et les deux autres sont ajoutés pour établir la capacité des algorithmes à ne pas tenir compte d'information superflue pour la classification. Les données sont présentées dans le Tableau II.2. Les descripteurs avec leurs modalités respectives sont :

- **Cheveux:** Blonds, Bruns et Roux
- **Taille:** Grande, Moyenne et Petite
- **Poids:** Elevé, Moyen et Faible
- **Application Crème Solaire:** Oui, Non

Tableau II.2: Description des Individus.

Nom	Cheveux	Taille	Poids	Crème Solaire	Résultat
Sarah	Blonds	Moyenne	Faible	Non	Coup de Soleil (1)

Dana	Blonds	Grande	Moyen	Oui	Non (2)
Alex	Bruns	Petite	Moyen	Oui	Non (2)
Annie	Blonds	Petite	Moyen	Non	Coup de Soleil (1)
Emily	Roux	Moyenne	Elevé	Non	Coup de Soleil (1)
Pete	Bruns	Grande	Elevé	Non	Non (2)
John	Bruns	Moyenne	Elevé	Non	Non (2)
Katie	Blonds	Petite	Faible	Oui	Non (2)

Les données transformées en descripteurs quantitatifs sont présentées dans le Tableau II.4. La représentation des variables qualitatives en variables quantitatives telle qu'elle a été décrite précédemment, a été employée. Ainsi pour la taille et les attributs de poids, l'échelle de $\{0, \frac{1}{2}, 1\}$ a été utilisée. Pour l'attribut « application de crème solaire », une association binaire a été faite et dans le cas de la couleur des cheveux, le descripteur a été transformé en trois descripteurs (Blonds, Bruns et Roux) ; la présence de la modalité dans l'individu est représentée par un 1 dans le descripteur correspondant et 0 pour les deux autres.

Tableau II.3: Représentation quantitative des Individus.

Individu	Blonds	Bruns	Roux	Taille	Poids	Crème Solaire	Classe
1	1	0	0	0.5	0	0	1
2	1	0	0	1	0.5	1	2
3	0	1	0	0	0.5	1	2
4	1	0	0	0	0.5	0	1
5	0	0	1	0.5	1	0	1
6	0	1	0	1	1	0	2
7	0	1	0	0.5	1	0	2
8	1	0	0	0	0	1	2

II.2.5 Logique floue et système expert:

La logique floue et les systèmes experts sont fortement liés. En effet, la construction d'un modèle en logique floue passe toujours par une expertise humaine sous la forme de règle floue, celle-ci est ensuite utilisée par le moteur d'inférence d'un système expert. Comme ces règles sont exprimées en langage naturel à l'aide de termes vagues, elles sont en général facilement formulables par l'expert lui-même, ce qui n'est pas le cas pour les systèmes experts classiques qui nécessitent des valeurs précises comme valeurs manipulées.

II.3 Les Réseaux Neuro-Flous

II.3.1 Introduction:

Les Systèmes neuro-flous qui combinent la logique floue, et les réseaux de neurones, prouvent leur efficacité dans une variété de problèmes du monde réel et dans l'industrie. Chaque technique intelligente a des propriétés particulières (par exemple capacité d'apprentissage, explication de décisions). Chaque technique permet de résoudre certains problèmes particuliers. En effet, les réseaux neuronaux sont par exemple utilisés pour la reconnaissance des modèles. Cependant, ils sont incapables d'expliquer comment ils atteignent leurs décisions. Les systèmes à base de logique floue peuvent raisonner avec l'information imprécise et expliquer leurs décisions mais ne peuvent cependant pas acquérir automatiquement les règles qu'ils utilisent pour prendre ces décisions. Ces limites ont été une raison derrière la création de systèmes neuro-flous.

-Définition et principe: Un système neuro-flou est un réseau de neurone avec un signal flou, des poids flous, une fonction de transfert floue et un module d'adaptation de la structure de RNF par un apprentissage d'un ensemble de données [Nau, 95].

Dans le cas où 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 d'un système capable de générer des relations à partir des exemples (ou échantillons d'apprentissage).

Par nature, un réseau neuro-flou est un réseau de neurones multicouches [Nau, 94]; [Nau, 95] avec des principes de fonctionnement comme ceux décrits plus haut. La modification se fait par l'introduction des ensembles flous dans la modélisation des données entrées et la mesure des poids de connexions. Alors on peut dire que le réseau neuro-flou est une version « fuzzifiée » du réseau de neurones multicouches.

Les entrées du modèle sont les vecteurs d'attributs $x = \{x_1, x_2, \dots, x_n\}$ représentant l'échantillon d'apprentissage. Chaque attribut x_i comprend q_i sous-ensembles flous $\mu_1(i), \mu_2(i), \dots, \mu_{q_i}(i)$ (il n'est pas nécessaire que les q_i soient égaux pour tous les attributs). Les sorties sont donc les classes C_j . Entre ces deux couches d'entrée et de sortie, se trouve la couche des règles R_k (couche cachée).

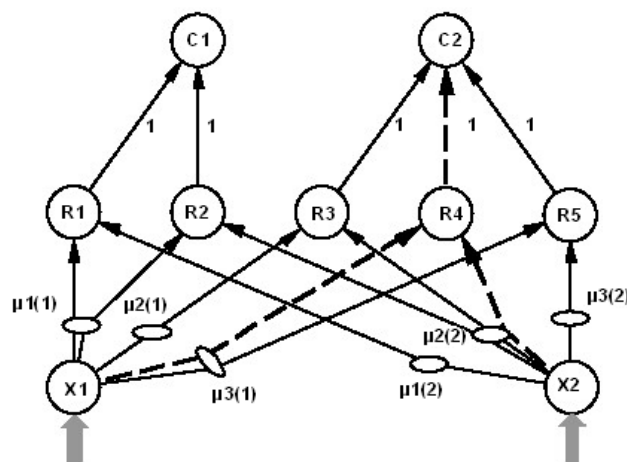


Figure II.14 : Système Neuro-Flou

La figure II.14 présente un modèle neuro-flou à 3 couches comprenant :

- 2 variables d'entrée x_1 et x_2 (attributs) représentées chacune par 3 sous ensembles flous : $\{\mu_1(1), \mu_2(1), \mu_3(1)\}$ et $\{\mu_1(2), \mu_2(2), \mu_3(2)\}$ (couche d'entrée).
- 5 règles R_1, \dots, R_5 (couche cachée)
- 2 variables de sortie (classes) C_1 et C_2 (couche de sortie)

Le processus d'apprentissage présente deux problèmes :

- La généralisation des règles
- La détermination des sous-ensembles flous pour chaque attribut.

Ces deux problèmes seront résolus parallèlement en propageant l'échantillon d'apprentissage à travers le système. Les principes d'apprentissage sont :

- La détermination des ensembles flous de l'espace des critères : en se fondant sur le rang des données entrées, le système fixe arbitrairement une partition initiale des sous-ensembles flous pour chaque attribut. Il s'agit de donner des seuils à chaque variable linguistique.
- L'élaboration des règles : lors qu'un individu étiqueté (ou individu de l'échantillon d'apprentissage, on parle aussi de pattern) est propagé, le système enregistre les sous-ensembles flous de chaque critère auxquels appartient l'objet en question. S'il n'existe pas encore de règle avec des prémisses identiques, on crée une règle et une connexion sera faite entre cette règle et la classe à laquelle appartient l'objet. Le processus se répète jusqu'à ce qu'il ne reste plus de pattern à propager [Nau, 95].

Par exemple dans la figure II.14, la règle R_4 s'énonce ainsi « Si x_1 est grand ($\mu_3(1)$) et x_2 est

moyen (μ_2 (2)), alors l'objet appartient à la classe C_2 ». Ce raisonnement est illustré par l'itinéraire pointillé dans la figure. Le réajustement des seuils des sous-ensembles flous de l'espace des critères : le processus de réajustement est un peu plus compliqué. Il s'agit d'une étape de rétro propagation, où l'on cherche de modifier les seuils des sous-ensembles flous en vue de palier la différence entre la sortie désirée (en général égale à 1) et celle obtenue. Plus précisément, lors de la propagation d'un individu à travers le modèle, en parallèle avec l'élaboration des règles, le système calcule aussi le degré d'appartenance de l'individu à sa classe (dans la plupart des cas, ce degré est inférieur à 1) ainsi que l'écart avec la valeur théorique (égale à 1). Un processus inverse donnera donc la modification des seuils nécessaire pour neutraliser cet écart.

II.3.2 Architectures des réseaux Neuro-Flous:

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

- *Première architecture*: Le premier type d'utilisation, le plus répandu, est représenté par le cas où un système d'inférence floue est mis sous la forme d'un réseau multicouche, dans lequel généralement les poids correspondent aux paramètres du système comme le montre la Figure II.15:

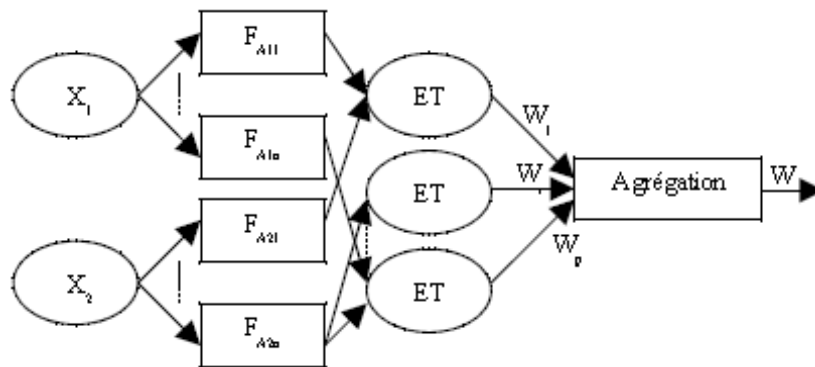


Figure II.15: Architecture d'un réseau Neuro-Flou hybride

- *Deuxième architecture*: Le deuxième grand type d'association neuronale et floue correspond à l'utilisation de réseaux de neurones et de systèmes flous organisés en série ou en parallèle. On distingue plusieurs possibilités d'association illustré dans la Figure II.16 (a), (b), (c).

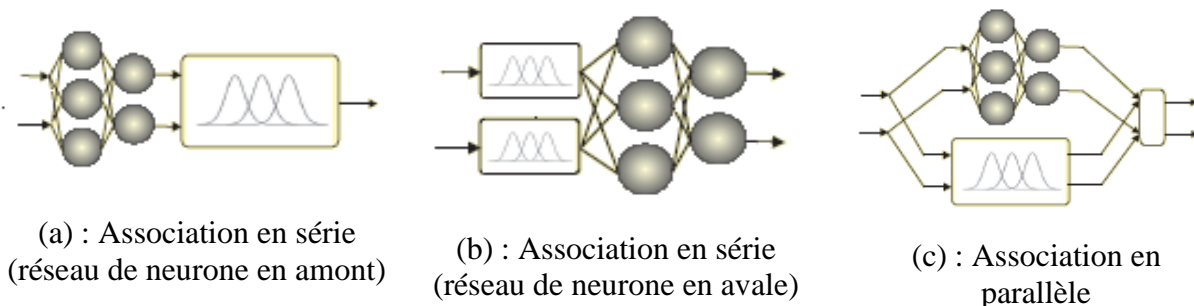


Figure II.16: Différents architectures d'un réseau Neuro-Flou Série/Parallèle.

- *Troisième architecture*: Ce type d'association entre réseau de neurones et systèmes flous, correspond à l'utilisation des réseaux de neurones pour remplacer toutes ou quelques composantes d'un système flou. De tels réseaux servent à l'apprentissage des fonctions d'appartenance, au calcul de l'inférence, à la réalisation de la phase d'agrégation et de défuzzification.

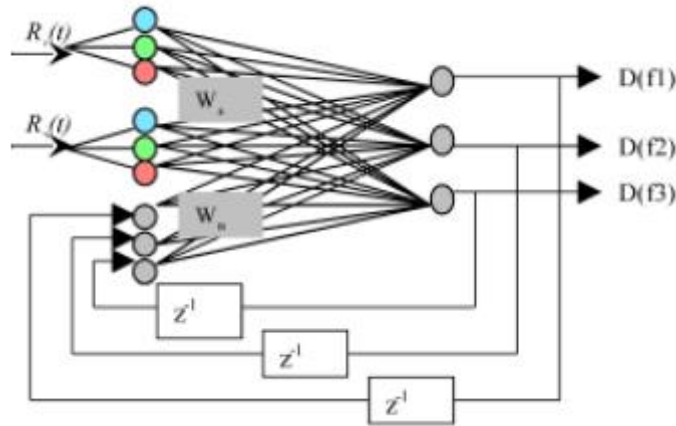


Figure II.17: Architecture d'un réseau Neuro-Flou dynamique

II.4 Application des RN, LF, et NF dans le diagnostic

II.4.1 Introduction:

Les problèmes de conception surgissent régulièrement dans les systèmes modernes de diagnostic de défaut. Malheureusement, les techniques analytiques classiques souvent ne peuvent pas fournir les solutions acceptables à ces problèmes. Ceci explique pourquoi les techniques de l'intelligence artificielle telle que les réseaux de neurone et la logique floue deviennent de plus en plus populaires dans des applications industrielles de diagnostic de défaut. L'objectif principal de ce chapitre est de présenter des développements concernant l'application des réseaux de neurones et la logique floue au diagnostic de défaut.

Nous avons adopté la méthode de diagnostic à base de modèle de type boîte noire. Notre travail se divise en deux étapes :

- Elaboration d'un modèle neuronal pour la génération des résidus,
- Concevoir un classifieur pour classer les fautes.

En première étape nous avons modélisé le système par des réseaux de neurones. Dans la 2^{ème} étape, nous avons conçus trois classifieurs le premier est neuronal, le deuxième est flou, et le troisième est neuro-flou.

II.4.2 Génération de résidus a base de Modèle Neuronal:

Les techniques classiques sont variées mais pas toujours efficaces. En effet, les procédés physiques sont bien souvent des systèmes dynamiques complexes, présentant de fortes non linéarités dans leur fonctionnement. Des simplifications s'imposent pour formuler un modèle exploitable. Ces méthodes dégradant les performances du modèle mathématique.

Une autre approche réside dans la modélisation neuronale des procédés. Le but reste de concevoir un modèle en vue de la génération de résidus.

Le concept général de la génération de résidus reste le même que pour les modèles analytiques. Il consiste à comparer les sorties du procédé à leurs estimées. Mais dans ce cas, les estimées sont calculées par un modèle neuronal.

Le vecteur de résidus $r(t)$ est calculé par la différence entre le vecteur de sortie d'actionneurs $y(t)$ et le vecteur de sortie du modèle neuronal $\hat{y}(t)$ (Figure II.18).

$$r(t) = y(t) - \hat{y}(t) \quad (\text{II.26})$$

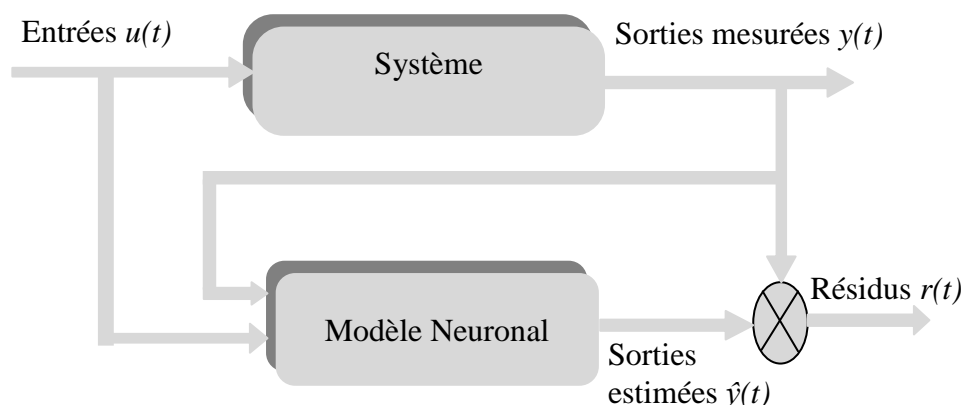


Figure II.18: Génération de résidu par le modèle neuronal

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

Cette base de données sera partager en deux, une partie majeure sera utilisée pour l'apprentissage, et l'autre sera nécessaire pour la validation.

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

- *Choix de la structure du réseau de neurones:* Généralement on opte pour la structure d'un réseau de neurones multi-couches MLP (Multi-Layer Perceptron), qui est une structure convenable pour les modèles non linéaires.

Un réseau de neurones à deux couches (une couche cachée et une couche de sortie) est largement suffisant pour l'identification d'un système dynamique non linéaire multi-variable (plusieurs entrées, plusieurs sorties). La couche cachée est constituée par plusieurs neurones, dont le nombre est choisi de façon expérimentale. Les fonctions généralement choisies pour ces neurones sont de type sigmoïde. Les sorties du réseau sont les sorties estimées du système. Les fonctions d'activation choisies pour les sorties sont généralement de type linéaire.

- *Apprentissage:* Les poids ainsi que les biais sont initialement choisis de façon aléatoire, puis adaptés par un algorithme d'apprentissage, de façon à minimiser l'erreur quadratique. L'algorithme généralement utilisé comme dans le cadre de notre application est l'algorithme de Levenberg- Marquardt.

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

II.4.3 Classification:

Une fois les résidus générés, la deuxième étape consiste à les évaluer dans le but de classifier les fautes diagnostiquées. Pour cela nous allons nous intéresser, à des techniques d'évaluation ou de classification à base de RN, LF, et de NF.

- *Classification par RN:* Le classifieur neuronal se fait à base de réseaux de neurones de type MLP, après l'apprentissage du classifieur avec les résidus comme signaux d'entrée et les

fautes comme signaux de sortie, il sera capable de localiser un défaut, dans le cas de l'occurrence de se dernier.

L'avantage de l'utilisation des réseaux de neurones dans la classification est qu'ils peuvent construire des frontières de décision linéaires ou non linéaires entre les différentes classes d'une manière non paramétrique et offrent une méthode pratique pour résoudre les problèmes complexes de classification. La procédure est la même que celle citée auparavant pour l'obtention d'un modèle neuronal.

- *Classification par la LF*: L'évaluation de résidus par l'intermédiaire de la logique floue a été proposée par Frank et al. [Fra, 89]. Le résultat de l'inférence donnera des signaux indiquant la présomption de défauts, mais à l'aide de termes linguistiques. La procédure comporte deux étapes:

- Génération d'une base de connaissance traduisant les relations défauts – résidus.
- Représentation floue des résultats afin d'être exploités par la suite par un opérateur humain.

L'idée est de constituer une base de connaissance afin de pouvoir déterminer l'inférence floue, sous hypothèse de consistance de cette base de connaissance. Les règles sont du type :

$SI < r11 = Grand > ET < r22 = Petit > ALORS < défaut m = Grand >$

$SI < défaut1 = Grand > ET < défaut2 = Petit > ET < défaut m = Petit > ALORS < cause = défaut1 >$

Où *Grand*, *Petit* sont des ensembles flous connus par expérience et *défaut m* représente la $m^{ème}$ défaillance. Elle est représentée par son degré d'appartenance au sous-ensemble flou *Défaut*. A partir de cette information floue, il est facile d'intégrer une localisation floue.

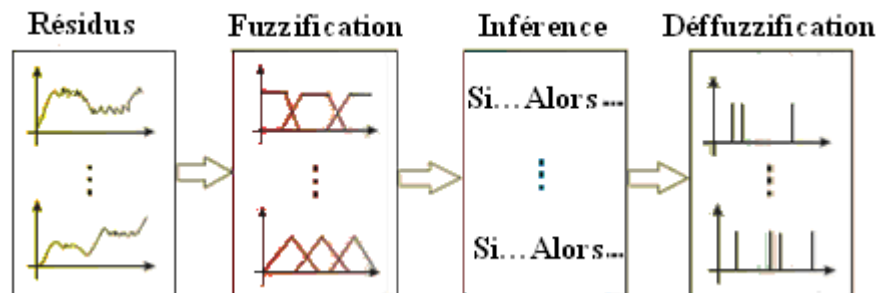


Figure II.19: Classification par LF.

L'avantage principal d'utiliser un système d'inférence floue pour la classification est sa capacité à intégrer les connaissances expertes.

- *Classification par la technique NF*: L'objectif de ce classifieur est de générer des règles floues à partir d'un ensemble d'échantillons dont la partition est déjà connue. Les règles floues sont des propositions de type : « si x_1 est μ_1 et x_2 est μ_2 et ... x_n est μ_n alors l'individu appartient à la classe i » - ($\{x_1, x_2, \dots, x_n\}$ étant le vecteur représentant l'individu examiné). La tâche du modèle est de découvrir ces règles ainsi que de définir les sous-domaines flous de l'espace des critères μ_i . Pendant la phase d'apprentissage, le classifieur neuro-flou modifie sa structure interne pour refléter le rapport entre les entrées et les sorties dans l'ensemble de la formation (base de connaissance).

L'exactitude du classifieur neuro-flou est vérifiée après que le cycle d'apprentissage soit complet en utilisant un ensemble séparé d'entrées et sorties appelé ensemble de validation (Figure II.20).

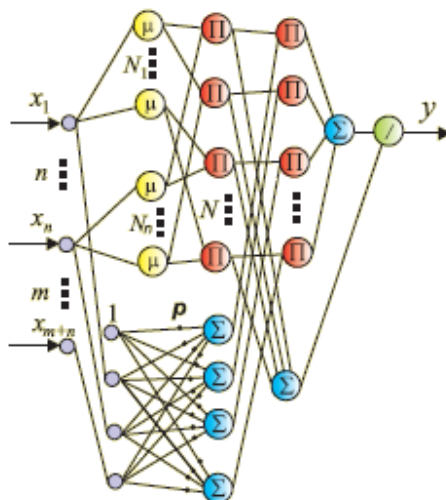


Figure II.20: Architecture d'un classifieur NF.

II.5 Conclusion

Nous avons présentés tout d'abord dans ce chapitre un aperçu sur les RN, la LF, et le NF. On a vu qu'à partir d'un modèle simple des neurones biologiques, on peut construire un modèle plus complexe, celui des perceptrons multi-couches. Ces outils permettent de calculer des fonctions complexes, adaptables à un ensemble d'exemples par le biais d'algorithmes d'optimisation utilisant la technique de la rétro-propagation. De cette façon, on peut espérer faire apprendre une fonction complexe à un *MLP* simple, contenant assez peu de neurones. On obtiendra ainsi une modélisation analytique compacte d'une fonction obtenue expérimentalement.

La logique floue ouvre des possibilités remarquables pour la codification des connaissances des experts. L'utilisation faite par la logique floue d'expression du langage courant permet au système flou de rester compréhensible pour les personnes non expertes.

Les réseaux de neurones avec leurs capacités à s'adapter à des situations inconnues par le biais de l'apprentissage, et la logique floue avec sa capacité de modélisation de connaissances imprécises et gestion de l'incertitude, ont été combinés dans un système appelé réseau neuro-flou. Ensuite, nous avons présenté leurs applications dans le domaine du diagnostic. Dans le chapitre suivant on donnera une description du système qui nous intéresse, sur lequel nous appliquerons nos techniques de diagnostic. Les résultats de ces simulations seront présentés et commentés dans les prochains chapitres.

Chapitre III

Méthodes intelligentes pour la détection des défauts

Sommaire

III.1	Introduction	59
III.2	Conception d'observateurs intelligents.....	60
III.2.1	Modèle neuronal des comportements sains.....	60
III.2.2	Modèle neuro-flou des comportements sains.....	60
III.3	Détection par analyse des résidus.....	65
III.4	Application de la détection de défauts au Benchmark DAMADICS.....	66
III.4.1	Description du système	66
III.4.2	Les défauts et leurs caractéristiques	66
III.4.3	Modèle Neuronale de comportement normal.....	71
III.4.4	Modèle Neuro-flou de comportement normal.....	76
III.4.5	Génération des résidus et détection de défaut	82
III.4.6	Comparaison des résultats et discussion	89
III.5	Conclusion.....	90

III.1 Introduction

Le contrôle tolérant aux pannes et les problèmes de fiabilité des systèmes et des procédés industriels technologiques nécessitent le développement des approches pour la détection et la localisation des défauts (FDI). L'objectif principal de la détection et de la localisation rapides des défauts est de fournir des alertes précoces aux opérateurs, de telle sorte que des mesures appropriées peuvent être prises pour prévenir les pannes du système après l'apparition de ces défauts. En conséquence, les méthodes FDI contribuent à éviter les pannes et les dommages matériels des systèmes. La méthode classique pour détecter les défauts consiste à vérifier les variables mesurables d'un système avec une certaine tolérance sur les valeurs normales et déclenche des messages d'alarme si les tolérances sont dépassées.

Durant ces dernières années de nombreuses investigations ont été faites en utilisant des approches analytiques pour les FDI, basées sur des modèles mathématiques. Parmi les méthodes à base de modèle on peut citer: l'estimation des paramètres, l'espace de parité et la conception des observateurs qui sont les techniques les plus souvent appliquées [Wit, 07], [Del, 01], [Sod, 89], [Bou, 04], [Mru, 08]. Malheureusement, ces approches nécessitent une description mathématique des systèmes considérés. Cette description n'est pas si souvent disponible dans la pratique du génie industriel. Afin de résoudre ce problème, les stratégies d'identification du système peuvent être appliquées [Gup, 03], [Nel, 01]. L'une des approches les plus populaires pour l'identification des systèmes non linéaires est basée sur l'application des réseaux neuronaux artificiels (RNA) [Sco, 03], [Tan, 07], [Ura, 07], [Vie, 04], [Bar, 06], [Ise, 05], [Ise, 06]. Les RNA sont des modèles de calcul qui possédant des propriétés particulières telles que la capacité d'apprendre, la simplicité de mise en œuvre, la généralisation et les propriétés de bonne approximation.

Le but de la détection et du diagnostic de défauts est de prévenir l'opérateur en cas d'occurrence d'un défaut dans le système surveillé, de déterminer le type et l'ampleur du défaut et de localiser les défauts. Les procédures de détection et de diagnostic sont basées sur les symptômes observés et sur les connaissances à priori sur le processus. Ainsi, les entrées d'une base de connaissances du système sont observées et mesurées et les défauts pertinents du système, connus à priori doivent être surveillés [Kor, 04], [Bas, 93]. Ensuite, le diagnostic des défauts (actionneurs, capteurs, ou inhérents au système) peut être réalisé soit par un système de surveillance et de diagnostic à distance, soit par l'utilisation d'un système de diagnostic intelligent locale et d'auto-validation des méthodes citées dans ces travaux [Yan, 99], [Hen, 00], [Tom, 02], [Gom, 99]. De telles techniques d'auto-validation et de surveillance d'état sont largement utilisées pour de nombreuses applications dans divers domaines. En générale la difficulté des techniques FDI se situe au niveau du diagnostic précoce. Même si la détection est généralement obtenue avec des algorithmes en ligne, le diagnostic nécessite de plus amples informations (par exemple des symptômes) résumées selon les données recueillies puis traitées selon des techniques hors lignes [Kou, 10a], [Kou, 10b], [Bla, 03].

Notre intérêt dans ce travail, sera porté sur le diagnostic des systèmes complexes décrits par des modèles non linéaires, nous débutons notre contribution par une description des démarches menant au diagnostic des systèmes. Entre autre, nous détaillons la description des particularités associées à différents types de modèles réalisés. En effet, le diagnostic (la détection et l'isolation de défauts) nécessite une succession d'étapes imbriquées. Une étude bibliographique détaillée permet une présentation de chacune d'entre elles. Ainsi, ce chapitre se focalise sur les problèmes de détection, tandis que le diagnostic de défauts (hors ligne et en ligne) en utilisant des modèles non linéaires conçus à l'aide des réseaux de neurones sera traité par la suite. La détection de défauts en ligne est basée sur la conception et l'analyse des résidus avec l'estimation du moment

d'apparition du défaut. L'étape d'isolation du défaut en ligne se base sur des résultats de l'analyse d'une banque de ces résidus. Ces résidus sont obtenus à partir de l'application de différents types de défauts et sont rangé dans une base de données des défauts candidats. Ces modèles fonctionnent en continu et sont mis à jour par les temps estimés d'apparition des défauts. Puis les résultats de diagnostic en ligne sont obtenus à partir d'une comparaison des résidus en fonction de la distance euclidienne. L'efficacité de cette approche est illustrée par les résultats obtenus en simulation sur un actionneur du benchmark DAMADICS.

III.2 Conception d'observateurs intelligents

III.2.1 Modèle neuronal des comportements sains:

A la fin des années 1960 jusqu'au début des années 1970, les systèmes experts ont commencé à émerger comme une branche de l'Intelligence Artificielle (IA). Dans les années 1980, les systèmes experts issus des laboratoires et des applications commerciales développées grâce au logiciel puissant pour le développement de systèmes experts ainsi que les nouvelles possibilités du matériel. E. Feigenbaum [Fei, 82], définit un système expert comme «Un programme informatique intelligent qui utilise des procédures de connaissances et d'inférence pour résoudre les problèmes qui sont déjà assez difficile et qui exigent une importante expertise humaine pour leur solution». Un système expert effectue un raisonnement en plus des représentations de la connaissance humaine et fait des calculs numériques sur des données récupérées en utilisant la base de connaissances et le moteur d'inférence séparément. En plus un système expert résout des problèmes en utilisant les connaissances heuristiques plutôt que les relations formulées avec précision dans des formes qui reflètent plus précisément la nature de la connaissance la plus humaine face à des valeurs symboliques et des procédures.

Le diagnostic est l'un des principaux domaines où les systèmes experts trouvent leurs applications à partir de leurs premiers stades. L'approche basée sur la connaissance de détection et d'isolation, a reçu une attention considérable au cours de cette dernière décennie [Tza, 89]. Les premières techniques de diagnostic de défaut basées sur les systèmes experts ont été développées au début des années 1970 par MIT (*Massachusetts Institute of Technology*), et rapportées par Scherer et White [Sch, 89]. Depuis de nombreux systèmes ont été construits, les enquêtes sur les premiers systèmes de diagnostic experts et des procédés technologiques sont fournies par Pau [Pau, 86], Tzafestas [Tza, 89], Scherer et White [Sch, 89].

Les réseaux de neurones trouvent leurs applications dans la détection des défauts grâce à leurs capacités de reconnaissance des formes. Les RNA sont conçus pour apprendre à partir de la présentation des exemples pour former une représentation interne du problème. Pour le diagnostic, il est nécessaire de relier les mesures des capteurs aux causes de défauts, et la distinction entre les états normaux et anormaux.

La modélisation est une étape nécessaire pour la compréhension et l'analyse de la dynamique de fonctionnement d'un système industriel. Elle a pour objectif d'établir une représentation des différents modes de fonctionnement du procédé et d'estimer les modèles dynamiques associés à ces modes.

Choix d'utilisation des réseaux de neurones: La modélisation du comportement des systèmes complexes sous forme mathématique est plus au moins délicate. Une alternative est de choisir une technique de modélisation basée sur les Réseaux de neurones. L'algorithme qui nous permet de sélectionner un modèle neuronal robuste, est obtenu selon les étapes suivantes :

- Connaissance approfondie sur le système à identifier.

- Le choix des entrées et des sorties du modèle neuronal.
- Détermination du comportement interne du modèle neuronal (le nombre de couches cachées et le nombre des neurones dans chaque couche cachée).
- Choix des fonctions d'activation.
- Choix de la précision et du nombre d'itération proposés.

Par la suite nous considérons des systèmes dynamiques avec des entrées $u_i(t)$ et des sorties $y_k(t)$ ($i=1..q, k=1..n$) en supposant que les variables d'état ne sont pas mesurables. Ces systèmes se présentent souvent sous forme complexe dynamique et non linéaire. En conséquence, les modèles basés sur la connaissance ne sont pas faciles à obtenir. Les approches basées sur les RNA consistent à utiliser des modèles basés sur les données [Mok, 08], [Zha, 08]. L'objectif est de concevoir des modèles des comportements sans défaut et défectueux qui seront utilisés pour la génération automatique des résidus (Figure III.1).

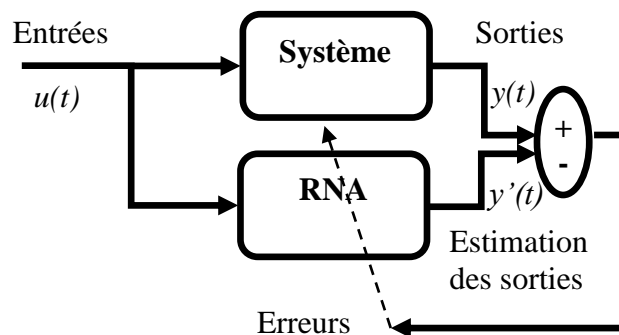


Figure III.1: Conception du modèle a base de données

Afin d'obtenir la meilleure architecture des RNA, plusieurs configurations sont testées par essai-erreur ou par utilisation d'algorithmes spéciaux (OBS : Optimal Brain Surgeon, ou OBD : Optimal Brain Damage) pour la détermination du nombre de neurones par couche cachée et du nombre de couches cachées. Par suite l'apprentissage du RNA est effectué par l'algorithme de Levenberg-Marquardt avec un arrêt précoce (Early stopping). Cet algorithme est connu pour sa convergence rapide. Lors de l'étape d'apprentissage, le RNA apprend avec des données collectées pendant le fonctionnement normal (sans-défaut du modèle). Puis les modèles de référence RNA sont validés avec un autre ensemble de données [25].

III.2.2 Modèle neuro-flou des comportements sains:

Dans certains cas, la modélisation floue d'un système est avantageuse par rapport à une approche conventionnelle basée sur les lois de la physique [Ped, 94]. Si les caractéristiques mathématiques du processus sont difficiles à établir, les informations à la fois imprécises et entachées d'incertitudes, s'expriment plus facilement dans un langage naturel. La représentation et la manipulation de ces données linguistiques sont alors aisément réalisées en utilisant les sous-ensembles flous. Même si ce type d'approche est délicat, il faut retenir qu'un modèle flou est un approximateur universel [Wan, 92]. Initialement, pour concevoir un modèle flou on avait besoin d'un opérateur connaissant parfaitement le comportement du système considéré. Cette démarche se heurte à de nombreux problèmes pour les systèmes complexes (systèmes d'ordre élevé mono-variables ou multi-variables). En effet, un expert peut difficilement formaliser toutes les propositions nécessaires à une approximation précise. Par conséquent, une autre approche, dans

laquelle le modèle flou est identifié par une méthode analytique à partir d'une base de données, peut être envisagée.

La modélisation floue suscite beaucoup d'intérêt depuis de nombreuses années et la littérature relative à ce sujet est importante. Les techniques permettant de définir la structure du modèle et le réglage des paramètres caractéristiques abondent. Toutefois, si ces travaux se différencient par rapport à la méthode employée (analytique, réseaux de neurones, algorithmes génétiques, ...), ils se répertorient facilement par rapport à la syntaxe des règles. Trois classes distinctes de modèle sont décrites dans la littérature: les modèles flous linéaires développés par Takagi-Sugeno [Tak, 85], les modèles linguistiques introduits par Zadeh [Zad, 75a], [Zad, 75b], [Zad, 75c] et les modèles aux relations floues développées par Czogala E., Pedricz W., [Czo, 81].

Les modèles flous linéaires de Takagi-Sugeno sont très efficaces pour approcher des systèmes non-linéaires complexes, car la base de règles est composée d'un ensemble de modèles locaux (généralement des fonctions linéaires). Par exemple, un système dynamique non -linéaire simple se représente par des règles de la forme:

$$R_i \text{ Si } u(k) \text{ est } U_i \text{ et } y(k) \text{ est } Y_i \text{ alors } y(k+1) = a_i y(k) + b_i u(k) + c_i$$

Où $u(k)$ est la commande; $y(k)$ la sortie du système; U_i, Y_i les sous-ensembles flous décrivant la commande et la sortie sur leur univers de discours respectif; a_i, b_i, c_i les paramètres de la conséquence. L'identification de la structure débute par le choix d'une partition de l'espace des entrées. Généralement, une méthode heuristique [Tak, 85] ou une technique de clustering [Bab, 95] est employée. Puis, il faut identifier la structure des conclusions (choix des variables de la fonction linéaire) et rechercher les paramètres de la conséquence. La plupart des méthodes classiques d'optimisation sont applicables (gradient, moindres carrés,...).

Architecture de l'ANFIS : (Adaptive Network Based Fuzzy Inference System) c'est un système d'inférence adaptatif neuro-flou qui consiste à utiliser un réseau de neurone de type MLP à 5 couches pour lequel chaque couche correspond à la réalisation d'une étape d'un système d'inférence floue de type Takagi Sugeno. Pour la simplicité, nous supposons que le système d'inférence flou à deux entrées x et y , et à une sortie f . Supposons que la base de règle contient deux règles floues de type Takagi-Sugeno.

Règle 1 : SI x est A_1 et y est B_1 ALORS $f_1 = p_1 x + q_1 y + r_1$ (III.1)

Règle 2 : SI x est A_2 et y est B_2 ALORS $f_2 = p_2 x + q_2 y + r_2$ (III.2)

L'ANFIS à une architecture composée de cinq couches comme représenté sur la figure III.2.

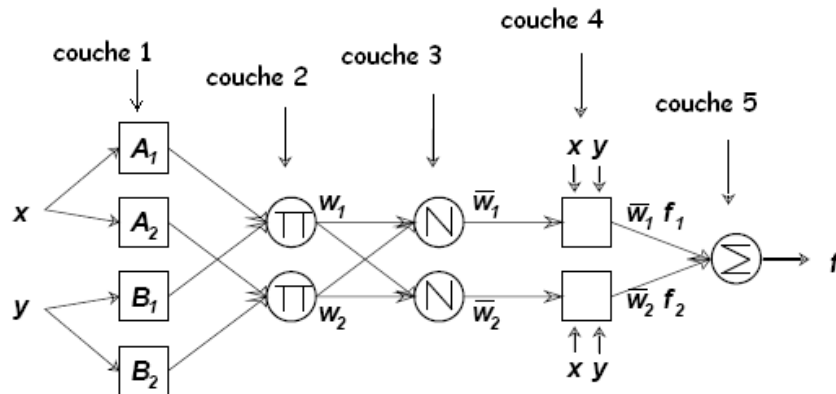


Figure II.2: L'Architecture de l'ANFIS.

Une architecture classique peut être décrite de la manière suivante :

- **La première couche:** une architecture de type ANFIS comporte autant de neurones qu'il y'a de sous ensembles flous dans le système d'inférence représenté. Chaque neurone calcule le degré de vérité d'un sous ensemble flou particulier par sa fonction de transfert. La seule restriction sur le choix de cette fonction concerne sa dérivabilité. En retrouve dans la littérature, l'utilisation, de fonctions gaussiennes et les paramètres modifiables sont le centre et la pente de la gaussienne (variance).

La fonction d'activation des neurones i de la première couche :

$$f_i^1 = \mu_{A_i}(x) \quad (\text{III.3})$$

telle que x est l'entrée au neurone i , et A_i est un sous ensemble flou correspondant à la variable x . En d'autres termes, f_i^1 est la fonction d'appartenance du sous ensemble A_i et il indique le degré de quantification auquel une donnée x satisfait. Nous choisissons $\mu_{A_i}(x)$ selon une forme souhaitée (Gaussien, triangle, trapézoïdal) avec le maximum égal à 1 et le minimum égal à 0, tel que les fonctions généralisée de ces formes sont :

$$\text{Triangle : } \mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (\text{III.4})$$

$$\text{Trapézoïdale : } \mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (\text{III.5})$$

$$\text{Gaussienne : } \mu(x) = \exp\left(-\frac{(x-c)^2}{\sigma^2}\right) \quad (\text{III.6})$$

où $\{a, b, c, \sigma\}$ est l'ensemble des paramètres. Pendant que les valeurs de ces paramètres changent, les fonctions d'appartenances précédente changent en conséquence, à cet effet, diverses formes de fonction d'appartenance sur la variable linguistique A_i sont présentées. Les paramètres dans cette couche sont désignés sous le nom des paramètres de fonction d'appartenance.

La deuxième couche cachée: sert à calculer le degré d'activation des prémisses. Les neurones de cette couche représentent chacun la prémisse d'une règle. Ils reçoivent en entrée le degré de vérité des différents sous-ensembles flous composant cette prémisse et ont en charge le calcul de son propre degré de vérité. Les fonctions d'activation utilisées pour ces neurones dépendant des opérateurs présents dans les règles (*ET - OU*).

La fonction d'activation des neurones i de la première couche :

$$W_k = \mu_{A_i}(x) \times \mu_{B_j}(y) \quad (\text{III.7})$$

Ou k : représente le nombre de règle, i : représente le nombre de partition de x , et j : le nombre de partition de y .

La troisième couche cachée: normalise le degré d'activation des règles. Chaque neurone dans cette couche est représenté par un cercle noté N . le $i^{\text{ème}}$ neurone calcule le rapport entre le $i^{\text{ème}}$ poids des règles et la somme de tous les poids des règles. Cette opération est appelée **la normalisation des poids**.

$$\overline{W}_k = \frac{W_k}{\sum W_i} \quad (\text{III.8})$$

L'ensemble des sorties de cette couche seront appelées les poids normalisé.

La quatrième couche cachée: sert à déterminer les paramètres, la partie conséquence des règles (p, q, r). La fonction de chaque neurone dans cette couche est la suivante

$$f_k^4 = \overline{W}_k \times f_k = \overline{W}_k \times (p_k x + q_k y + r_k) \tag{III.9}$$

Où \overline{W}_k est la sortie de la troisième couche, et $\{p_i, q_i, r_i\}$ est l'ensemble des paramètres. Ces paramètres sont désignés sous le nom de paramètres conséquents. [8]

La couche de sortie: contient un seul neurone représenté par un cercle noté S qui calcule la sortie globale comme addition de tous les signaux entrants, c'est-à-dire :

$$f^5 = \sum_k \overline{W}_k \times f_k^4 \tag{III.10}$$

La figure III.3 représente un système ANFIS, à 2 entrées chaque entrée est répartie en trois sous ensemble flou et 9 règles.

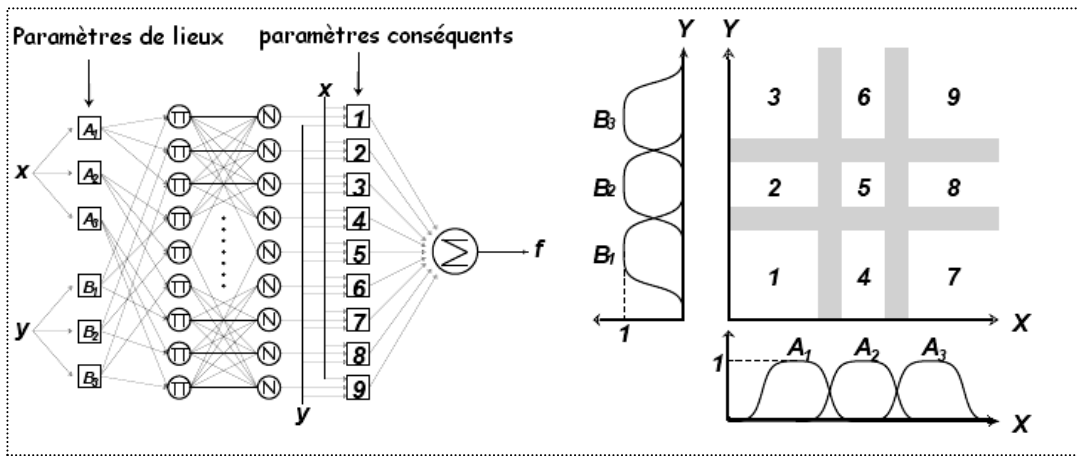


Figure III.3: Exemple d'ANFIS à 2 entrées avec 9 règles.

Tableau III.1: les différentes couches d'un système ANFIS

Les différentes couches	Type des couches	Le nombre de neurones dans la couche
Couche 0	Les entrées	n
Couche 1	Les valeurs	$(p.n)$
Couche 2	Les règles	p^n
Couche 3	La normalisation	p^n
Couche 4	Linéarisation des fonctions	p^n
Couche 5	Somme	1

Avec :

n : le nombre des entrées.

p : le nombre des sous ensembles flous d'entrée (partition flou).

Noter que les neurones dans un ANFIS ont différentes structures:

- Valeurs [fonction d'appartenance définie par différentes formes].
- Règles [habituellement produit].
- Normalisation [division de somme et d'arithmétique].

- Fonctions [régressions linéaires et multiplication avec \bar{w} , tel que \bar{w} est la normalisation du poids w].
 - La sortie [Somme Algébrique].
- **Les avantages de L'ANFIS:** Cette technique possède des avantages stratégiques:
- ✓ Exploitation de la connaissance disponible, grâce à la base de règles.
 - ✓ Réduction de la taille de la base de règles : il suffit d'avoir des règles générales, les détails seront fournis par le RN.
 - ✓ Réduction de la complexité de l'apprentissage : le RN doit simplement apprendre les cas particuliers ou les exceptions, pas le problème complet.
 - ✓ Efficacité immédiate dès le début de l'apprentissage et possibilité d'éviter des comportements initiaux erratiques. [Mir, 05]

III.3 Détection par analyse des résidus

Dans la littérature, les performances d'une méthode de détection et d'isolation sont évaluées en tenant compte des paramètres suivants:

- La sensibilité de détection des défauts: la capacité de la méthode pour détecter des défauts à amplitude relativement faible.
- Le retard de détection: la capacité de la méthode pour détecter les défauts avec un retard relativement court.
- Robustesse: Compétences de la méthode en présence de bruit, perturbations et d'autres défauts possible avec un nombre minimal de fausses alarmes.

Le diagnostic des systèmes dynamiques stochastiques complexes est généralement effectué en deux étapes : La génération de résidus suivit par la prise de décision. La reconstruction des défauts n'est pas toujours prise en compte dans la génération des résidus et pourtant il est très intéressant de simplifier la phase de décision par la génération de résidus reflétant directement l'amplitude des défauts, afin de permettre la conception simplifiée d'un test de détection toujours très difficile à concevoir dans le cas multi-défauts.

Le résidu $r_{10}(t)$ illustré dans la Figure III.3 est fourni a partir de l'une des méthodes utilisées pour la génération des résidus comme vue précédemment, c'est une source d'information sur les défauts pour un traitement ultérieur. La détection de défauts est basée sur l'évaluation de l'ampleur des résidus. Chaque résidu $r_{k0}(t)$, $k=1, \dots, N$, (avec k le nombre de défaut) devrait normalement être proche de zéro dans le cas sans défaut, et loin de zéro dans le cas avec défaut. Ainsi, les défauts sont détectés en fonction du seuil S_{k0} admis des résidus (un résidu unique et une seule panne sont considérés pour la simplicité). L'analyse de résidu $r_{10}(t)$ fournit également une estimation de l'instant d'apparition τ_j du défaut f_j avec un certain *Retard*. Lorsqu'on utilise plusieurs signaux de résidus, l'estimation de cet instant est donnée par:

$$\tau = \min\{\tau_k, k = 1, \dots, n\} \quad (\text{III.11})$$

Les défauts sont détectés lorsque la valeur absolue du résidu $|r_{k0}(t)|$ dépasse le seuil S_{k0} :

$$\begin{cases} |r_{k0}(t)| \leq S_{k0} : \text{aucun défaut détecté à l'instant } t \\ |r_{k0}(t)| > S_{k0} : \text{un défaut est détecté à l'instant } t. \end{cases} \quad (\text{III.12})$$

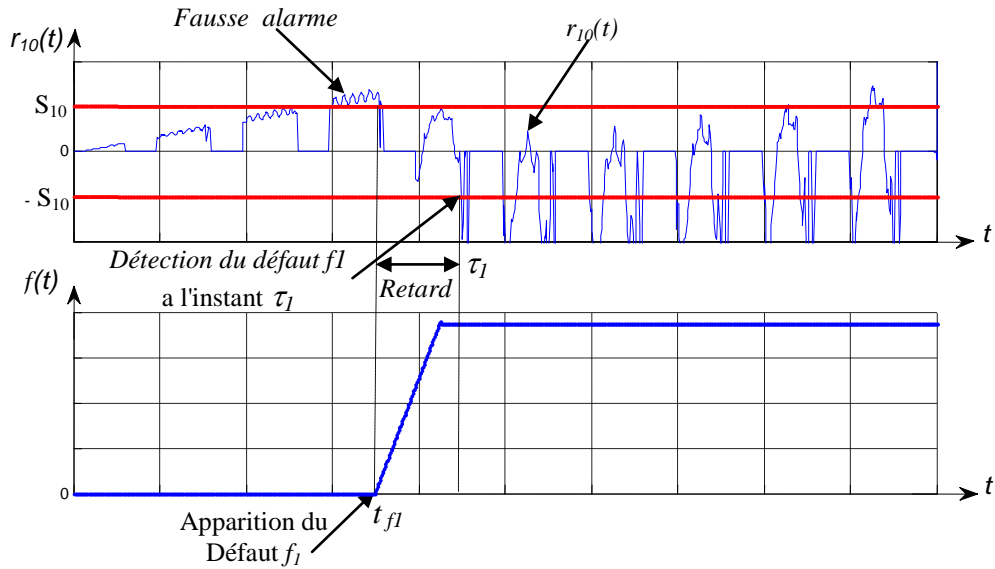


Figure III.3: Détection d'un défaut par analyse du résidu

La difficulté principale de cette évaluation est que la mesure des sorties du système $y_1(t)$ est généralement bruitée par des perturbations (par exemple, le bruit de mesure). Dans la pratique, en raison des incertitudes de modélisation et de perturbations, il est nécessaire d'affecter un seuil élevé S_{10} afin d'éviter les fausses alarmes. Cette augmentation du seuil peut engendrer la non-détection du défaut f_1 . Pour cette raison, il est nécessaire de choisir une technique adaptée pour fixer le seuil.

III.4 Application de la détection de défauts au Benchmark DAMADICS

L'actionneur DAMADICS (**D**evelopment and **A**pplication of **M**ethods for **A**ctuator **D**iagnosis in **I**ndustrial **C**ontrol **S**ystems) [8] est un élément du processus d'automatisation. Sa puissance de traitement local autorisée permet d'avoir plus d'information à traiter et donne plus de possibilités au système global. En plus des possibilités de contrôle, il crée un grand nombre de données pouvant être communiqué à plusieurs opérateurs intervenant tout au long de son cycle de vie. Les actionneurs intelligents sont en général intégrés dans un système global avec les plus hauts objectifs de la production.

L'exploitation correcte de ces objectifs est obtenue aussi longtemps que l'actionneur est capable de traiter correctement le fonctionnement demandé qui leur est communiqué par les plus hauts niveaux. L'introduction des techniques de détection et de diagnostic de défauts (FDI) peut limiter les coûts de l'inspection périodique.

III.4.1 Description du système:

Nous proposons deux étapes pour l'application de la méthode FDI. La première étape est l'élaboration d'un modèle dans le but de distinguer l'état de fonctionnement normal et anormal du système. Durant l'étape d'apprentissage, le modèle de type "boîte noire" apprend le comportement du système. Les signaux d'entrée et de sortie du système sont utilisés pour générer les résidus. En général, en régime de fonctionnement normal, ce signal est statistiquement nul et s'écarte notablement de zéro en présence de défauts d'où la détection. La deuxième étape consiste en l'élaboration ou la conception d'un classifieur pour l'isolement et l'identification du défaut.

Présentation: Dans le cadre de notre thèse. Nous allons étudier les défauts susceptibles de survenir au niveau d'un actionneur électropneumatique, cet actionneur est un élément de commande, dans l'usine de sucre à Lublin, Pologne (Figures III.4 et III.5). Il constitue un Benchmark mis à la disposition de différents groupes de recherche à travers le monde, sur lequel ils pourront tester différentes techniques de diagnostic. Le schéma fonctionnel est représenté dans par la Figure III.6.

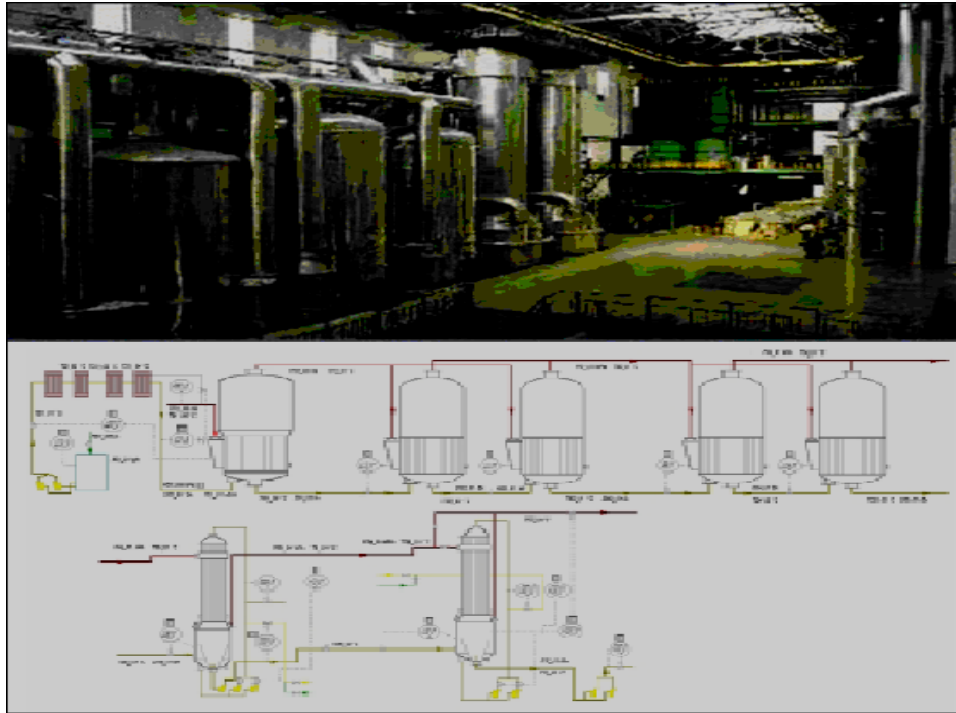


Figure III.4: l'usine de sucre à Lublin, Pologne.

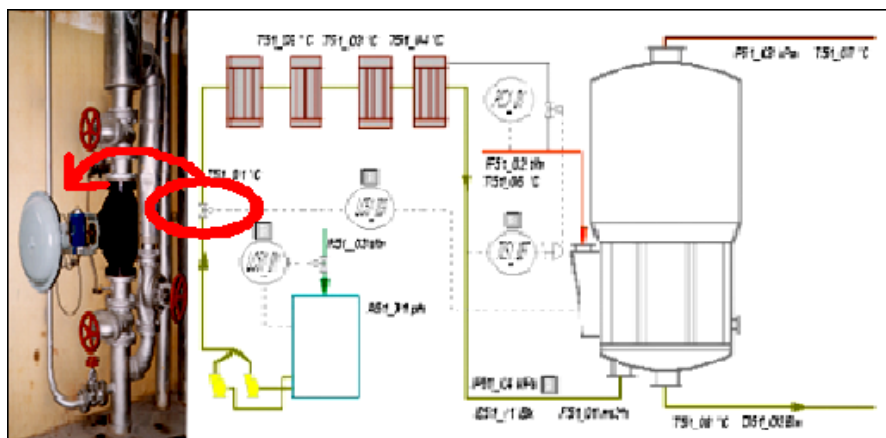


Figure III.5: Emplacement de l'actionneur dans l'usine.

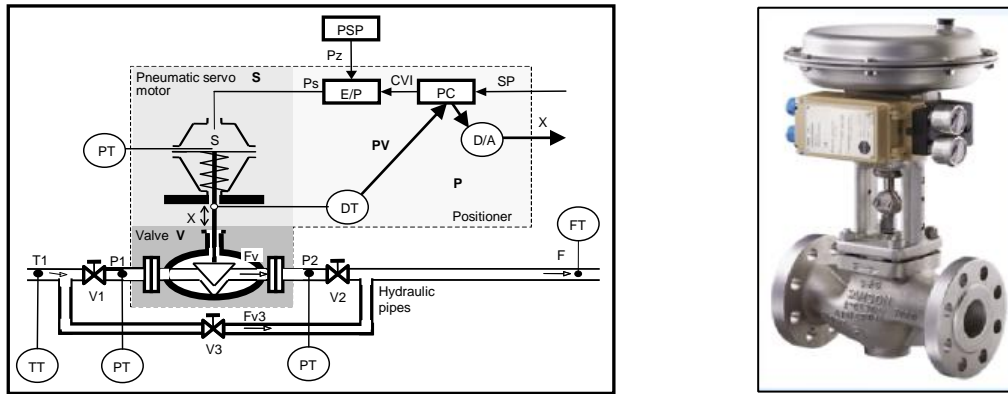


Figure III.6 : Actionneur DAMADICS et leur schéma de fonctionnement

Le bloc de cet actionneur considéré comme un système a quatre entrées et deux sorties comme le montre la Figure III.7.

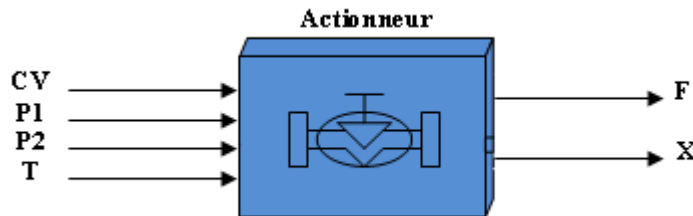


Figure III.7 : Modèle de l'actionneur

Avec les entrées U et les sorties Y du Tableau III. 2 sont définis comme suit :

$$\begin{aligned}
 U &= f(CV; P1; P2; T1) \\
 Y &= f(F; X)
 \end{aligned}
 \tag{III.13}$$

où les variables du processus peuvent être mesurées.

Tableau III.2 : les variables de notre système et leurs caractéristiques.

Entrés	Gamme	Unité physique	Description
CV	< 0,1 >	100%	La variable de commande est le signal de sortie du contrôleur [8].
P1	-	Pa	La valeur de la pression sur l'admission de soupape de commande [8].
P2	-	Pa	La valeur de la pression sur la sortie de soupape de commande [8].
T1	-	C °	La température du milieu [8].
X	< 0,1 >	-	La valeur du déplacement de tige. [8]
F	< 0,1 >	-	Le flux moyen bruité [8]. C'est l'écoulement ou le débit du produit (jus) à la sortie de la valve.

Le modèle « Matlab/Simulink » : On a utilisé un modèle Simulink construit par « DAMADICS Actuators Benchmark Library » plus simplement appelés « DABLIB » (Figure III.9).

La validité du modèle a été vérifiée par Louro [Lou, 03] à l'aide des mesures réelles fournies par le personnel de l'usine de sucre. Ces mesures contiennent les informations liées aux comportements sains et défectueux.

A défaut d'avoir toutes les données nécessaires dans le cas de la simulation réelle, il est nécessaire d'utiliser un modèle synthétique image du procédé réel. Ce dernier modèle permet la prise en compte en simulations de tous les défauts susceptibles d'apparaître dans l'actionneur.

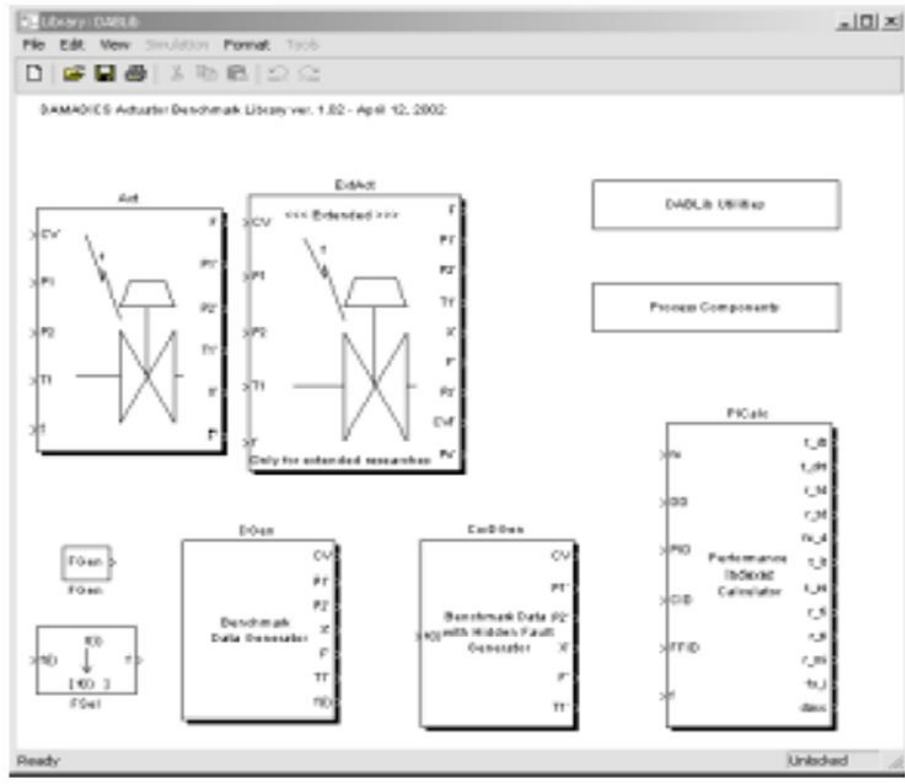


Figure III.8: Les composants du modèle DABLIB.

III.4.2 Les défauts et leurs caractéristiques:

Pour générer un défaut artificiel, on doit savoir quelques caractéristiques standards (Figure III.9) :

- Le temps d'apparition du défaut t_{from} .
- L'amplitude du défaut f_{SMax} .
- Le temps de réponse du défaut t_{fd} .
- Le temps de correction du défaut t_o

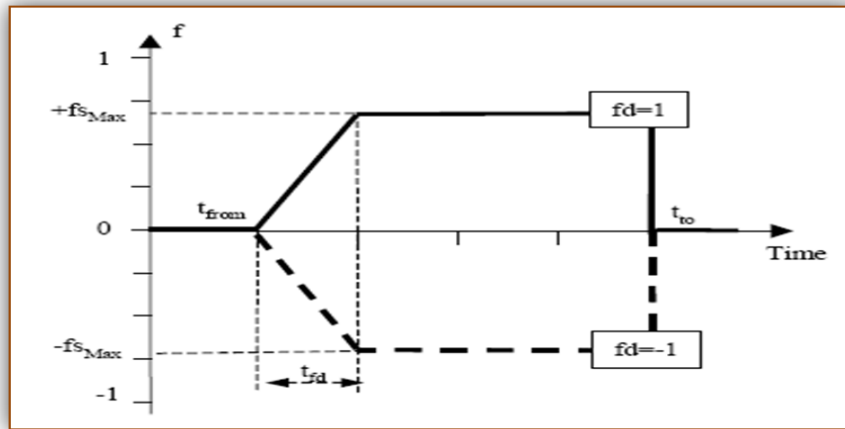


Figure III.9: L'allure d'un défaut.

Il y a deux types de défaut brusque « Abrupt » et naissant « Incipient », le premier se compose de trois types « petit », « grand » et « moyen » ; comme indiquée par la Figure III.10.

Les fautes recensées dans le cas de cet actionneur sont nommés dans le tableau III.3.

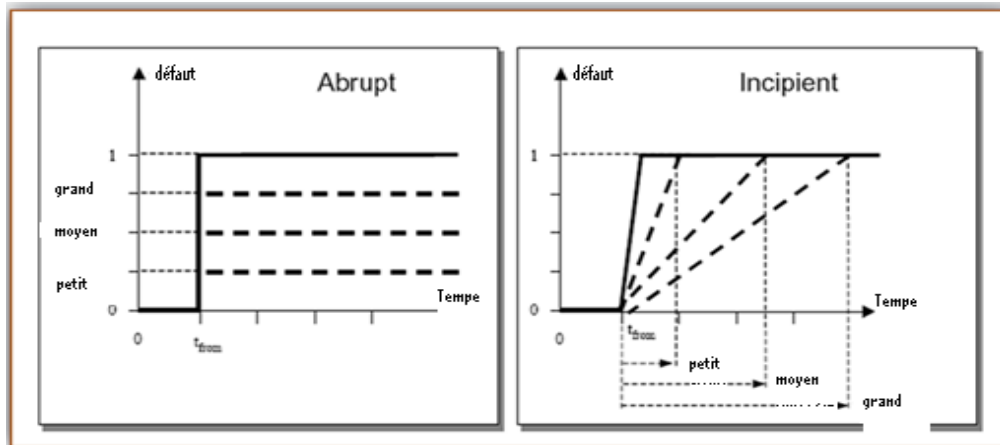


Figure III.10: Les différents types de défauts artificiels.

A partir du model SIMULINK nous avons généré tous les défauts possibles (Tableau III.3). On peut faire des essais avec les deux défauts f_1 et f_{13} avec « f_1 c'est le défaut de blocage de la Vanne » et « f_{13} c'est le défaut dans le capteur de déplacement de la tige X », les deux défauts sont de types « Abrupt Big » (brusque grand) et « Incipient Big » ; et nous avons utilisés les sorties du système pour générer les résidus.

Tableau III.3: Les défauts de l'actionneur et leur type. [Lou, 03]

Défauts	Description	Le temps de développement			
		S	M	B	Naissant
f_1	Bouchage de valve	*	*	*	
f_2	Dépôt de prise de valve ou de siège de valve			*	*
f_3	Erosion de prise de valve ou de siège de valve				*
f_4	Accru de frottement de valve ou de douille				*
f_5	Fuite externe (douille, couverture, borne perméable)				*
f_6	Fuite interne (étanchéité de valve)				*
f_7	Evaporation moyenne ou écoulement critique	*	*	*	
f_8	Tige de piston du servomoteur tordu	*	*	*	
f_9	Étanchéité des bornes du servomoteur				*
f_{10}	Perforation du diaphragme du servomoteur	*	*	*	
f_{11}	Défaut du ressort du servomoteur			*	*
f_{12}	Défaut électropneumatique du transducteur	*	*	*	
f_{13}	Défaut de la sonde de déplacement de Rod	*	*	*	*
f_{14}	Défaut de la sonde de pression	*	*	*	
f_{15}	Défaut de rotation du positionneur			*	
f_{16}	Chute de pression par le positionneur	*	*	*	
f_{17}	Changement inattendu de pression de la valve			*	*
f_{18}	Clapets de dérivation entièrement ou partiellement ouverts	*	*	*	*
f_{19}	Défaut de la sonde de débit	*	*	*	

Dans notre cas parmi les 19 défauts, nous avons choisie d'une manière aléatoire quatre défauts, f_7 , f_{10} , f_{15} et f_{17} , juste pour nous permettre de montrer l'efficacité de l'utilisation d'un modèle neuronal et neuro-flou pour la génération des résidus, et les techniques d'intelligence artificielle (Réseau de neurone, Logique floue, et les systèmes Neuro-flou) pour la classification.

Pour cela, nous avons utilisé des données réelles obtenues en 2001 par l'usine de sucre à Lublin, Pologne [8].

III.4.3 Modèle Neuronale de comportement normal:

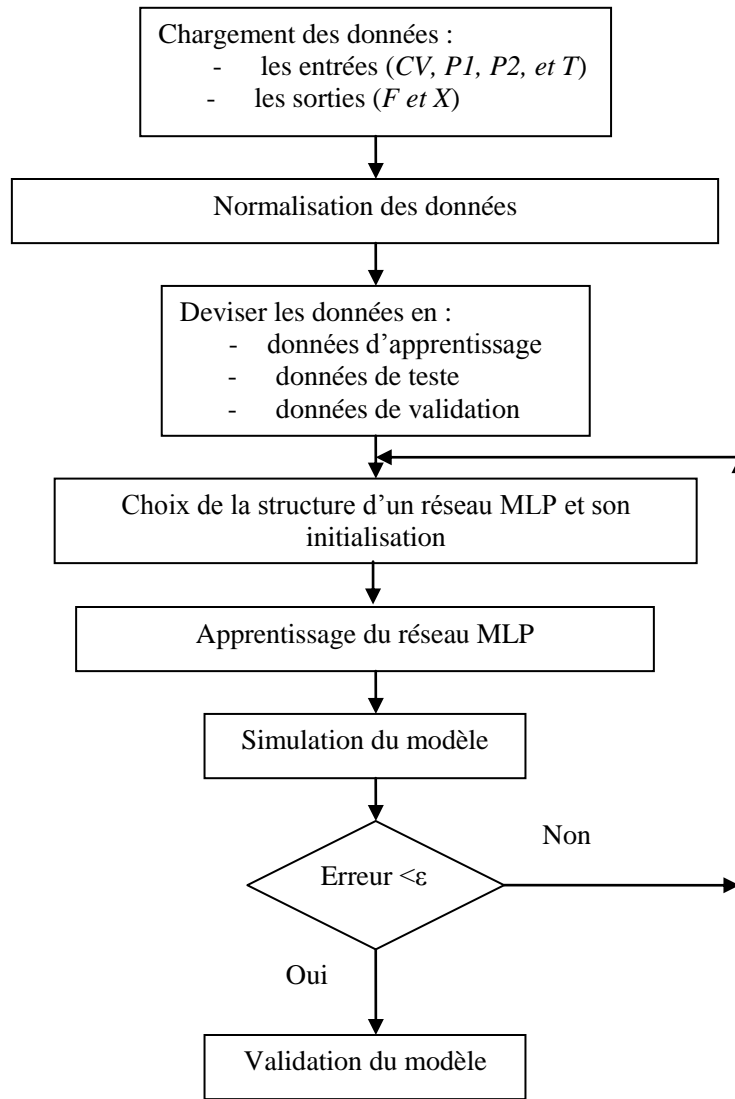
Première étape : Obtention d'un modèle neuronal: L'actionneur est un système à 4 entrées et 2 sorties. Dans cette application pour modéliser l'actionneur nous avons utilisé deux réseaux de neurones de type MLP, donnant chacun la relation entre les entrées et une sortie F ou X.

$$X = R_x(CV, P1, P2, T)$$

$$F = R_f(X, P1, P2, T)$$

(III.14)

Dans ce cas le schéma de la procédure de création du modèle se présente dans l'organigramme III.1.



Organigramme III.1: Procédure d’élaboration d’un modèle neuronale.

On a fait plusieurs essais pour obtenir l’architecture qui simule le fonctionnement de l’actionneur, ces expériences prennent beaucoup de temps, tel que le choix du nombre de couches cachées et le nombre de neurones par couches pour avoir un modèle simulant le fonctionnement du système réel avec une erreur entre la sortie système et du modèle inférieure ou égale à ϵ .

Le tableau III.4 donne les meilleures architectures expérimentées pour la modélisation de ce système industriel :

Tableau III.4: Apprentissage des réseaux optimaux obtenus.

Le réseau	Couche 1	Couche 2	Couche de sortie	MSE
netF	26	12	1	$4.03 \cdot 10^{-3}$
netX	24	12	1	$3.77 \cdot 10^{-3}$

MSE : l’erreur quadratique moyenne.

L'apprentissage est effectué par l'algorithme de Levenberg-Marquardt avec arrêt précoce (Early Stopping), pour son efficacité et sa rapidité de convergence. Les données utilisées sont des mesures réelles prélevées par les ingénieurs de l'usine considérée.

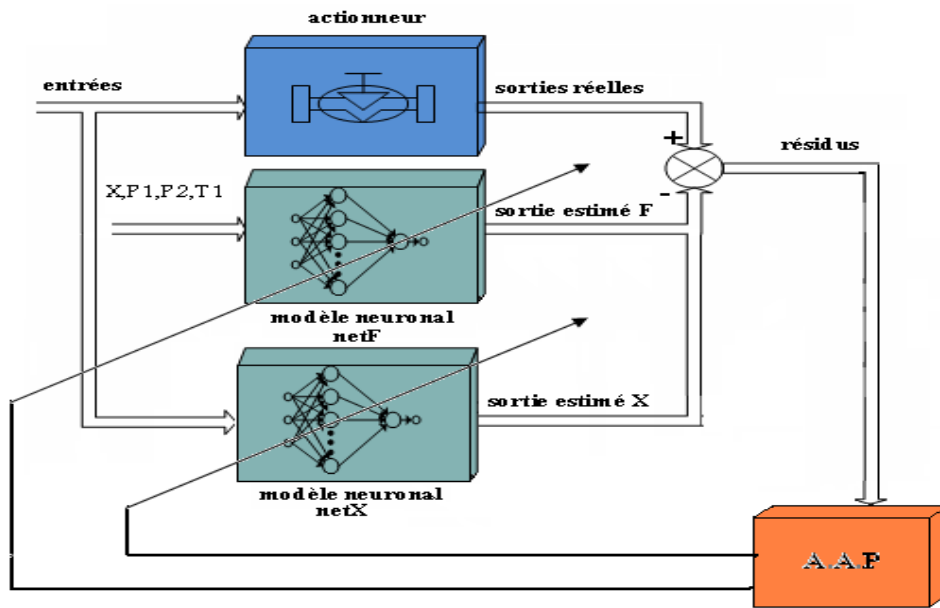


Figure III.11: Schéma générale de l'apprentissage du modèle neuronal.

L'apprentissage de ces deux modèles (*netF* et *netX*), donne des résultats illustrés dans les Figures III.12 à III.15.

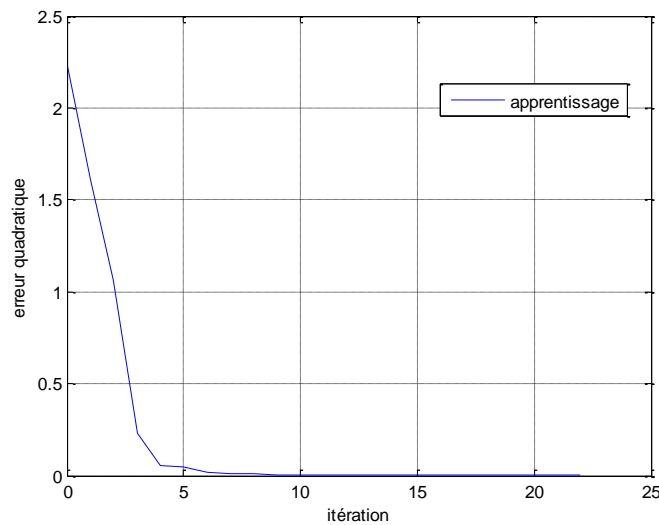


Figure III.12: L'erreur quadratique d'apprentissage du modèle *netF*.

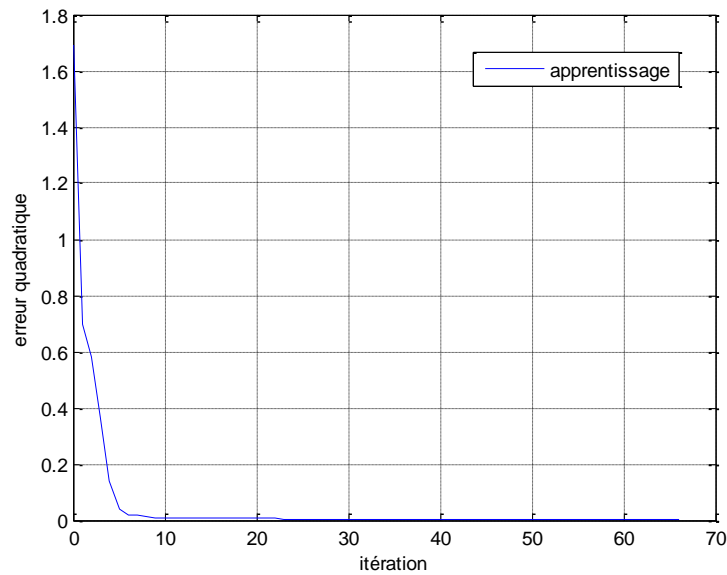


Figure III.13: L'erreur quadratique d'apprentissage du modèle *netX*.

Apprendre plusieurs fois produira des résultats différents en raison des conditions initiales différentes. La moyenne des erreurs carrées (Mean Squared Error MSE) est la moyenne des carrés de la différence entre les sorties du système et ceux du modèle.

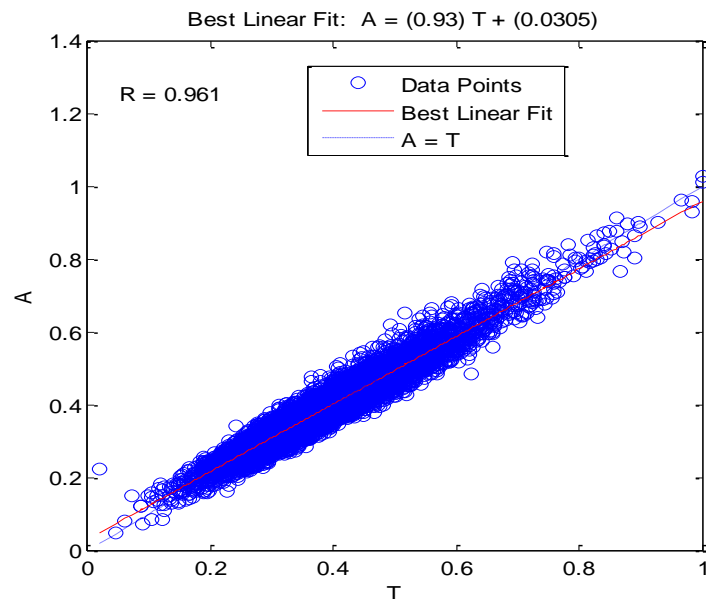


Figure III.14: La régression linéaire entre la sortie du modèle *netF* et la sortie réelle F.

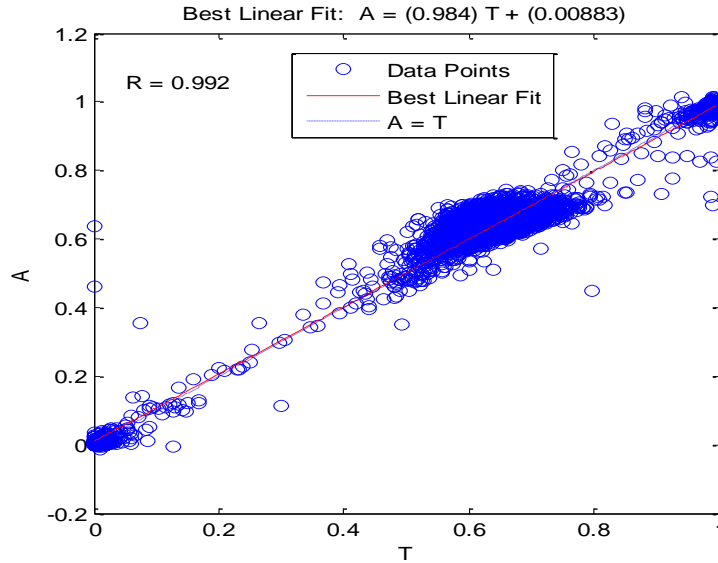


Figure III.15: La régression linéaire entre la sortie du modèle *netX* et la sortie réelle X.

La régression R des Valeurs, mesure la corrélation entre les sorties et les cibles. Une valeur de R de 1 signifie une relation parfaite, 0 une relation aléatoire.

Pour *netF* on a une régression $R=0.962$ et pour le modèle *netX* on a $R= 0.992$, dans les deux cas les sorties des modèles neuronaux suivent bien les cibles.

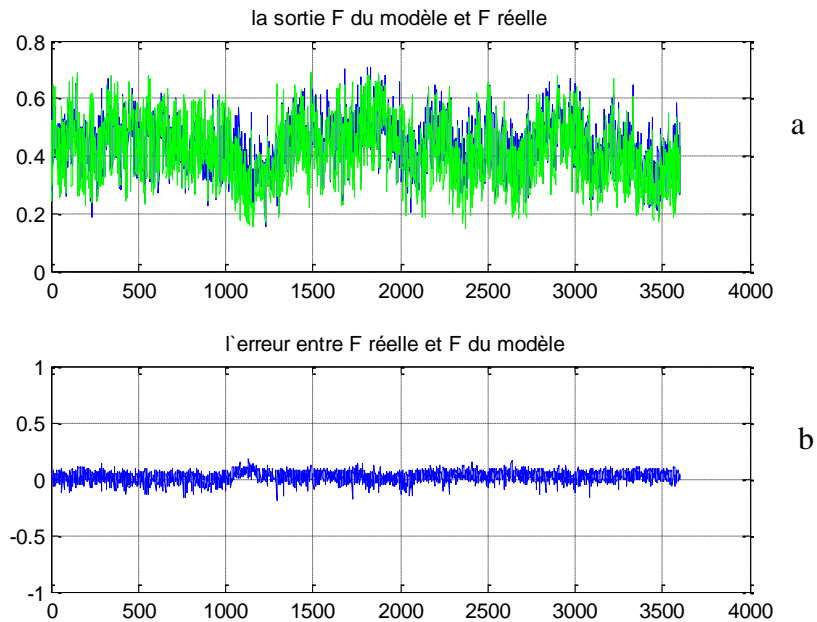


Figure III.16: (a) La sortie réelle F et la sortie du modèle *netF*. (b): l'erreur entre ces sorties.

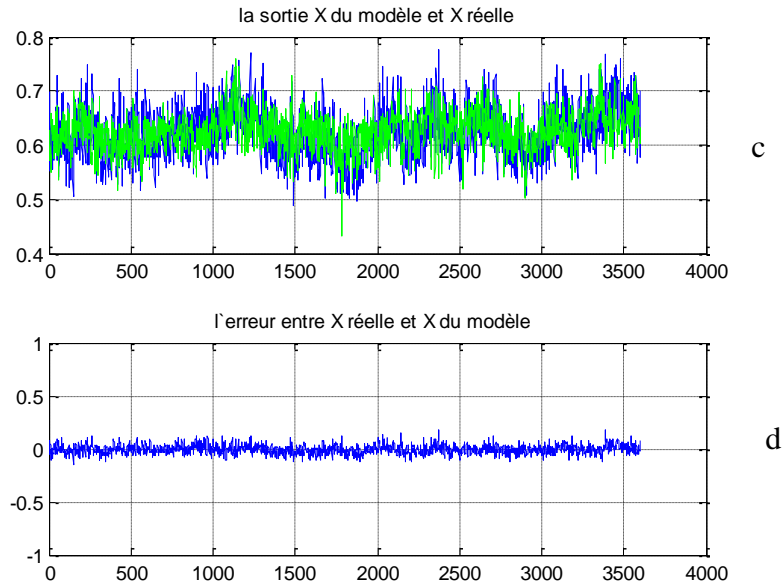


Figure III.17: (c) La sortie X et la sortie du $netX$. (d) l'erreur entre X et la sortie du $netX$.

On remarque qu'après la phase d'apprentissage des réseaux $netF$ et $netX$ soumis aux mêmes excitations, l'actionneur génère des sorties considérablement semblables aux sorties réelles F et X de ce dernier. L'erreur de sortie dans les deux cas est assez faible (acceptable), donc on peut considérer que les deux réseaux reproduisent bien la dynamique de l'actionneur.

III.4.4 Modèle Neuro-flou de comportement normal:

Nous avons besoin d'une autre technique de création d'un modèle des comportements sains pour les comparer avec les résultats des réseaux de neurone nous avons choisi les systèmes neuro-flou de type ANFIS qui ont prouvé leurs aptitudes et leur capacité de modélisation des systèmes non linéaires.

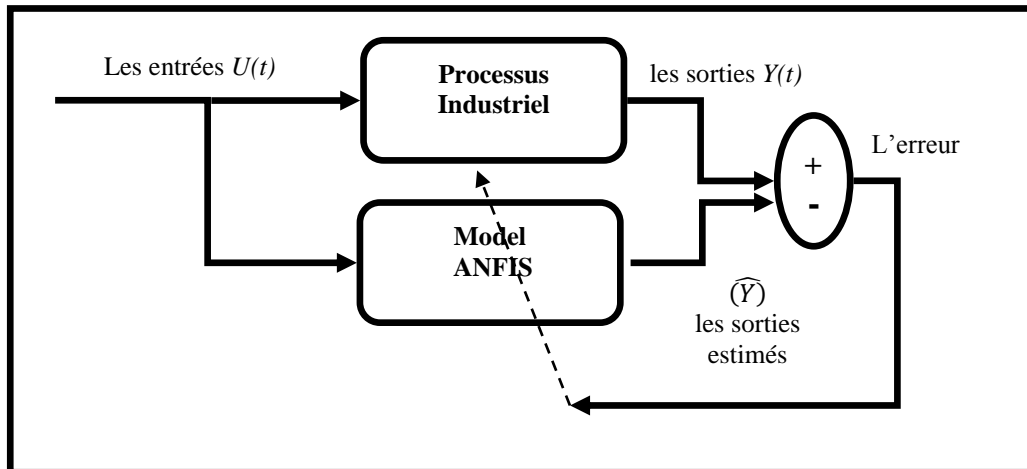


Figure III.18: Schéma général de la modélisation neuro-flou.

Les entrées et les sorties du modèle Neuro-Flou: Le système qui a été choisi est la vanne de DAMADICS [8] qui contient quatre variables comme entrées et deux comme sorties, ces

variables sont plus intéressantes pour choisir notre architecture de réseaux neuro-flous donnant un modèle plus efficace [Mir, 05].

On utilise les variables qui sont indiquée dans les Figures III.19 et III.20, du tableau III.2 et suivant le travail de Mr Mrugalski et al. [Mru, 06] sur notre actionneur, nous avons choisi un modèle composé de deux ANFIS, dont les deux équations III.15 et III.16 représentent les différentes relations entre les entrées et les sorties :

$$X = R_x (CV, P1, P2, T1) \tag{III.15}$$

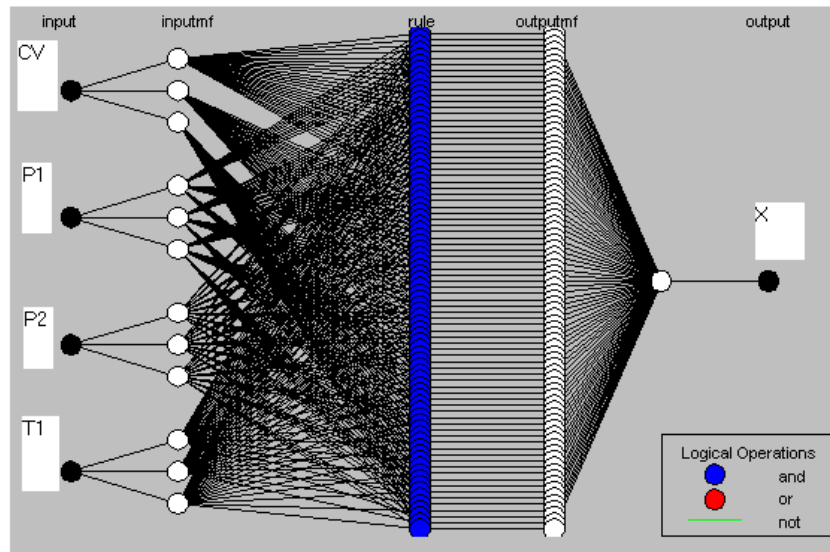


Figure III.19: Le réseau du modèle (Rx).

$$F = R_f (X, P1, P2, T1) \tag{III.16}$$

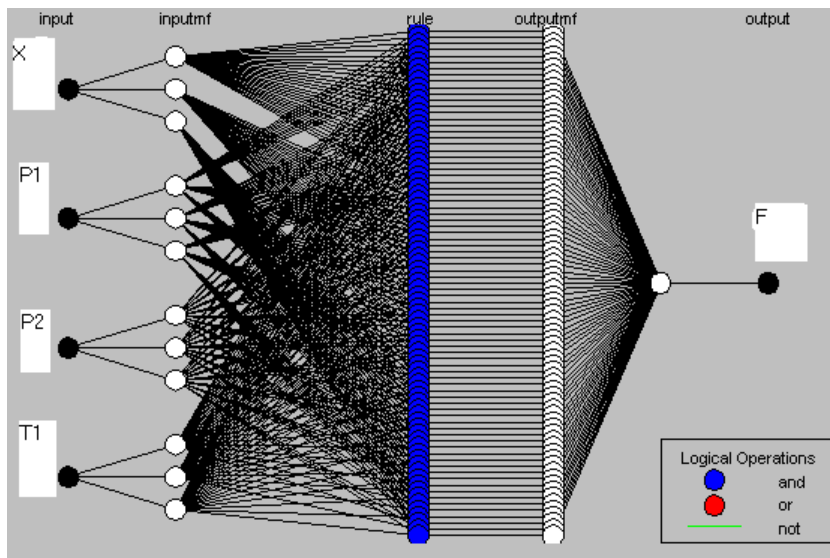


Figure III.20: Le réseau du modèle (Rf)

Les données que nous avons utilisées sont téléchargées au site web [8], ces données contiennent des jours réel des entrées et des sorties de ce système et aussi les données avec défauts.

L'architecture de l'ANFIS: Après plusieurs essais nous avons choisi l'architecture qui simule le fonctionnement de l'actionneur, ces expériences prennent beaucoup de temps, tel que le choix des données, le nombre d'ensemble flou et le type de fonctions d'appartenance.

Le premier réseau du modèle R_x , il est composé de quatre entrées CV, P1, P2 et T1; et une sortie correspondant à la valeur du déplacement de la tige (X)

- Univers de discours de l'entrée 1 : CV [0, 1]
- Univers de discours de l'entrée 2 : P1 [2000, 4 e+006]
- Univers de discours de l'entrée 3 : P2 [2000, 4 e+006]
- Univers de discours de l'entrée 4 : T1 [30, 110]
- Univers de discours de sortie : X [0, 1]

Chaque entrée est répartie en 3 ensembles flous, la forme des fonctions d'appartenance est **Gaussienne**. Donc ce modèle ANFIS est composé comme suit :

- Fonctions d'appartenances **Gaussiennes**.
- 81 règles floues.
- 30 à 100 Itération d'apprentissage.
- 05 couches parallèles.
- 12 neurones dans la première couche cachée (fuzzification).
- 81 neurones dans la deuxième couche cachée (règles floues).
- 81 neurones dans la troisième couche cachée (normalisation).
- 81 neurones dans la quatrième couche cachée (linéarisation).
- Un seul neurone dans la cinquième couche c'est la sortie.

Pour le réseau du modèle R_f , nous avons pris quatre entrées X, P1, P2 et T1; et comme sortie le Flux (F)

- Univers de discours de l'entrée 1 : X [0, 1]
- Univers de discours de l'entrée 2 : P1 [2000, 4 e+006]
- Univers de discours de l'entrée 3 : P2 [2000, 4 e+006]
- Univers de discours de l'entrée 4 : T1 [30, 110]
- Univers de discours de sortie : F [0, 1]

Chaque entrée est répartie en 3 ensembles flous la forme des fonctions d'appartenance est **Trapézoïdale**. Donc ce modèle ANFIS se compose de :

- Fonctions d'appartenances **Trapézoïdales**.
- 81 règles floues.
- 30 à 100 Itération d'apprentissage.
- 05 couches parallèles.
- 12 neurones dans la première couche cachée (fuzzification).
- 81 neurones dans la deuxième couche cachée (règles floues).
- 81 neurones dans la troisième couche cachée (normalisation).
- 81 neurones dans la quatrième couche cachée (linéarisation).
- Un seul neurone dans la cinquième couche de sortie.

L'apprentissage de l'ANFIS

L'apprentissage consiste à un entraînement du réseau :

Le réseau du modèle (R_x)

Le système ANFIS de R_x consiste de quatre entrées CV, P1, P2, T1 et la sortie X, après la fuzzification de la première couche cachée, l'ANFIS calcule le degré de chaque règle, avec une base de règles contenant 81 règles implantées dans 81 neurones de la deuxième couche cachée, la troisième couche cachée a une opération de normalisation. Tant que notre ANFIS est de type Takagi-Ségéno, la quatrième couche cachée est la sortie de réseau et donne le résultat de la somme, donc l'entraînement du réseau demande des données. Nous avons choisi 3600 échantillons à partir de la base de données existante (le choix des jours sans défauts), Alors l'apprentissage continu jusqu'à la condition d'erreur ou sur le nombre d'itérations est atteint. Ce qui donne l'ajustement des nouveaux paramètres des ensembles flous.

La sortie après l'apprentissage est présentée sur la figure suivante :

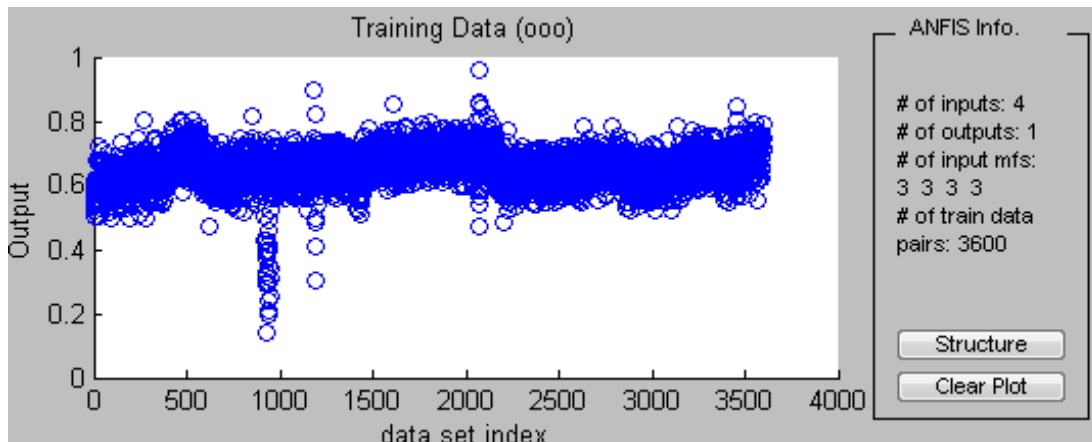


Figure III.21: la sortie d'apprentissage du modèle R_x

La phase de test comme indique son nom, consiste à un teste du réseau, c'est-à-dire la validation des résultats obtenus après la phase d'apprentissage. L'ensemble de test est composé de 3600 nouveaux échantillons à partir de la base de données existante (le choix des jours sans défauts).

La sortie de la phase de test est représentée sur la figure suivante :

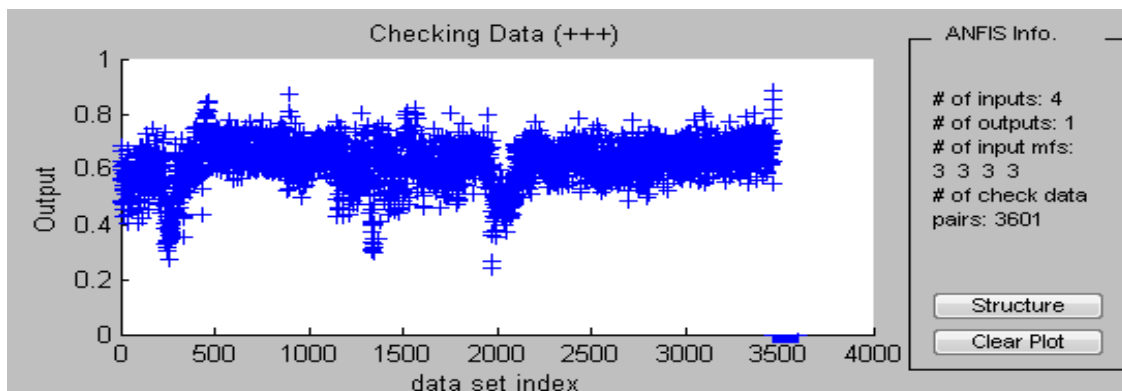


Figure III.22: La sortie de la phase de teste.

La même chose pour la phase de vérification nous avons choisi un autre ensemble de 3600 échantillons à partir de la base de données existante (le choix des jours sans défauts).

La sortie de la phase de vérification est présentée sur la figure suivante :

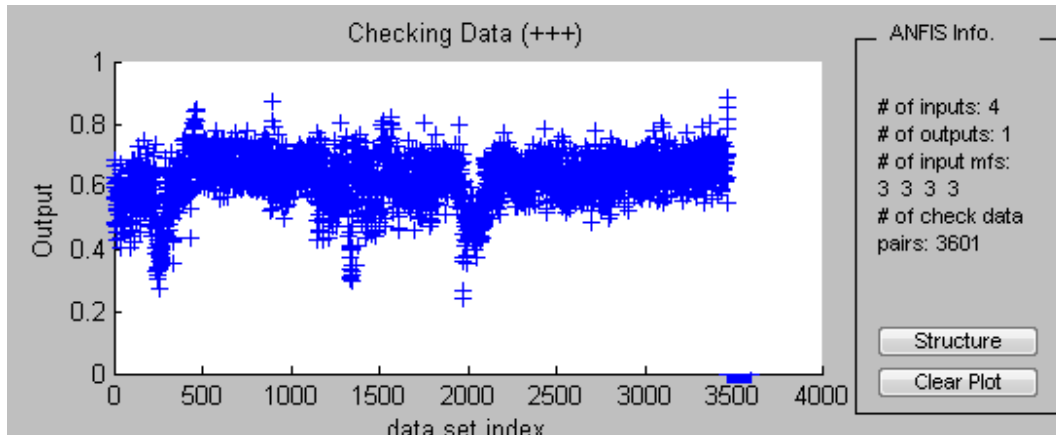


Figure III.23: La sortie des données de validation du R_x .

Le réseau du modèle (R_f)

La même chose pour la modélisation de la fonction R_f , nous avons l'ANFIS à quatre entrées X, P1, P2, T1 et une sortie F, En utilisant les mêmes étapes que précédemment pour le modèle R_x . Les sorties après apprentissage, test et validation sont représentés par les figures suivantes :

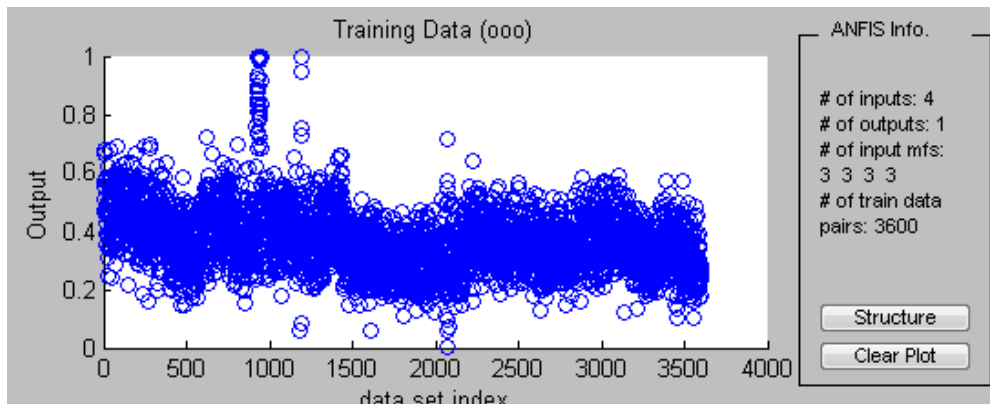


Figure III.24: la sortie d'apprentissage (R_f)

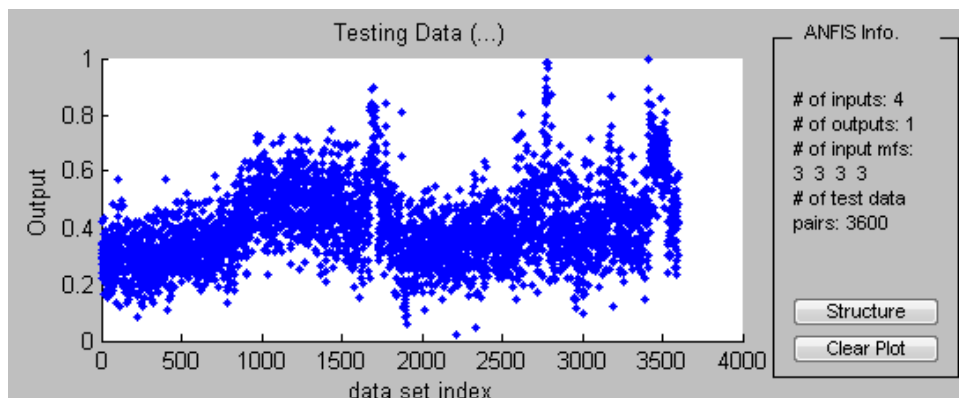


Figure III.25: La sortie de teste pour R_f .

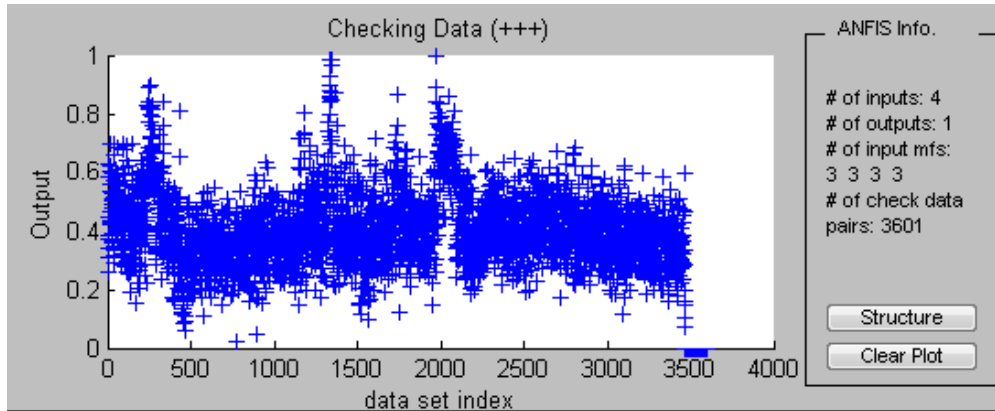


Figure III.26: La sortie de validation du R_f .

Résultat de la modélisation: Les deux tableaux III.5 et III.6 donnent les meilleures architectures de R_x et R_f expérimentées pour la modélisation du système industriel :

Tableau III.5: Les meilleurs essais pour le réseau R_x obtenu.

les réseaux (R_x)	Nombre des ensembles flous	type des fonctions d'appartenance	Itération	l'erreur d'apprentissage	l'erreur de test	l'erreur d'évaluation
essaie 1	3 3 3 3	gaussmf	100	0.039	0.086	0.134
essaie 2	3 3 3 3	pimf	30	0.043	0.403	0.397
essaie 3	3 3 3 3	tramp	100	0.042	0.146	0.143

Nous avons pris l'essaie 1 pour le modèle de R_x , avec l'erreur entre la sortie réelle et la sortie désirée représentée sur la figure suivante :

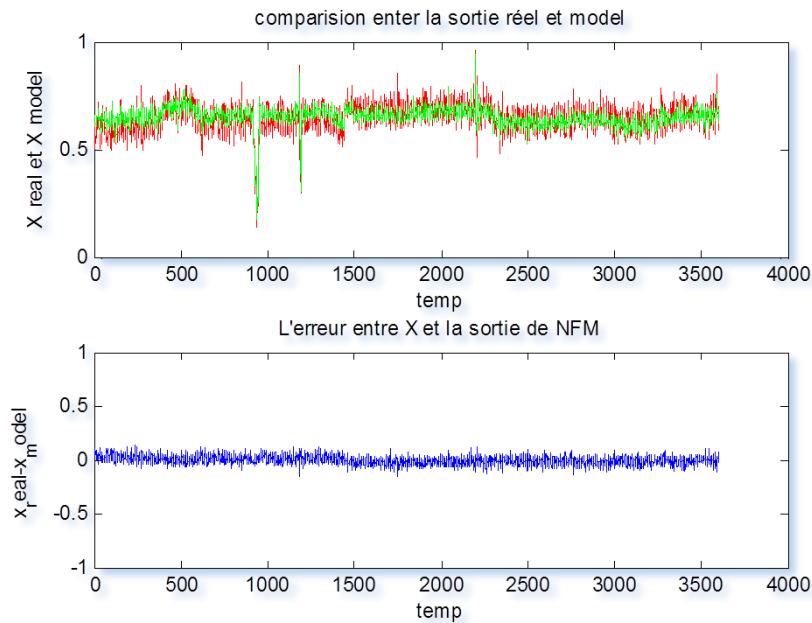
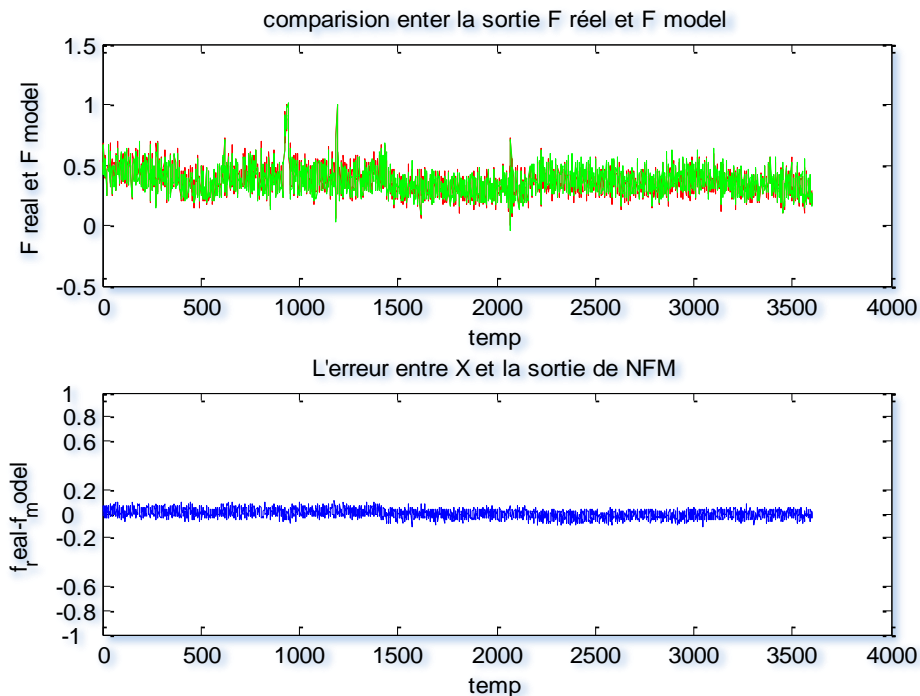


Figure III.27: La sortie réelle du système et la sortie du modèle et l'erreur entre eux pour R_x .

Tableau III.6: Les meilleurs essais pour le réseau Rf obtenu.

les réseaux (Rf)	Nombre des ensembles flous	type des fonctions d'appartenance	Itération	l'erreur d'apprentissage	l'erreur de test	l'erreur d'évaluation
essaie 1	3 3 3 3	gaussmf	100	0.034	0.125	0.140
essaie 2	3 3 3 3	pimf	50	0.084	0.400	0.381
essaie 3	3 3 3 3	tramp	100	0.031	0.121	0.113

Après plusieurs essais nous donnons les meilleurs résultats du modèle Rf avec l'erreur entre la sortie réelle et la sortie désirée représentée sur la figure suivante :

**Figure III.28:** La sortie du système et la sortie du modèle et l'erreur entre eux pour Rf .

III.4.5 Génération des résidus et détection de défaut:

III.4.6.1 Utilisation du modèle neuronal: Le modèle neuronal conçu, reproduit les comportements sains de l'actionneur DAMADICS. On peut maintenant introduire la deuxième étape qui consiste à générer les résidus pour la détection de défauts.

Dans notre cas, on a deux résidus suivant les équations ci-dessous. Afin d'illustrer la méthode nous avons choisi les quatre défauts : f_7 , f_{10} , f_{15} et f_{17} à détecter par le modèle neuronal et f_1 et f_{13} par le modèle ANFIS. Les résidus sont rF et rX . Ces résidus sont illustrés par les Figures III.30 et III.31 après filtrage passe-bas.

$$rF = F_{réel} - F_{mod} \quad (III.17)$$

$$rX = X_{réel} - X_{mod} \quad (III.18)$$

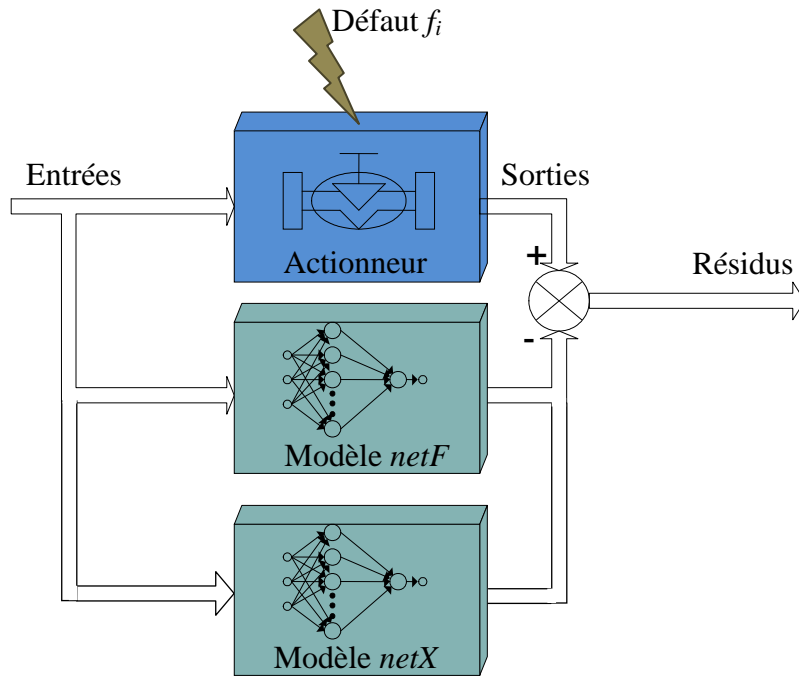


Figure III.29: Génération des résidus par les réseaux de neurones.

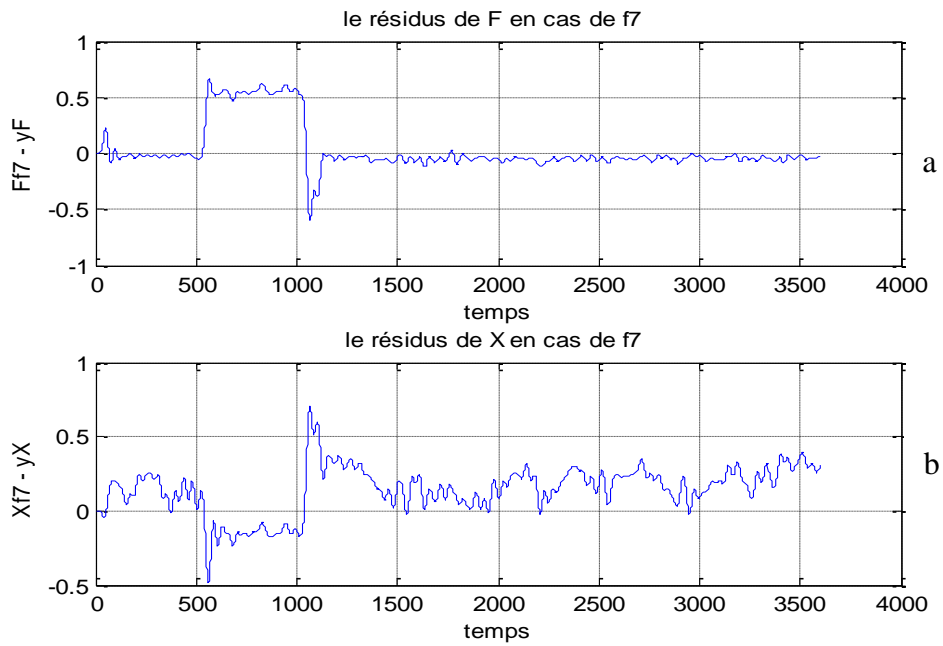


Figure III.29: (a) Résidu r_F , (b) Résidu r_X dans le cas de f_7 .

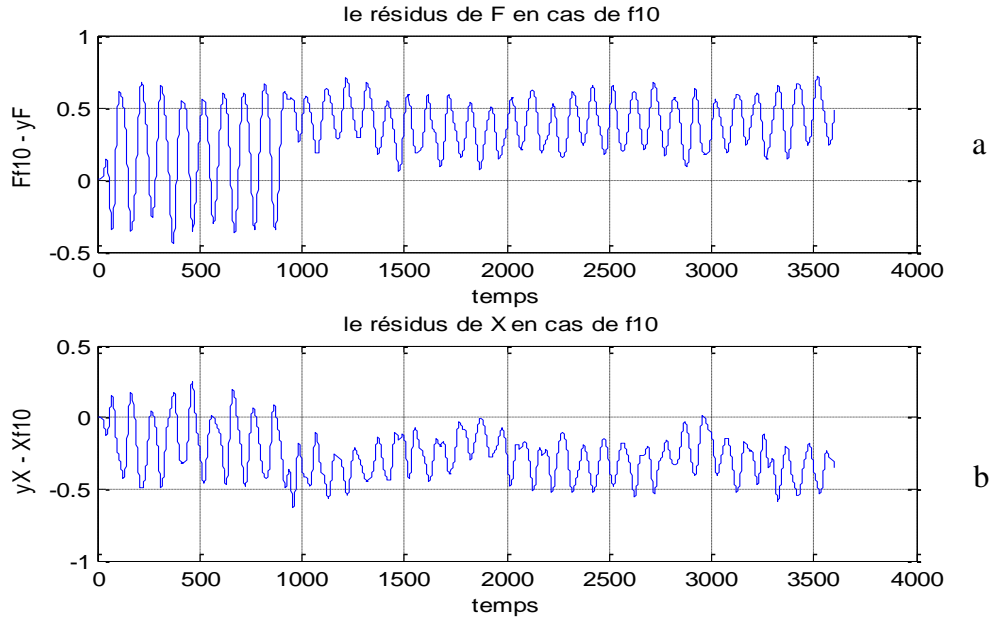


Figure III.30: (a) Résidu rF, (b) Résidu rX dans le cas de f_{10} .

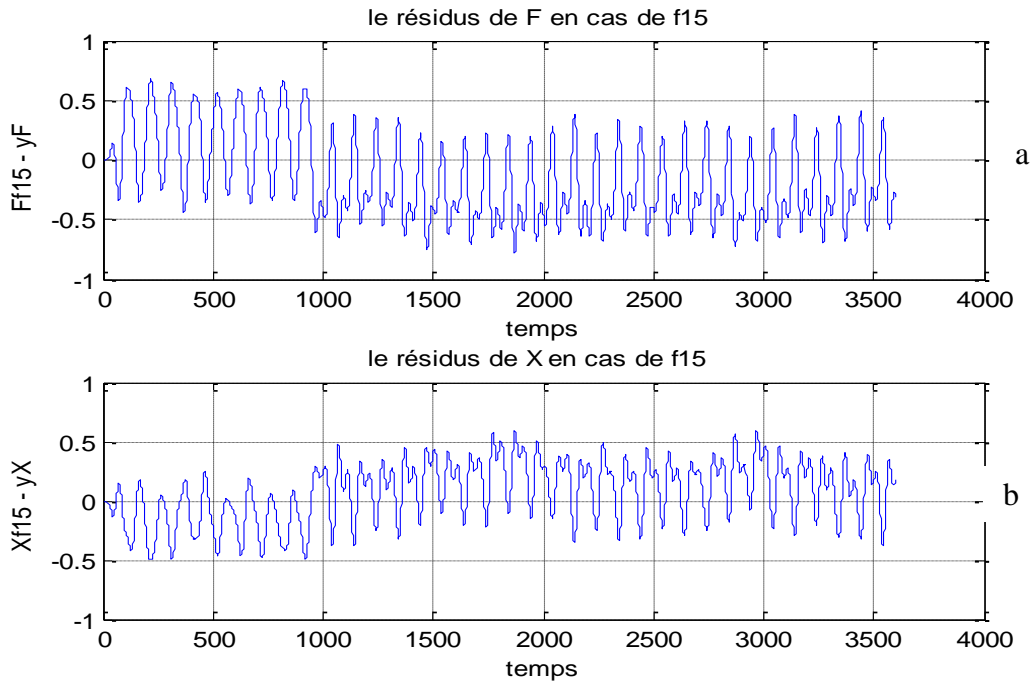


Figure III.31: (a) Résidu rF, (b) Résidu rX dans le cas de f_{15} .

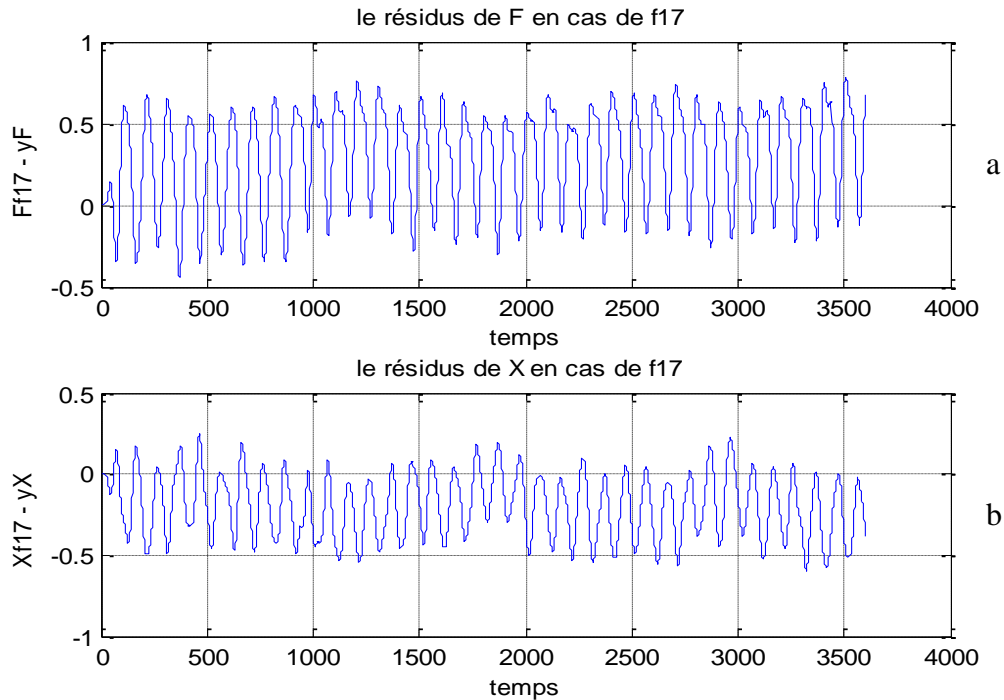


Figure III.32: (a) Résidu rF , (b) Résidu rX dans le cas de f_{17} .

En absence de défaut, nous remarquons que l'amplitude crête à crête des résidus est comprise dans l'intervalle $[-0,2 \ 0,2]$. Cette remarque est faite, dans le cas où nous voudrions appliquer un seuil de détection, ce qui n'est pas notre préoccupation essentielle dans ce travail dont le but est de montrer le faisabilité, l'utilité et l'efficacité des techniques de l'intelligence artificielle dans la détection et la reconnaissance des défauts.

Après avoir généré ces résidus, nous les avons filtré, pour simplifier et rendre plus efficace la tâche de classification.

III.4.6.2 Utilisation du modèle ANFIS: Selon les sorties avec défauts et les sorties du modèle sans défauts, nous avons défini les variables suivantes qui caractérisent les résidus:

$$rX = X_{real} - X_{mod} \quad (\text{III.19})$$

$$rF = F_{real} - F_{mod} \quad (\text{III.20})$$

L'indice « *real* » définit le signal réel, et « *mod* » définit la sortie du modèle.

rX : Résidus pour la sortie X et rF : résidus pour la sortie F, comme indiqué par la Figure III.33.

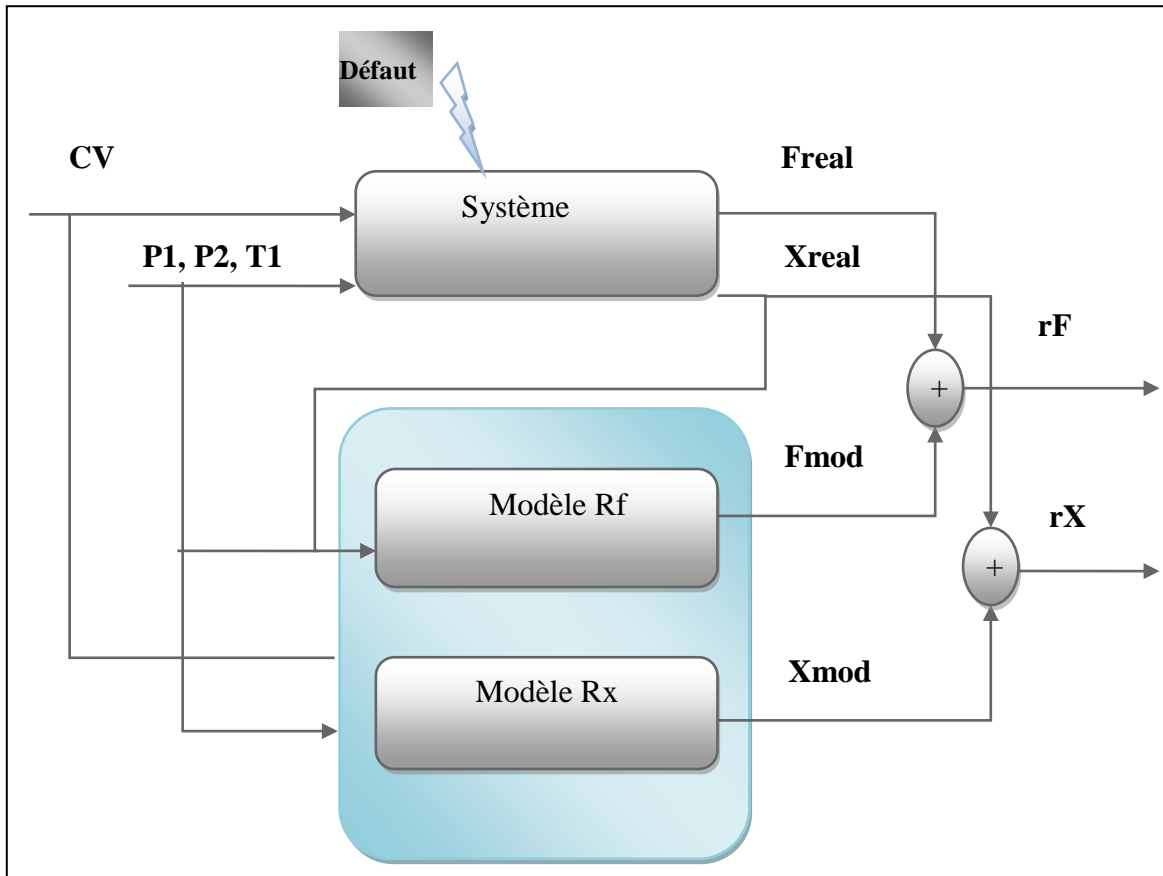


Figure III.33: La génération des résidus par un modèle ANFIS.

Les défauts f_1 et f_{13} ont des caractéristiques spéciales, nous donnons les paramètres de chaque défaut dans le tableau III.7 :

Tableau III.7 : les paramètres de défauts.

Paramètres	Fs	t_{from}	t_{to}	fd	type
Défaut f_1	1	1000	2000	1	Abrupt grand
Défaut f_{13}	1	2500	3600	1	Incipient

Les résidus rX , et rF obtenues à partir des modèles Rx et Rf sont présenté ci-dessous :

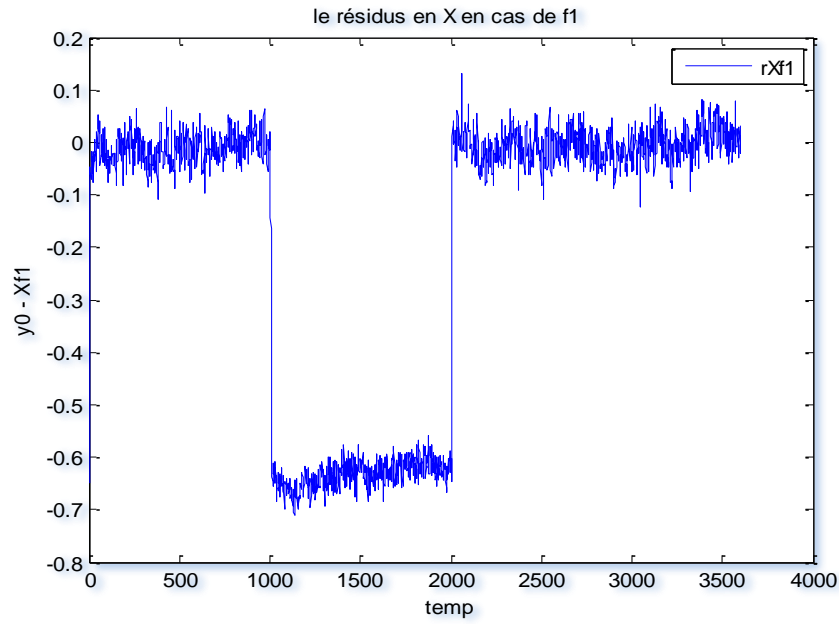


Figure III.17: Le résidu rX dans le cas ou le défaut f_1 est présent dans le système.

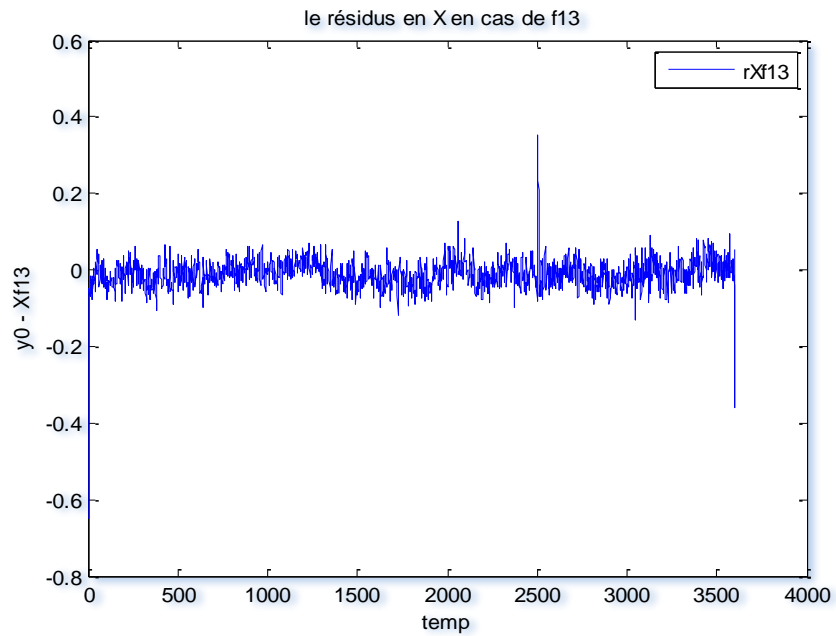


Figure III.18: Le résidu rX dans le cas de défaut f_{13} est présent dans le système.

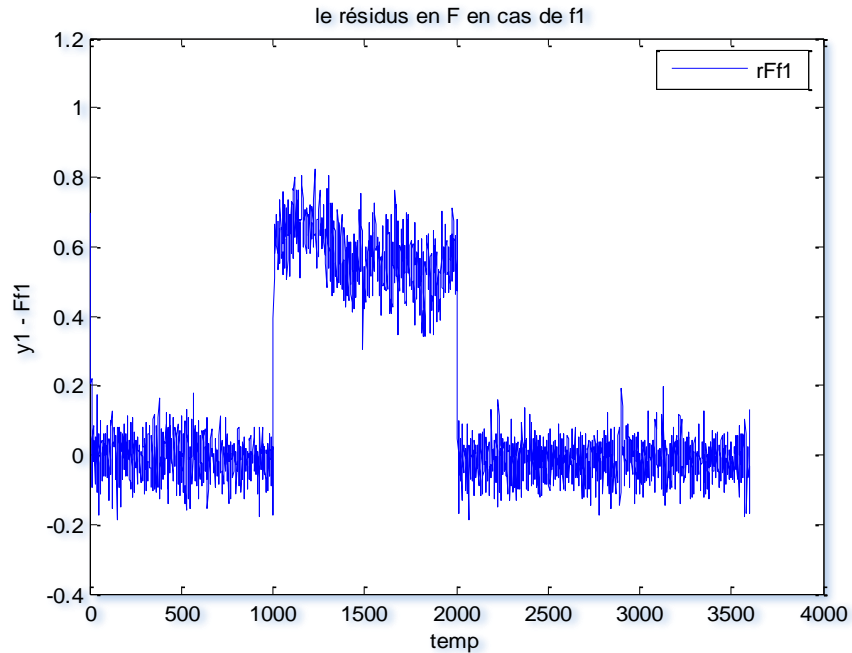


Figure III.19: Le résidu rF dans le cas du défaut f_1 présent dans le système.

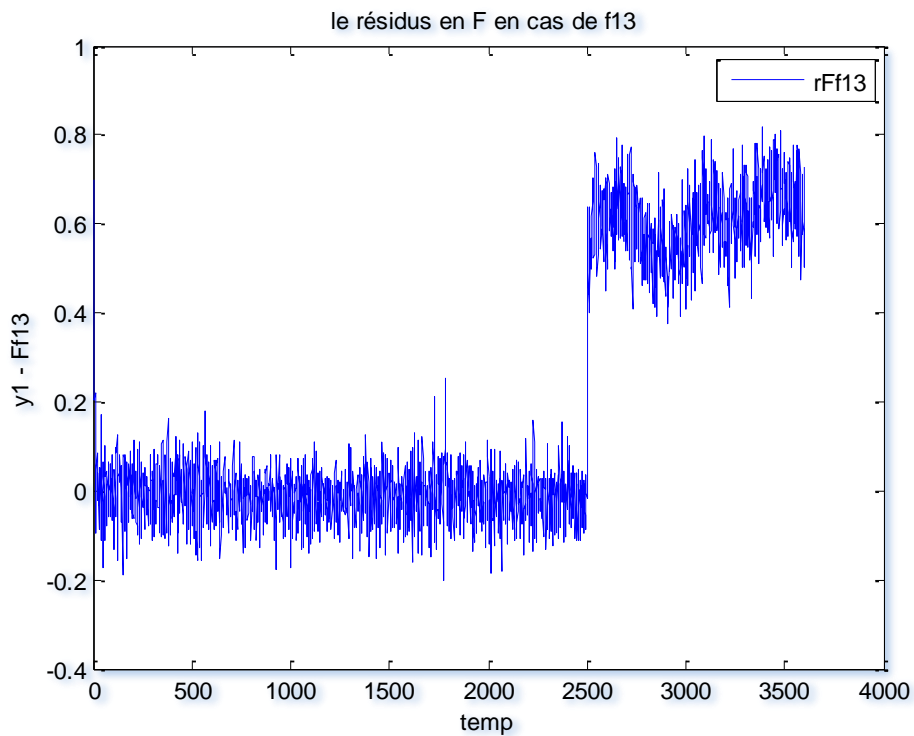


Figure III.20: Les résidus rF dans le cas du défaut f_{13} présent dans le système.

Pour mieux caractériser les résidus, nous les avons introduit dans un filtre Bass-bas de type *Butterworth*, pour éliminer les perturbations, les sorties filtrées sont présentées Figure III.21.

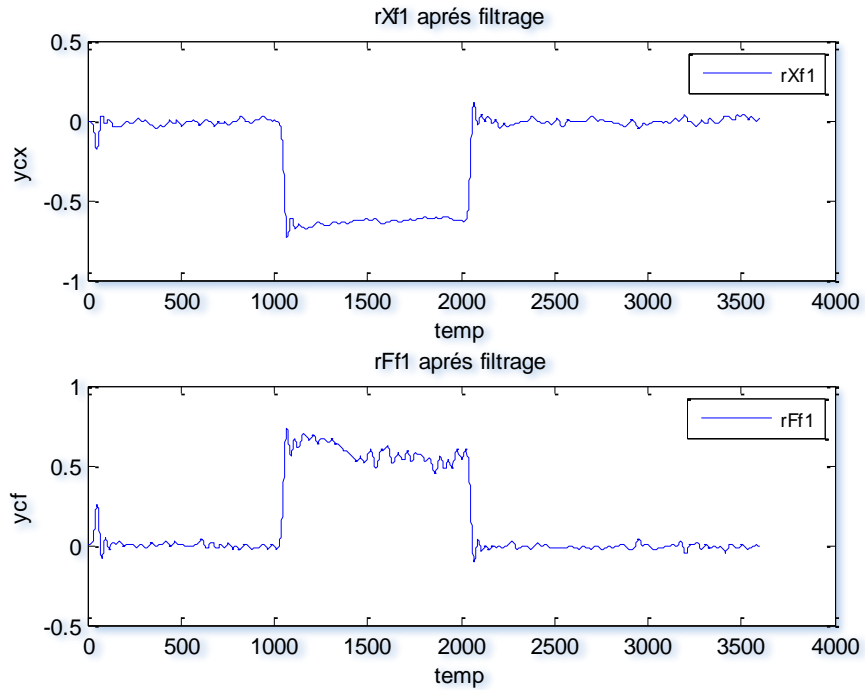


Figure III.21: Les résidus dans le cas de défaut f_1 après le filtrage.

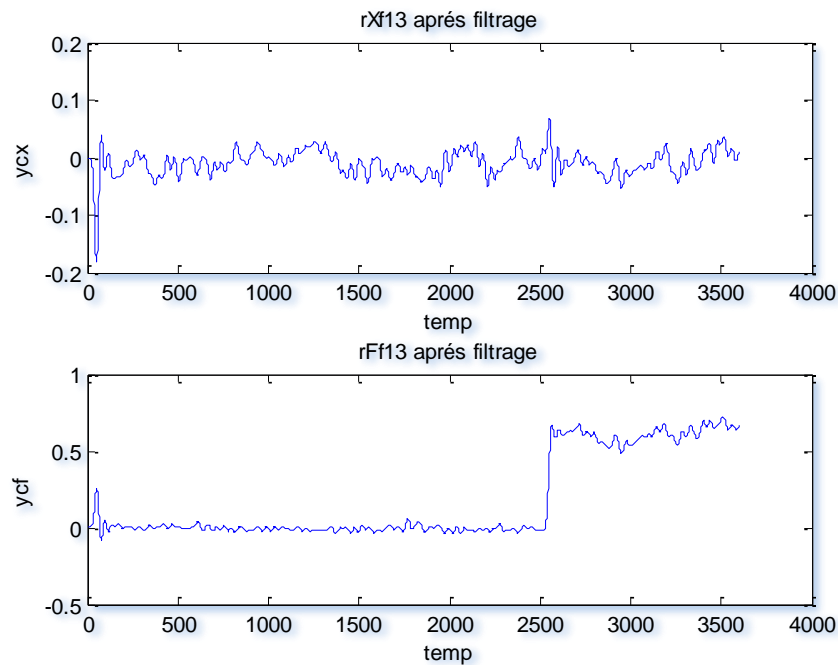


Figure III.22 : Les résidus dans le cas de défaut f_{13} après le filtrage.

III.4.6 Comparaison des résultats et discussion:

Nous avons appliqué une technique FDI pour un système industriel (Actuateur DAMADECS) par l'utilisation de modèles de comportement sains en utilisant deux approches différentes (réseaux

de neurone et ANFIS). Les figures précédentes illustrent les résultats de simulation pour la détection qui comporte les trois étapes suivantes:

- 1- La modélisation : apprentissage, test et validation du modèle,
- 2- La génération des résidus,
- 3- La détection de défauts par analyse de ces résidus.

Suivant ces étapes et leurs résultats de simulation, on peut conclure que dans la phase d'apprentissage (Figures III.12, III.13, III.21 et III.24) l'erreur d'apprentissage diminue avec l'augmentation du nombre d'itérations et du nombre d'exemples (Apprentissage, test et validation). Après obtention des modèles des comportements sains par les deux techniques, on commence la deuxième étape de génération de résidus en utilisant comme défauts introduits f_7 , f_{10} , f_{15} et f_{17} avec l'approche neuronale et les défauts f_1 et f_{13} avec l'ANFIS, afin de tester la capacité de détection de défauts de ces deux techniques. Ces défauts sont détectés à l'aide des résidus R_x et R_f . Par exemple le défaut f_1 est détecté dans la période de temps [1000s, 2000s], et le défaut f_{13} est détecté durant la période [2500s, 3600s].

Enfin d'après les résultats on constate que la détection de défauts par les réseaux de neurones ou par les réseaux de type ANFIS est possible : chaque modèle a quatre entrées et une seule sortie. Avec ces architectures on obtient deux signaux de résidus insuffisants pour localiser et isoler les 19 défauts de l'actionneur DAMADICS. Il est donc nécessaire de développer une autre technique utilisant les outils (réseaux de neurones, système ANFIS) pour détecter diagnostiquer l'ensemble des défauts d'une manière robuste et rapide. Cette contribution sera proposée dans le chapitre suivant.

III.5 Conclusion

Dans ce chapitre, deux modèles ont été présentés en détail pour la génération des résidus et la détection : un modèle neuronal et un modèle Neuro-Flou. Ces deux modèles simulent bien le comportement de l'actionneur industriel sain.

Pour avoir une bonne précision avec ces deux modèles, il faut faire des choix comme cela a été mentionné dans le chapitre II :

- Pour les réseaux de neurones : le choix de l'architecture, du nombre de couches et de neurones par couche.
- Pour l'ANFIS : le choix des fonctions d'appartenance, des univers de discours, et des règles d'inférence.

Nous avons utilisé ces deux modèles pour la génération des résidus et la détection de certains défauts choisis aléatoirement parmi les défauts de l'actionneur. Les résultats obtenus montrent clairement que ces deux techniques peuvent être employées dans la détection de défauts.

Le chapitre suivant, concerne l'étape du diagnostic. Dans cette étape nous allons utiliser les réseaux des neurones comme outil pour créer des modèles de comportements sain et défectueux de l'actionneur DAMADICS. Sur cette base, un algorithme sera développé pour la détection, et l'isolation des 19 défauts existant dans cet actionneur.

Chapitre IV

Diagnostic rapide par synthèse de modèles de dysfonctionnement

Sommaire

IV.1 Introduction.....	92
IV.2 Synthèse de modèles de dysfonctionnement.....	94
IV.3 Diagnostic par évaluation des résidus.....	95
IV.4 Diagnostic par calcul de la probabilité des modèles.....	96
IV.5 Structuration des résidus.....	98
IV.5.1 Matrice des signatures.....	98
IV.5.2 Réduction de l'ensemble des résidus.....	100
IV.6 Diagnostic des défauts de l'actionneur DAMADICS.....	101
IV.6.1 Diagnostic par analyse des modèles de référence.....	101
IV.6.2 Diagnostic par analyse des modèles de dysfonctionnement.....	110
IV.6.3 Structuration et discussion.....	134
IV.6.4 Résultat de la réduction de l'ensemble des résidus.....	135
IV.7 Conclusion.....	137

IV.1 Introduction

Effectuer le diagnostic de fonctionnement d'un système complexe consiste à détecter et localiser les défauts internes (affectant le processus lui-même) ou les défauts externes (affectant les actionneurs, les capteurs) et les fonctionnements anormaux (phase de détection), puis à estimer les caractéristiques principales des défauts mis en évidence (phase de localisation et d'identification des défauts). Dans la plupart des situations pratiques, le diagnostic de fonctionnement doit être effectué en présence de bruits de mesure, de perturbations, de bruits agissant sur le processus et d'erreurs de modélisation. Les performances de la procédure de diagnostic en termes de détection et localisation de défauts dépendent donc directement de l'exactitude de ces relations [Fra, 00].

Les relations mathématiques entre les différentes variables du système prennent généralement deux formes [Ger, 97]:

- ✓ La première est un ensemble de relations explicites de type entrée/sortie entre les variables du système. Elles sont extraites du modèle du système qui est lui-même souvent issu d'une étape antérieure d'identification dans laquelle sont pris uniquement en compte des critères de performance liés à la capacité du modèle à reproduire la sortie du système et non des critères de performance de la procédure de diagnostic.
- ✓ L'alternative consiste à ne pas formuler explicitement ces relations de redondance. Les méthodes alors utilisées sont basées uniquement sur les signaux d'entrée et de sortie du système et sont particulièrement bien adaptées à la mise en évidence de toutes les relations linéaires ou polynômiales entre les variables du système sans les formuler explicitement. Ce type de représentation permet la prise en compte de critères propres au diagnostic lors de sa synthèse [Dun, 96].

La conception de tels résidus comporte deux étapes. Tout d'abord, il convient de définir les sensibilités (et insensibilités) désirées des résidus par rapport aux différents défauts à (ou à ne pas) détecter. Puis, en fonction de ces contraintes, il faut définir le générateur de résidus adéquat.

Définition 1: La structure d'un résidu r_i par rapport à un ensemble de défauts F de dimension p est le mot binaire M_{r_i} composé de p bits ($M_{i,j}$) positionnés de la manière suivante:

- $M_{i,j} = 1$ si le résidu r_i est affecté par le $j^{\text{ème}}$ élément de $\{F\}$
- $M_{i,j} = 0$ si le résidu r_i n'est pas affecté par le $j^{\text{ème}}$ élément de $\{F\}$

Les résidus structurés (comme les trois résidus dans la Figure IV.1) sont conçus de manière à être sensibles à un sous-ensemble de défauts et robustes par rapport aux défauts restants. Lorsqu'un défaut apparaît, seul un sous-ensemble de résidus réagit.

Les informations de sensibilité et de robustesse souhaitées pour les résidus sont représentées dans une matrice binaire, appelée matrice des signatures théoriques ou encore table de codage. Celle-ci est construite de la façon suivante : lorsque le $i^{\text{ème}}$ résidu est sensible (respectivement robuste) au $j^{\text{ème}}$ défaut, alors la valeur binaire 1 (respectivement 0) est affectée à la ligne et à la colonne correspondante.

L'ensemble des résidus structurés est construit en respectant la procédure suivante :

1. Fixer la matrice des signatures théoriques que l'on souhaite obtenir.
2. Construire un vecteur de résidus ayant les propriétés désirées.
3. Si l'obtention d'un vecteur de résidus ayant les propriétés imposées par la matrice des signatures théoriques n'est pas possible, alors de nouvelles spécifications de sensibilité et robustesse doivent être proposées.

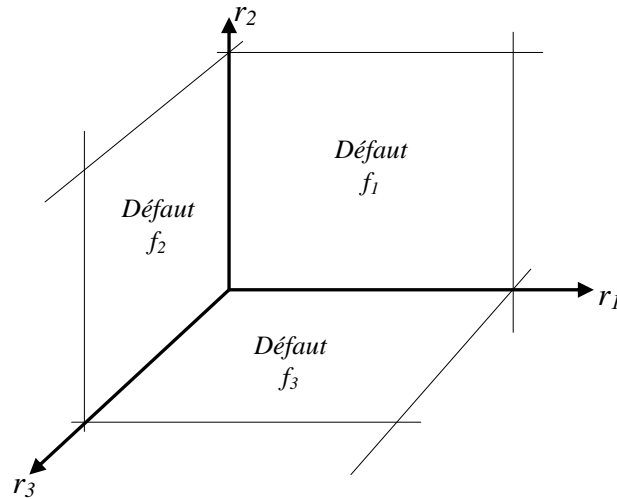


Figure IV.1: La représentation des trois résidus structurés.

Lorsque la matrice des signatures théoriques est construite, une procédure de détection, appliquée à chaque résidu, permet d'obtenir la signature réelle des résidus à un instant donné. Si cette signature est nulle, alors aucun résidu ne détecte de défaut et le système sera donc déclaré sain. Si au moins un résidu détecte le défaut, la signature deviendra non nulle. La procédure de localisation consistera alors à comparer la signature booléenne du résidu aux signatures de la matrice des signatures théoriques.

Définition 2: Un défaut est structurellement localisable si toutes les colonnes de la matrice de signatures théoriques sont différentes.

La matrice de signatures théoriques doit être proposée de telle façon que le vecteur de résidus structurés correspondant soit calculable et que les propriétés de séparabilité soient les plus intéressantes possibles. La qualité de la séparabilité de la table est donnée par l'ordre de localisation défini comme suit.

Définition 3: Un défaut est localisable d'ordre j si sa distance de *Hamming* par rapport à la signature du défaut la plus proche est de j . Nous allons illustrer ce propos à travers un exemple simple.

Exemple 1: Supposons que nous avons généré trois résidus r_1 , r_2 et r_3 à partir d'un système soumis à trois défauts f_1 , f_2 et f_3 . La Figure IV.2 représente quatre ensembles de signatures théoriques ayant des propriétés de séparabilité différentes.

(a)	f_1	f_2	f_3
r_1	1	1	0
r_2	1	1	1
r_3	0	0	1

(b)	f_1	f_2	f_3
r_1	1	1	0
r_2	1	0	1
r_3	0	0	1

(c)	f_1	f_2	f_3
r_1	1	1	0
r_2	1	0	1
r_3	0	1	1

(d)	f_1	f_2	f_3
r_1	1	0	0
r_2	0	1	0
r_3	0	0	1

Figure IV.2: Matrices de codages différents pour trois résidus et trois défauts.

➤ La matrice (a) dans la figure IV.2 illustre le cas de deux défauts non localisables (f_1 et f_2 possèdent la même signature).

- La matrice (b) dans la figure IV.2 illustre un autre cas des résidus structurés où les défauts sont localisables d'ordre 1 (un seul bit est différent entre la signature de f_1 et celle de f_2).
- Les deux matrice (c) et (d) montrent des défauts localisables d'ordre 2 (deux bits sont différents entre les signatures de deux défauts).

Malgré le critère de séparabilité identique entre les matrices (c) et (d) de la Figures IV.5, on observe que la matrice (d) sera la plus simple à traiter. D'une manière générale, lorsque deux matrices de signatures théoriques ont le même ordre de séparabilité, la matrice retenue est celle qui contient le plus de zéros.

IV.2 Synthèse de modèles de dysfonctionnement

Lorsque les défauts multiples dans un système complexe sont détectés leur localisation n'est pas triviale et le diagnostic précoce devient une tâche difficile. On peut multiplier les mesures et utiliser certains outils d'analyse (analyse des résidus, par exemple) afin de localiser ces défauts. Mais l'utilisation d'une telle approche comme nous avons vue dans le premier chapitre (par exemple, redondances de capteurs) est limitée. On considère une autre approche qui utilise l'historique des données collectées par exemple pour améliorer les connaissances sur les comportements défectueux et ensuite utiliser cette connaissance pour créer des modèles de défauts et des résidus supplémentaires. Nous avons engendré ces modèles pour estimer le défaut de chaque cas et les comparer avec des mesures pour fournir le défaut le plus probable en fonction de la distance entre les signaux et la mesure euclidienne estimée [Bla, 03], [Lef, 10]. Les modèles $FM(j)$, avec $j=1\dots p$ sont créés à base de réseaux neuronaux artificiels (Figure IV.3). Ces modèles sont formés conformément aux procédures similaires à celles utilisées pour la conception des modèles sans défauts. Chaque modèle de défaut est formé avec des données qui proviennent de l'observation du comportement défectueux correspondant à un seul défaut f_j . Ces données peuvent être fondées sur les données historiques collectées du système étudié ou avec une étude précise où les défauts sont imposés pour étudier les effets sur la sûreté du système et sa disponibilité. Une fois le modèle de défaut est formé et validé, il produit des estimations $y_k(t, f_j, \tau)$ à partir des signaux $y_k(t)$, avec $k=1\dots n$. Selon l'hypothèse que le défaut f_j se produit dans un délais τ . La comparaison de ces estimations avec les mesures réelles $y_k(t)$ sert à localiser le défaut f_j .

Les entrées du modèle $FM(j)$ sont les signaux d'entrée $u_i(t)$ du système, le défaut à diagnostiquer f_j et le délai τ estimé en présence du défaut fournie par l'étape de détection.

Les sorties du modèle défectueux $FM(j)$ sont les sorties défectueuses $y_k(t, f_j, \tau)$ estimées, avec $k=1\dots n$, $j=1\dots p$, obtenues en supposant que le défaut f_j perturbe le système à l'instant τ . La conception des modèles et leurs apprentissages sont obtenus avec la même technique que celle utilisée pour le modèle sans défaut.

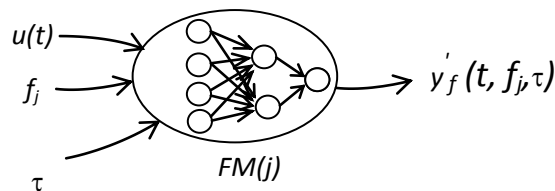


Figure IV.3: Construction d'un modèle défectueux.

Notre approche consiste à utiliser l'historique de données recueillies pour améliorer les connaissances sur les comportements défectueux, puis à utiliser cette connaissance pour concevoir des modèles de comportements défectueux et ajouter des résidus supplémentaires. Ces modèles seront utilisés pour fournir des estimations pour chaque comportement du défaut candidat et ensuite les résultats sont fournis par comparaison des estimations de décision avec les mesures collectées du système. La conception des modèles de comportements défectueux est similaire à la méthode décrite dans la partie du chapitre III.2. L'apprentissage de comportements défectueux est obtenu selon aussi par l'algorithme de Levenberg-Marquardt avec arrêt rapide. Chaque modèle est construit pour un défaut candidat f_i spécifique qui est considéré comme une entrée supplémentaire durant l'apprentissage. La Figure IV.4 présente le schéma général utilisé pour concevoir un modèle de comportements défectueux. Le vecteur $Y'_i(t)=(y'_{ki}(t))$, $k=1\dots n$, $i=1\dots p$ est déterminé avec n le nombre de sorties du modèle *NNFM* et p le nombre du modèles de défauts conçus pour le défaut f_i .

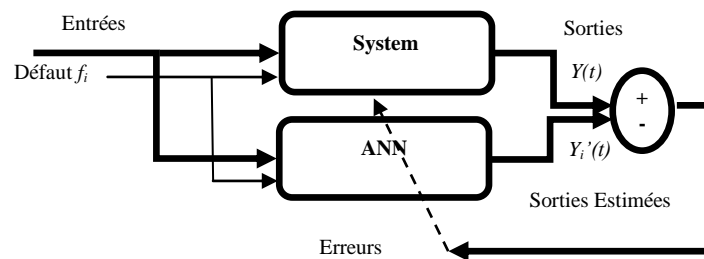


Figure IV.4: Conception de modèle neuronal des comportements défectueux.

IV.3 Diagnostic par évaluation des résidus (seuils)

Soient $d(t)$ un signal à détecter et $b(t)$ un bruit aléatoire. Deux cas peuvent se produire concernant le signal composite $m(t)$ observé [Wil, 99]:

$$\begin{aligned} H_0 : m(t) &= b(t) \\ H_1 : m(t) &= d(t) + b(t) \end{aligned} \quad (IV.1)$$

Les hypothèses H_0 et H_1 sont appelées hypothèses simples quand la distribution statistique de m sous H_0 et H_1 dépend de paramètres connus tels que l'amplitude ou la phase du signal ou la puissance du bruit. Quand la distribution de $m(t)$ et/ou de $b(t)$ sous ces hypothèses dépend de paramètres inconnus, les hypothèses sont appelées *hypothèses composites*.

Il est souvent pratique de formuler le problème de décision en considérant une statistique scalaire ou vectorielle $s(t)$ obtenue à partir des observations $m(t)$, au lieu de les traiter directement. Afin de décider entre les deux hypothèses, cette statistique $s(t)$ est comparée à un seuil. Si le seuil est dépassé pendant un intervalle de temps $[t_0 \ t_0 + \Delta t]$, on décide que le signal $d(t)$ est présent.

Dans la pratique, on doit régler le seuil pour assurer un faible nombre d'erreurs de décision. Il y a deux types d'erreurs possibles : l'erreur de non détection (décider H_0 sous H_1) et l'erreur de fausse alarme (décider H_1 sous H_0). Il y a toujours un compromis à établir entre le choix d'un seuil fort pour diminuer le nombre de fausses alarmes et celui d'un seuil faible pour diminuer le nombre d'erreurs de non détection. Pour quantifier ce compromis, il est nécessaire de spécifier les distributions statistiques de $m(t)$ sous chaque hypothèse H_0 et H_1 .

- L'utilisation de la courbe *COR* pour un seuil optimal:

On note S_i le seuil de détection et on définit la région de décision pour H_1 :

$$R_{H1} = \{s : s(t) > S_i, t \in [t_0, t_0 + \Delta t]\} \quad (IV.2)$$

Cette région est appelée région critique et elle spécifie simplement la condition sur $s(t)$ pour laquelle le détecteur déclare que le signal $d(t)$ est présent. Le détecteur détermine donc des régions de décisions binaires exclusives et la région critique spécifie complètement l'opération de détection. Les probabilités de fausse alarme P_f et de non détection P_n :

$$P_f = P(H_1/H_0) = 1 - P(H_0/H_0)$$

$$P_n = P(H_0/H_1) = 1 - P(H_1/H_1) \quad (IV.3)$$

Où $P(A=H)$ la probabilité d'un événement A sous l'hypothèse H, sont des fonctions de seuil S_i .

La probabilité de bonne détection $P_b = P(H_1/H_1)$ est communément appelée la *puissance* du détecteur et P_f est appelée le *niveau* du détecteur.

Le tracé de la paire $(P_f(S_i), P_b(S_i))$, pour $-\infty < S_i < +\infty$, produit une courbe appelée *Caractéristique Opérationnelle de Réception* (COR) [Wil, 99], [Len, 02].

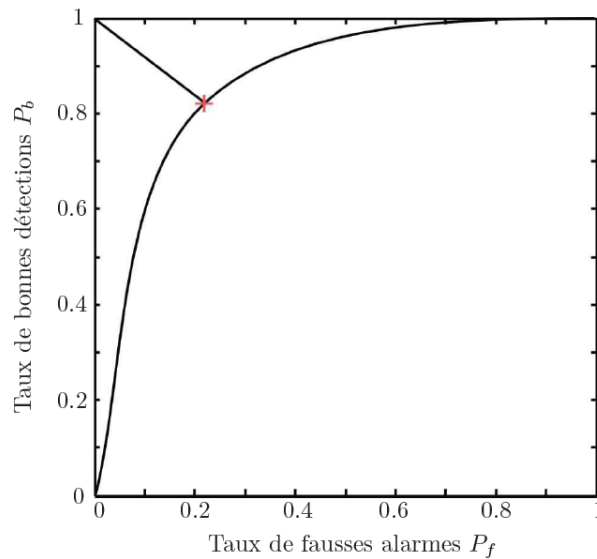


Figure IV.5: Exemple d'une courbe COR (*Caractéristique Opérationnelle de Réception*).

Une courbe COR (Figure IV.5) représente l'évolution de la puissance du détecteur P_b en fonction de son niveau P_f . En effet, ces deux probabilités sont conjointement déterminées par le choix du seuil S_i . Dans le cas où il n'y a pas de contrainte sur les probabilités d'erreurs P_f et P_n , le point optimal est le point le plus proche de la position $P_f=0, P_b=1$, qui correspond à un taux de bonnes détections égal à 1 et un taux de fausses alarmes nul, les valeurs des paramètres de réglage retenues sont celles correspondant à ce point.

IV.4 Diagnostic par calcul de la probabilité des modèles (distance)

D'habitude les résultats de diagnostic sont obtenus par la technique de seuillage comme nous avons vu précédemment. Nous avons introduit une autre technique pour un diagnostic rapide ou en ligne par la détermination des probabilités de défaut et le calcul du facteur de confiance [Kou, 11]. Dans la seconde méthode, les modèles défectueux sont exécutés simultanément à l'instant de détection du défaut $t = \tau$. Chaque modèle comportera un seul candidat du défaut et les comportements résultants seront comparés avec les données collectées pour fournir un diagnostic rapide. En cas de plusieurs défauts candidats $f_i, i=1 \dots p$, la sortie $Y'_i(t) = (y'_k(t), f_i, \tau)$ du modèle

$NNFM(i)$ ($i^{ème}$ Neural Network Faulty Model) est comparée avec le vecteur de mesure du système $Y(t)$ pour calculer les résidus additif $R_i(t)=(r_{ki}(t, \tau))$, $k=1 \dots n$. Le défaut candidat le plus probable est déterminé par comparaison de tous les résidus $r_{ki}(t, \tau)$, avec $k=1 \dots n$, $i=1 \dots p$ respectivement n le nombre de sorties et p le nombre du modèles de défauts:

$$r_{ki}(t, \tau) = y_k(t) - y'_{ki}(t, \tau) \quad (IV.4)$$

La méthode proposée utilise une fenêtre temporelle de dimension ajustable. Le diagnostic avec une grande fenêtre temporelle introduira un retard de détection, mais conduira à une décision avec un facteur de confiance élevé. Contrairement, le diagnostic avec une petite fenêtre temporelle conduira à un diagnostic rapide, mais avec un faible facteur de confiance. Pour évaluer la probabilité de défaut de chaque candidat nous définissons un paramètre $\rho_{ki}(t, T, \tau)$ donné par l'équation IV.5 traduisant les résidus cumulés sur l'intervalle de temps $[\max(0, t-T), t]$ de taille maximale T (T correspond à la taille de fenêtre temporelle).

$$\rho_{ki}(t, T, \tau) = \sqrt{\int_{\max(0, t-T)}^t (r_{ki}(u, \tau))^2 \cdot du} \quad (IV.5)$$

Ensuite, $D_j(t, T, \tau)$ est la norme euclidienne du vecteur $\Omega_i(t, T, \tau) = (\rho_{ki}(t, T, \tau))$ de dimension n :

$$D_i(t, T, \tau) = \sqrt{\sum_{k=1}^{k=n} (\rho_{ki}(t, T, \tau))^2} \quad (IV.6)$$

$D_j(t, T, \tau)$ est utilisé pour décider quel est le défaut le plus probable durant le diagnostic hors ligne ou en ligne.

Le diagnostic hors ligne à un instant t est défini par l'analyse des résultats $D_j(t, t, \tau)$ a posteriori élaboré dans l'intervalle de temps $[0, T]$ (avec $T = t$). Le diagnostic du défaut le plus probable est donné par l'équation suivante:

$$j^*(0, t) = \operatorname{argmin}_j \{D_j(t, t, \tau), j = 1, \dots, p\} \quad (IV.7)$$

On calcule la probabilité du défaut P_i par l'équation IV.8 pour confirmer que le défaut dans le système est de type f_j .

$$P_i(t, t, \tau) = \frac{1}{D_i(t, t, \tau) \sum_{k=1}^{k=p} \left(\frac{1}{D_k(t, t, \tau)} \right)} = \left[\sum_{k=1, k \neq i}^{k=p} \left(\frac{D_i(t, t, \tau)}{D_k(t, t, \tau)} \right) + 1 \right]^{-1} \quad (IV.8)$$

Le diagnostic rapide en ligne du défaut le plus probable est donnée par les valeurs de $D_j(t, 0, \tau)$ à l'instant $T=0$ et par l'équation (IV.9) :

$$j^*(t, t) = \operatorname{argmin}_j \{D_j(t, 0, \tau), j = 1, \dots, p\} \quad (IV.9)$$

Afin d'atténuer les effets du bruit et les valeurs indésirables, la détermination du défaut le plus probable (l'équation IV.10) se fait par comparaison des résidus cumulés de chaque intervalle de temps $[\max(0, t-T), t]$ de taille maximale T .

$$j^*(t - T, t) = \operatorname{argmin}_j \{D_j(t, T, \tau), j = 1, \dots, p\} \quad (IV.10)$$

La probabilité P_i est obtenue par l'équation IV.11 :

$$P_i(t, T, \tau) = \left[\sum_{k=1, k \neq i}^{k=p} \left(\frac{D_i(t, T, \tau)}{D_k(t, T, \tau)} \right) + 1 \right]^{-1} \quad (IV.11)$$

La taille T de la fenêtre est définie afin de satisfaire les conditions de temps réel pour le diagnostic rapide ou en ligne.

Nous avons introduit un autre indicateur (le facteur de confiance CF_j (équation IV.12)). Les résultats obtenus par cet indicateur CF_j confirme que le défaut détecté est un défaut de type i :

$$CF(t, T, \tau) = \frac{\max(P_i(t, T, \tau) : i=1, \dots, p) - \max(P_k(t, T, \tau) : k=1, \dots, p, k \neq i)}{\max(P_i(t, T, \tau) : i=1, \dots, p)} \quad (\text{IV.12})$$

CF_j est proche de 1 lorsque D_j est plus proche de 0 et au contraire loin de la valeur 1 lorsque D_j est plus loin de 0.

Notons que la méthode précédente peut aussi être combinée avec la technique de seuillage afin d'éviter la multiplication des résidus et de fournir une décision fiable selon un schéma hiérarchique. Dans un premier temps, un petit nombre de résidus sont évalués et analysés. Cette étape conduit à la détermination d'un sous-groupe de défauts possibles qui ont la même signature. Ensuite, les probabilités de défaut sont élaborées au sein de ce sous-groupe afin de sélectionner le candidat le plus probable de défaut.

IV.5 Structuration des résidus

Durant la procédure de diagnostic, une étape d'évaluation des résidus (localisation des défauts) nécessite néanmoins une structuration. De façon idéale, on cherche à ce qu'un résidu ne soit sensible (ou insensible) qu'à un seul défaut. Cette structuration, qui correspond à un découplage, peut être effectuée de différentes façons. Il y a plusieurs manières de structurer les résidus, suivant notre étude nous avons abordé les trois différentes techniques citées ci-dessous.

IV.5.1 Matrice des signatures:

La structuration des résidus (matrice des signatures) est la base pour la technique localisation des défauts. Cette matrice des signatures théoriques traduit la sensibilité des résidus par rapport aux différents défauts possibles. Elle peut-être dérivée à partir des équations qui décrivent les résidus. Généralement, il est souhaitable que les résidus soient fortement localisables, ce qui permet aux petits défauts d'être correctement localisés dans le cas où certains résidus ne sont pas assez grands pour atteindre le seuil de détection. Deux exemples de structuration des résidus fortement localisables sont présentés dans la Figure IV.6.

	f_1	f_2	f_3
r_1	1	0	0
r_2	0	1	0
r_3	0	0	1

(a)

	f_1	f_2	f_3
r_1	0	1	1
r_2	1	0	1
r_3	1	1	0

(b)

Figure IV.6: Deux structurations des résidus fortement localisables.

La structuration (a) est plus simple que (b) pour la localisation des défauts multiples. Toutefois, elle est moins robuste contre les perturbations dans le cas sans défaut. Supposons que les perturbations sont présentes, on peut obtenir une fausse alarme dans la structuration (a) il suffit qu'un seul résidu à une variation de zéro à une valeur non nulle. Dans la structuration (b) il est nécessaire que deux résidus changent de zéro à non zéro. De cette façon, la structuration des résidus (b) offre une robustesse contre les fausses alarmes. Il est donc important de savoir quel ensemble de résidus constitue la structuration optimale. Par exemple, est-ce que un avantage

d'inclure un résidu supplémentaire? Pour l'instant, on considère un seul défaut dans la structuration des résidus (b) de la figure IV.6. Evidemment au moins un des deux résidus r_2 et r_3 doivent être inclus. De nombreux résidus accroissent la robustesse contre les fausses alarmes et les non détections de défauts. Toutefois, sans plus de connaissances, il est difficile de déterminer la structuration optimale des résidus. Pour répondre à cette question, il faut d'abord préciser un critère de performance.

Après avoir détecté un défaut, l'étape de localisation est un deuxième niveau de diagnostic, qui consiste à déterminer de manière plus approfondie les défauts expliquant les observations recueillies. En somme, il s'agit de répondre à la question : quels défauts se sont produits ?

Répondre à cette question suppose que l'on dispose d'un minimum d'information permettant de discriminer l'influence de tel ou tel défaut. Il faut donc disposer de modèles de mauvais fonctionnement qui peuvent être regroupés, par exemple, au sein d'une table de signature des défauts. Une matrice de signatures est généralement booléenne, traduisant l'influence des différents défauts que l'on cherche à détecter et à localiser sur les résidus. En supposant que p défauts doivent être localisés à partir de n résidus, une table de signature booléenne est une matrice T de n lignes et de p colonnes telle que $T_{ij}=1$ si le résidu r_i est affecté par le défaut f_j et $T_{ij}=0$ sinon [Adr, 00]. On peut donner comme exemple 2 la Figure IV.7 avec $n=5$ résidus ; $p=5$ défauts:

	f_1	f_2	f_3	f_4	f_5
r_1	1	0	0	1	0
r_2	0	1	1	1	0
r_3	1	0	1	1	1
r_4	0	1	0	0	1
r_5	0	0	1	1	0

Figure IV.7: Exemple de Matrice de signature.

Afin de pouvoir localiser au mieux les défauts, une certaine structuration de la table de signature est nécessaire. Pour cela, on essaye de générer des résidus sensibles à différents groupes de défaut. Ainsi, une table de signature diagonale est idéale car elle permet de traiter simplement le cas de défauts multiples; elle est cependant généralement très difficile de voire impossible à obtenir en pratique du fait des nombreux couplages au sein du système étudié et d'une instrumentation souvent limitée. Du point de vue de la génération des résidus, la structuration d'une ligne de la table de signature peut se faire en synthétisant un résidu r_i détectant les défauts f_j tels que $T_{ij}=1$ et découple des défauts f_j tels que $S_{ij}=0$ et des perturbations. On se ramène de la sorte à plusieurs problèmes de génération de résidus pour la détection avec différents objectifs de découplage. Les informations contenues dans la table de signature sont ensuite comparées aux valeurs de tests (c-à-d résultats de la comparaison des résidus à des seuils, par exemple) afin de calculer à l'aide d'opérations logiques la décision. Cette dernière liste les défauts et/ou les combinaisons de défauts cohérentes avec les observations recueillies. La comparaison de différentes implantations de la décision à partir d'une table de signature ainsi que la formulation des hypothèses sous-jacentes sont étudiées par Cordier, et al. [Cor, 00].

Du fait des inévitables erreurs de modélisation, aucun découplage ne peut être considéré comme parfait. De plus, le dépassement du seuil en réponse à un défaut suppose que ce défaut ait une amplitude suffisante. Ainsi, la notion de "0" et de "1" dans la table de signatures est fortement liée au choix des seuils lors de l'évaluation des résidus. La notion de sensibilité et plus précisément de sensibilité relative aux différents défauts et perturbations joue donc un rôle très

important dans l'articulation entre la génération des résidus et la décision [Com, 03]. Cela a pu amener à des travaux sur la génération de résidus directionnels. Si les capacités de localisation sont théoriquement intéressantes a priori dans le cas de défauts simples, l'extension au cas de défauts multiples est quant à elle nettement plus délicate avec ce genre d'approches.

Lorsqu'un défaut est détecté, il est souvent nécessaire de préciser quel défaut explique les observations recueillies. Cela correspond à la localisation. Dans le but de structurer une table de signature en vue de localiser les défauts.

IV.5.2 Réduction de l'ensemble des résidus:

Le but de l'opérateur de réduction est de limiter la complexité de la matrice des signatures et de donner un nombre suffisant de résidus pour isoler les défauts. L'objectif est de conserver la meilleure caractérisation des relations de dépendance possible entre les signatures. L'opérateur de réduction est ainsi amené à jouer un rôle important dans la maîtrise du compromis entre la complexité des résidus et la qualité des décisions. Dans cette partie, on essaie de développer une technique pour réduire le nombre des résidus obtenus pour faciliter l'étape dévaluation et de diminuer le temps de calcul. Par exemple, on peut représenter ces résidus dans un plan composé de deux axes et on observe les concentrations de ces résidus dans ce plan. On fait la projection de ces résidus sur l'un de ces axes de ce plan. On peut donner un exemple pour bien expliquer la technique:

On a p résidus obtenus par les modèles $NNFM(k)$ avec $k=1..p$. La représentation de ces résidus dans le plan de deux dimension est la figure IV.8.

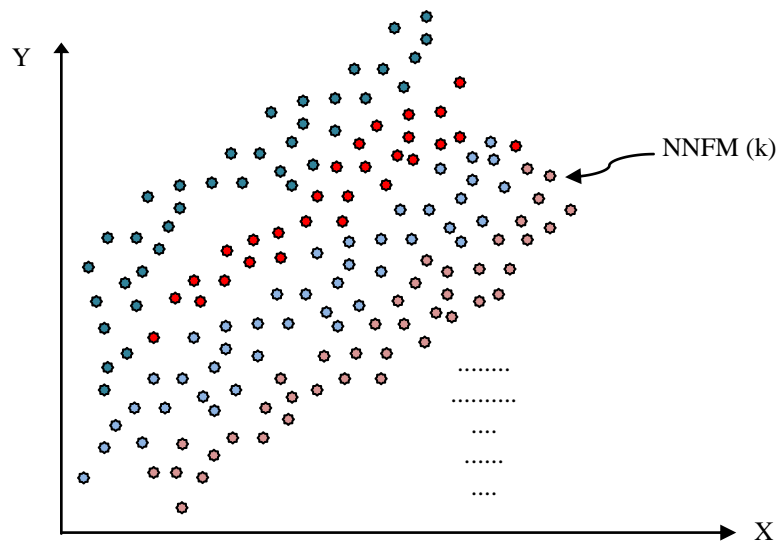


Figure IV.8: Exemple des résidus dans plan (X,Y).

Chaque point présente un résidu d'un modèle $NNFM$ à un instant donné. On peut projeter ces résidus et tracer les histogrammes sur les deux axes. Les valeurs moyennes et les écarts types sont donnés suivants les équations suivantes:

$$\sigma_X^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{m}_x)^2 \quad (\text{IV.13})$$

$$\sigma_Y^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{m}_y)^2 \quad (\text{IV.14})$$

On peut garder l'un de ces résidus selon les valeurs de variances (σ_X^2 , σ_Y^2) citées dans le tableau suivant:

Tableau IV.1: Comparaison des variances des résidus sur les axes.

Comparaison de la variance	Observation
$\sigma_X^2 \gg \sigma_Y^2$	Garder r_X les résidus projetés sur l'axe X
$\sigma_X^2 \cong \sigma_Y^2$	Pivoter l'un des deux axes par un angle α
$\sigma_X^2 \ll \sigma_Y^2$	Garder r_Y les résidus projetés sur l'axe Y

IV.6 Diagnostic des défauts de l'actionneur DAMADICS

Dans la littérature, plusieurs algorithmes d'FDI ont été testés et validés sur l'actionneur benchmark DAMADICS. Parmi ces techniques, on peut citer les travaux des [Pre, 06], DAMADICS, 2002 [8]. Avec plusieurs méthodes FDI, ce système a été expérimenté par des chercheurs Patan et al., [Pat, 05], Koscielnny et al., [Kos, 06] et Lefebvre et al., [Lef, 10]. Dans Patan et al., [Pat, 05], l'évaluation binaire des symptômes de défaut est explorée et les auteurs se sont focalisés sur l'optimisation de l'architecture de réseau de neurones en fonction du critère d'information Akaide (AIC Akaide Information Criteria) et d'erreur de prédiction finale (FPE Final Prediction Error). Ces deux critères comprennent l'erreur d'apprentissage et aussi un terme qui dépend de la complexité (taille du réseau en nombre de neurone) et sur la dimension de l'ensemble d'apprentissage, afin d'optimiser le taux de complexité/performances. Ces techniques fournissent des performances intéressantes avec des petits réseaux pour la détection de défauts, mais certains ne sont pas localisables. En comparaison, notre approche exigera un plus grand nombre de réseaux de neurones, mais tous les défauts seront détectés et isolés. En ce qui concerne le travail de Koscielnny et al., [Kos, 06], une évaluation des symptômes de défaut à valeurs multiples est introduite pour améliorer la localisation des défauts. Une telle méthode nécessite une connaissance heuristique sur l'influence des défauts sur les résidus. En comparaison avec notre technique, nous avons utilisé une triple évaluation des résidus pour le cas des comportements sans défaut et des doubles évaluations des résidus pour les comportements défectueux.

IV.6.1 Diagnostic par analyse des modèles de référence:

Plusieurs types de modèles existent dans la littérature : modèles de défaut, modèles heuristiques ou analytiques (*cf.* chapitre I). Le modèle retenu est un modèle analytique décrivant le comportement du système. La littérature traite abondamment ce sujet et le lecteur pourra se référer aux travaux suivants : [Bid, 91a], [Bid, 91b], [Bid, 93], [Con, 94], [Tho, 91] pour de plus amples précisions.

Les techniques dérivées par l'utilisation d'un modèle de référence sont variées mais pas toujours efficaces. En effet, les procédés physiques sont bien souvent des systèmes dynamiques complexes, présentant de fortes non-linéarités dans leur fonctionnement. Des simplifications s'imposent pour formuler un modèle exploitable. Ainsi, les techniques de linéarisation autour d'un point de fonctionnement ou la réduction d'ordre sont autant de méthodes dégradant les performances du modèle mathématique. De même, d'autres problèmes subsistent avec certains paramètres du modèle non mesurables ou variant dans le temps. Le concept général de la génération de résidus reste le même que pour les modèles analytiques. Il consiste à comparer les

sorties du procédé à leurs estimées, mais dans ce cas, les estimées sont calculées par un modèle neuronal.

IV.6.1.1 Diagnostic par les réseaux de neurone:

Le classifieur neuronal que nous avons conçu (Figure IV.9) se compose aussi d'un réseau de type MLP à deux entrées et quatre sorties. L'ensemble d'apprentissage de ces réseaux se compose des résidus générés dans la deuxième étape, comme entrées et les fautes f_7 , f_{10} , f_{15} et f_{17} comme sorties.

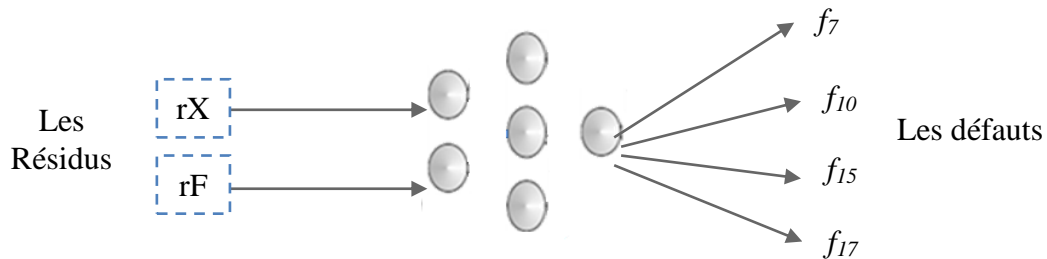


Figure IV.9: Classifieur neuronal utilisé.

Dans ce cas l'organigramme de la procédure est présenté dans la Figure IV.10.

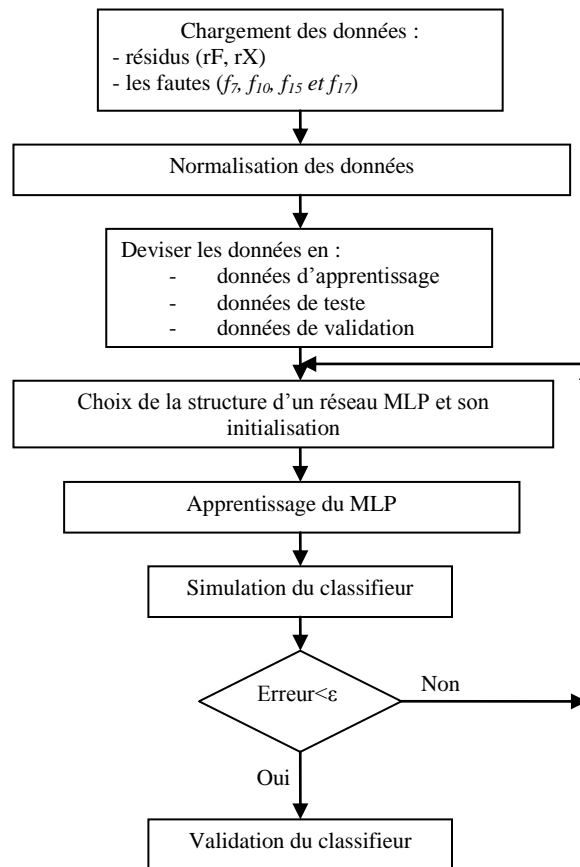


Figure IV.10: Procédure d'élaboration d'un classifieur neuronal.

Nous avons fait plusieurs essais (apprentissage) pour réaliser ce classifieur, la structure adoptée pour notre classifieur neuronal (class1) est donnée dans le tableau suivant :

Tableau IV.2: Apprentissage de réseaux de classification.

Le réseau	Couche 1	Couche 2	Couche de sortie	MSE
Class1	2	3	1	$64.37 * 10^{-3}$

Résultat de la simulation: L'apprentissage de ce classifieur, donne les résultats illustrés dans les figures ci-dessous.

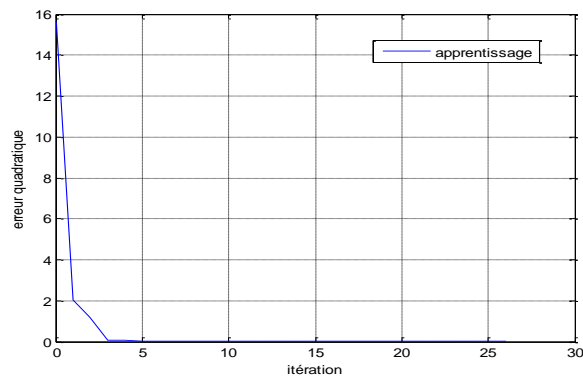


Figure IV.11: L'erreur quadratique d'apprentissage de Class1.

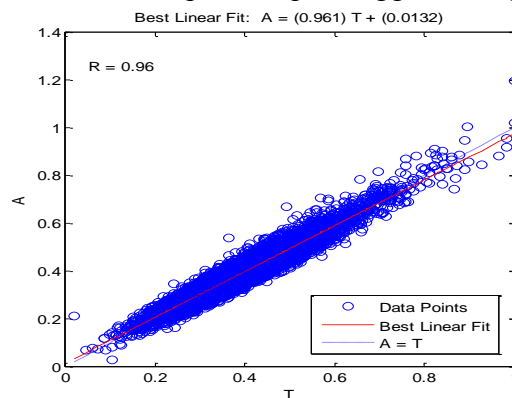


Figure IV.12: Régression linéaire entre la sortie de Class1 et la cible.

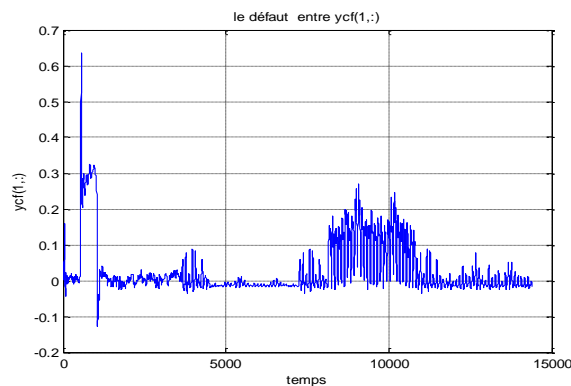


Figure IV.13 : La sortie f_7 du classifieur neuronal.

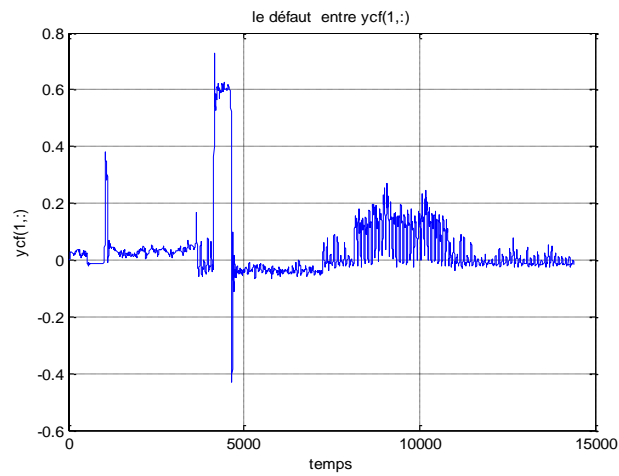


Figure IV.14 : La sortie f_{10} du classifieur neuronal.

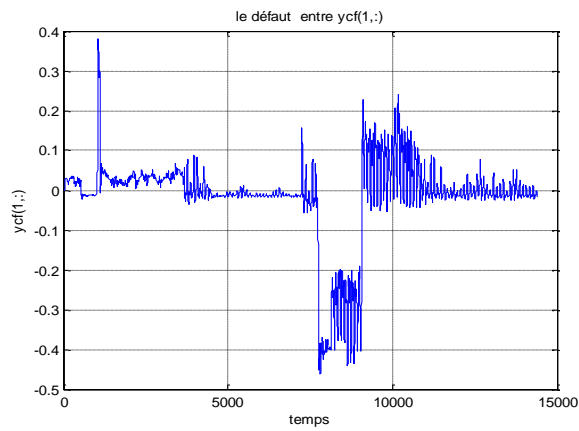


Figure IV.15: La sortie f_{15} du classifieur neuronal.

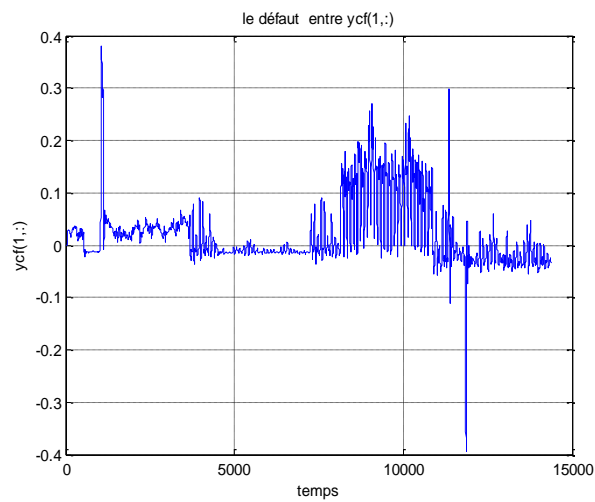


Figure IV.16 : La sortie f_{17} du classifieur neuronal.

On remarque qu'après l'apprentissage, les sorties des classifieurs f_7 , f_{10} , f_{15} et f_{17} ressemblent considérablement aux défauts réels survenus au niveau de l'actionneur. Donc comme on le voit ce genre de classifieur neuronal nous permet de distinguer les défauts survenus au niveau de l'actionneur.

IV.6.1.2 Diagnostic par la Logique Floue:

Pour le classifieur flou nous avons créé un système flou à deux entrées, et une seule sortie. Les variables d'entrée sont les deux résidus générés à l'aide des modèles neuronaux (netc). Après la détermination des degrés d'appartenance de chaque variable d'entrée à chaque état par des fonctions d'appartenances, les variables de sortie sont obtenus à l'aide d'une base de règles, cette base est définie à partir d'une analyse des résidus. [25]

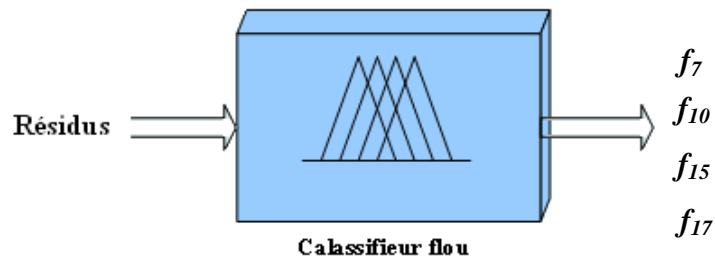


Figure IV.17: Classifieur flou des défauts.

Choix des fonctions d'appartenance et des intervalles

Les deux principales variables sont les deux résidus rX et rF . A partir de ces deux données obtenues dans la deuxième étape, le problème est de classifier les quatre défauts f_7 , f_{10} , f_{15} et f_{17} . Pour la fuzzification des variables d'entrée de notre classifieur rX et rF , nous avons associé une échelle de valeurs qui traduisent la classification.

GN : grand négatif

PN : petit négatif

Z : zéro

PP : petit positif

GP : grand positif

Et pour les variables de sortie nous avons opté pour :

TF : très faible

F : faible

M : moyenne

G : grande

TG : très grande

Les quatre tableaux d'inférence suivants résument la base de règles.

Tableau IV.3: Table d'inférence du défaut f_7 .

	rX				
rF	GN	PN	Z	PP	GP
GN	TF	TF	TF	G	G
PN	TF	TF	TF	TG	G
Z	TF	TF	TF	TF	TF
PP	TF	TF	TF	TF	TF
GP	TF	TF	TF	TF	TF

Tableau IV.4: Table d'inférence du défaut f_{10} .

	rX				
rF	GN	PN	Z	PP	GP
GN	TF	TF	G	TF	TF
PN	TF	TF	TG	TF	TF
Z	TF	TF	TF	TF	TF
PP	TF	TF	TF	TF	TF
GP	TF	TF	TF	TF	TF

Tableau IV.5: Table d'inférence du défaut f_{15} .

	rX				
rF	GN	PN	Z	PP	GP
GN	TF	G	TF	TF	TF
PN	TF	TG	TF	TF	TF
Z	TF	TF	TF	TF	TF
PP	TF	TF	TF	TF	TF
GP	TF	TF	TF	TF	TF

Tableau IV.6: Table d'inférence du défaut f_{17} .

	rX				
rF	GN	PN	Z	PP	GP
GN	TF	TF	TF	TF	TF
PN	TF	TF	TF	TF	TF
Z	TF	TF	TF	TF	TF
PP	G	TF	TF	TF	TF
GP	TG	TF	TF	TF	TF

Les fonctions d'appartenance sont illustrées dans les figures ci-dessous :

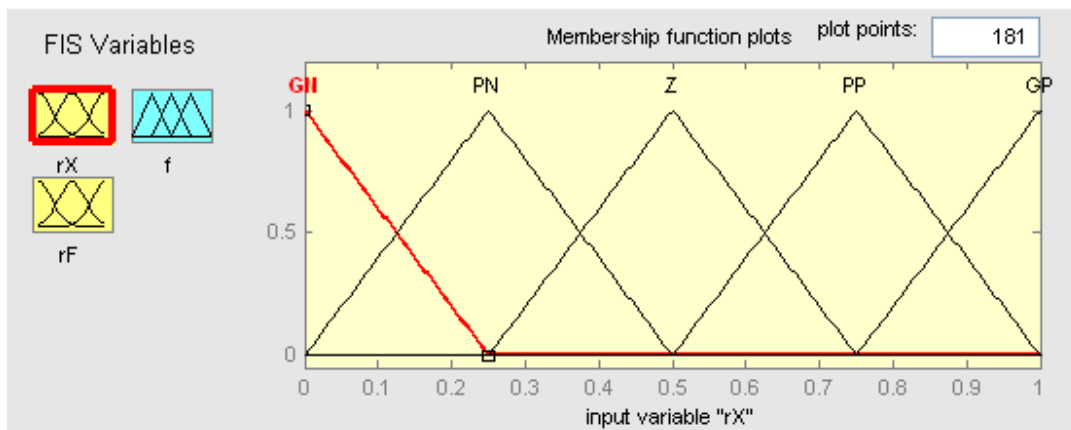


Figure IV.18: Choix des fonctions d'appartenance pour une variable d'entrée.

Résultat de la simulation:

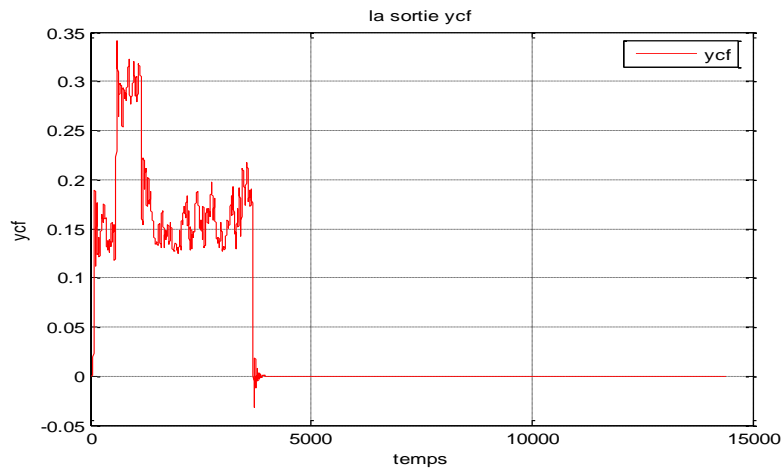


Figure IV.19: La sortie f_7 du classifieur flou.

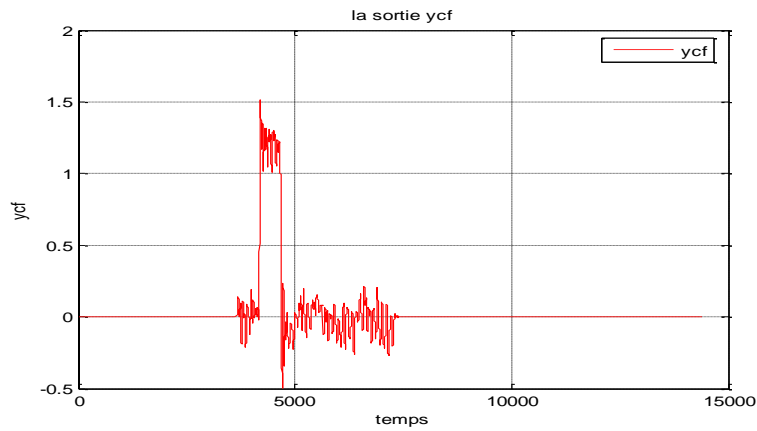


Figure IV.20: La sortie f_{10} du classifieur flou.

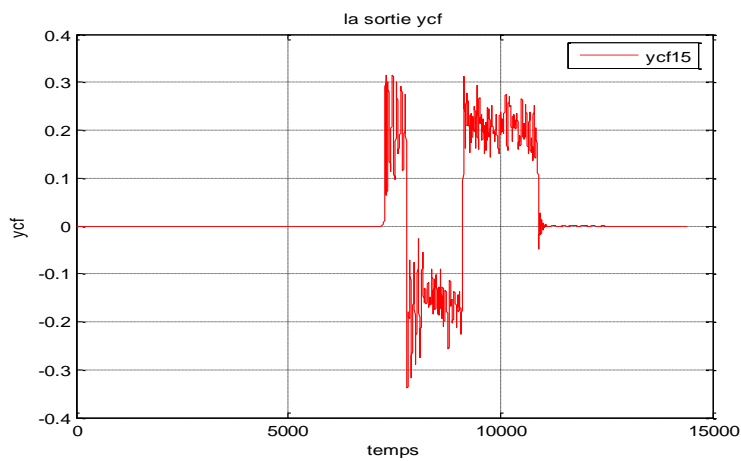


Figure IV.21: La sortie f_{15} du classifieur flou.

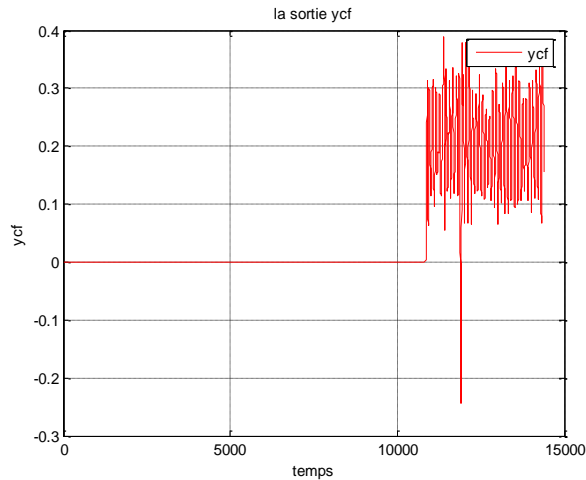


Figure IV.22: La sortie f_{17} du classifieur flou.

IV.6.1.3 Diagnostic par le système Neuro-Flou:

Pour le classifieur neuro-flou, nous avons créé une architecture hybride entre réseaux de neurones et logique floue. Notre classifieur se compose de deux réseaux neuro-flous, chaque réseau comportant deux entrées, une sortie. Les variables d'entrée sont les deux résidus rF et rX , et le variable de sortie est un indicateur concernant l'apparition des défauts f_7, f_{10}, f_{15} , ou f_{17} .

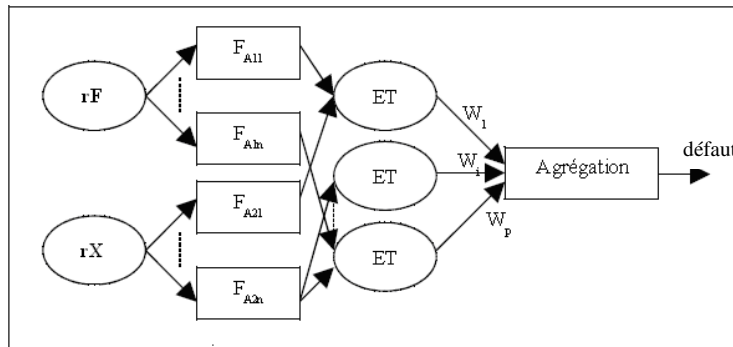


Figure IV.23: Architecture du classifieur neuro-flou des défauts.

Résultats de la simulation:

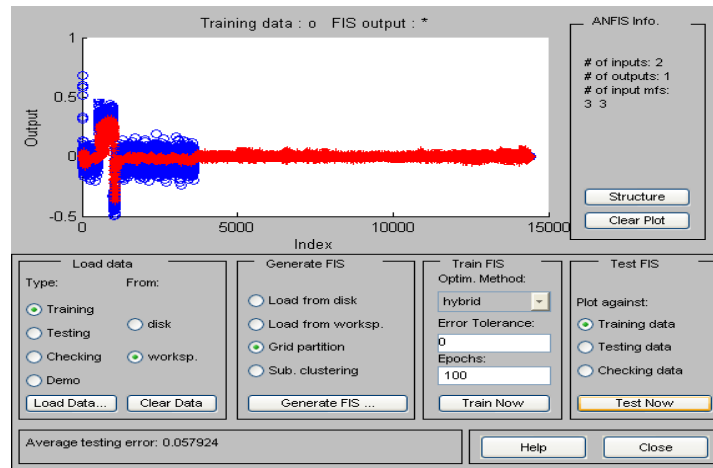


Figure IV.24: Représentation de l'apprentissage d'Anfis.

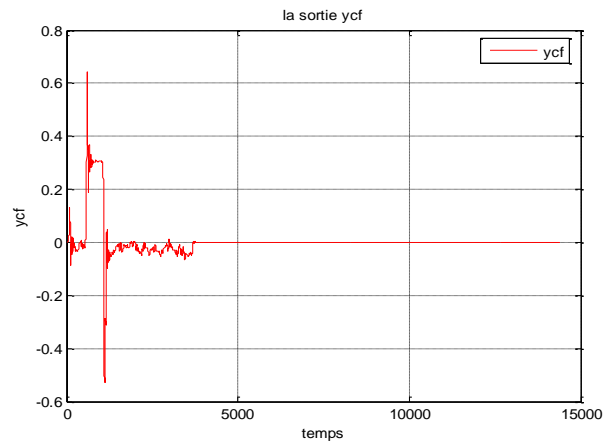


Figure IV.25: La sortie du classifieur neuro-flou dans le cas du défaut f_7 .

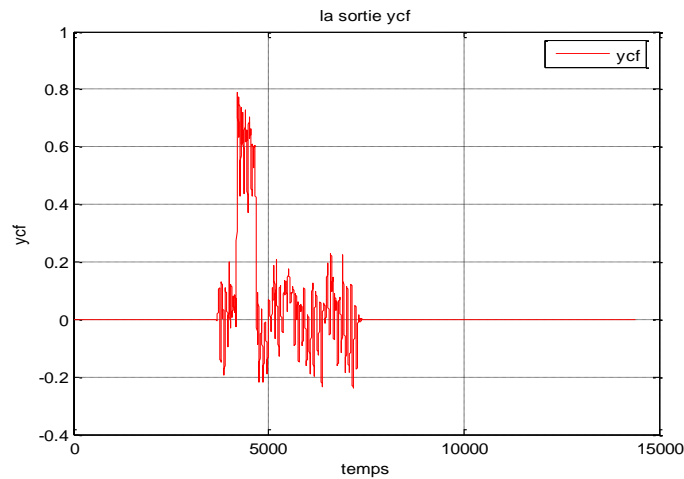


Figure IV.26: La sortie du classifieur neuro-flou Dans le cas du défaut f_{10} .

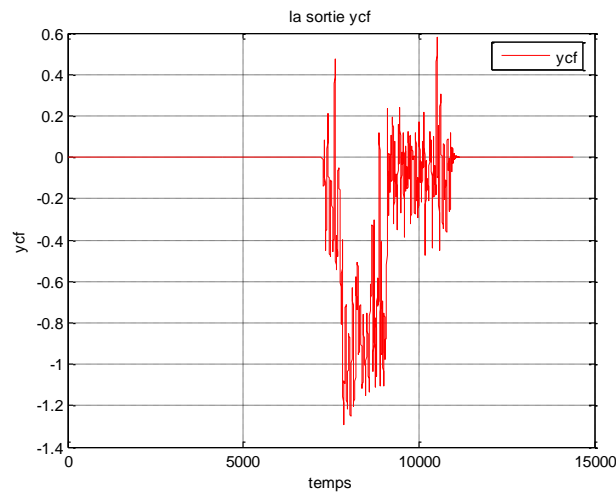


Figure IV.27: La sortie du classifieur neuro-flou dans le cas du défaut f_{15} .

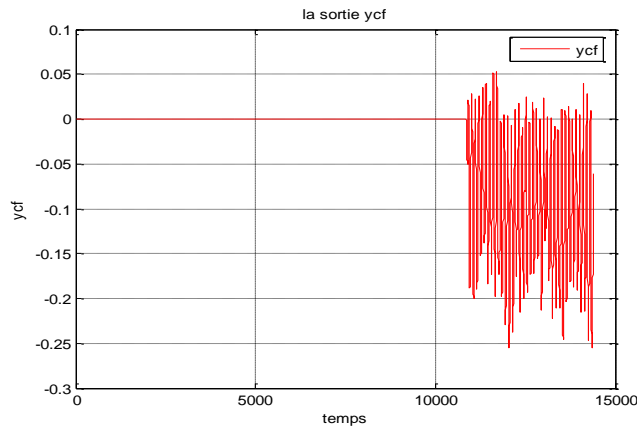


Figure IV.28: La sortie du classifieur neuro-flou dans le cas du défaut f_{17} .

IV.6.2 Diagnostic par analyse des modèles de dysfonctionnement:

Suivant les résultats de diagnostic que nous avons obtenus par les trois techniques citées précédemment (neuronale, floue et neuro-floue), on ne peut pas localiser les 19 défauts avec uniquement deux résidus r_X et r_F . Dans ce cas nous avons proposé une approche basée sur l'analyse des sorties obtenues avec les modèles NN de comportement sain et défectueux pour créer une base de données permettant de localiser tous les défauts existants dans le système DAMADICS (Figure IV.29). Les résultats du diagnostic sont donnés par la génération de résidus $R_i(t)$, avec $i=0..p$ selon un bloc de décision.

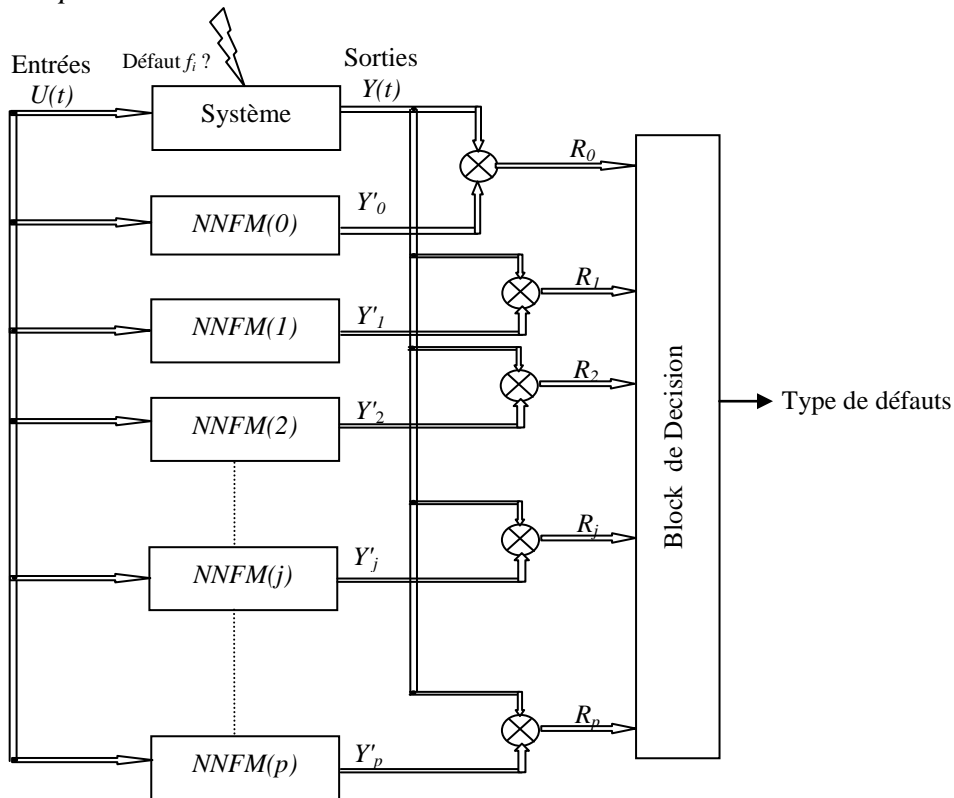


Figure IV.29: Conception d'un système FDI par des modèles de comportements sain et défectueux.

Cette technique est appliquée pour construire des modèles neuronaux correspondant aux 19 candidats de défauts considérés avec le benchmark DAMADICS. Dans ce contexte, il est nécessaire de créer une base de données qui contient des caractéristiques de tous les défauts [Kou, 11] présentés au système DAMADICS. La Figure IV.30 illustre la méthode pour le cas d'un défaut f_3 . Le réseau $NNFM(3)$ apprend la cartographie avec $q=6$ entrées et $n=2$ sorties dans le cas du défaut f_3 affectant le système à l'instant $t=0$:

$$(X'_3, F'_3) = NNFM(3) (CV, P_1, P_2, T_1, X_3, F_3) \quad (IV.15)$$

Pour sélectionner la structure de $NNFM(3)$, de nombreux tests ont été effectués afin d'obtenir les meilleures architectures. Les données d'apprentissage et de test ont été générés en utilisant Matlab-Simulink *DABLIB* [8]. La meilleure structure est un NN avec 6 neurones dans la première couche cachée, 3 neurones dans la deuxième couche cachée et deux neurones dans la couche de sortie. La validation se fait avec les données mesurées fournies par l'usine de sucre de Lublin en 2001[8].

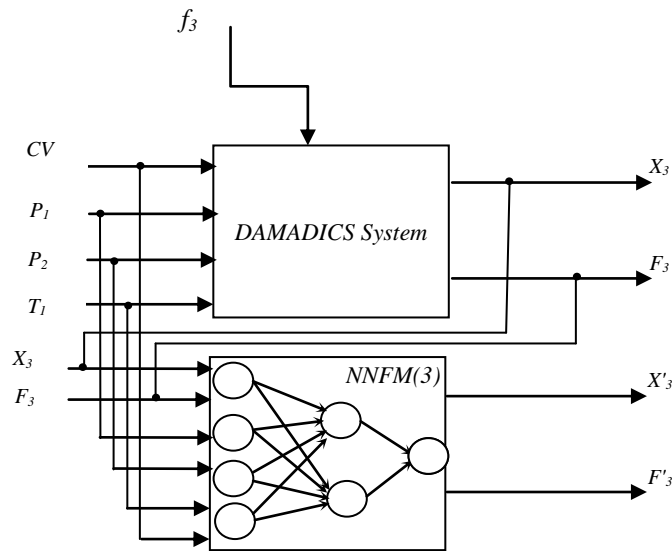


Figure IV.30: Modèle neuronal $NNFM(3)$ pour le cas du défaut f_3 .

IV.6.2.1 Matrice des signatures:

Nous avons fait au début de notre travail la modélisation du système DAMADICS par les réseaux de neurones en utilisant les données générées par le simulateur *datageneration* de *DABLIB* de durée de 3600s, les résultats de modélisation obtenus sont ci-dessous. Pour faire l'étape de génération des résidus nous avons simulé les 19 défauts par le même simulateur et les avons comparé par rapport aux sorties de notre modèle neuronal sans défaut. Pour évaluer les résidus et les localiser, on a besoin de choisir des seuils et de tracer le tableau de signatures des 19 défauts présentés ci-dessous. On obtient des sous ensembles de défauts non localisables : par exemple les signatures des défauts $f_3, f_6, f_9, f_{12}, f_{18}$ et f_{19} sont identiques. Il en est de même pour les signatures des défauts f_5, f_8 et f_{14} qui sont identiques au cas *sans-défaut* et pour les signatures des défauts f_1, f_7, f_{10} et f_{15} et f_{17} . Par contre les défauts f_2, f_{11}, f_{13} et f_{16} sont localisables. Pour localiser les défauts des mêmes signatures des sous ensemble nous avons proposé de créer un autre réseau de neurone avec les entrées X et F et la sortie correspondant à l'un des résidus du sous ensemble (Figure IV.31). Ce résidu doit être sensible à l'un des défauts et robuste par rapport aux autres défauts de ce même sous ensemble.

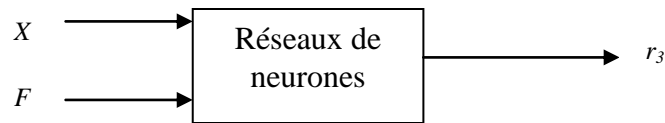
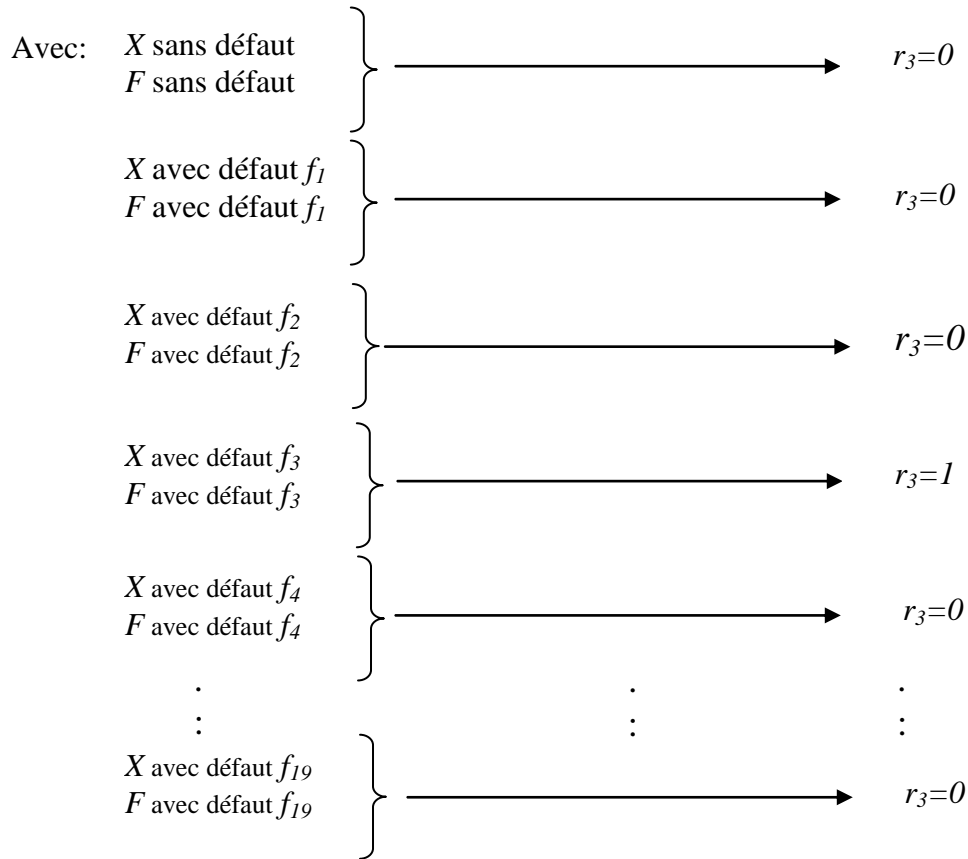


Figure IV.31: Générateur de résidu à base des réseaux de neurones.



Malheureusement, cette technique ne donne pas de solution satisfaisante pour localiser les 19 défauts du système DAMADICS. C'est pour cela on a cherché une autre méthode pour résoudre ce problème. La technique développée par la suite est basée sur l'analyse des résidus et la création des modèles des comportements sain et défectueux.

On génère des résidus sans défaut et avec les 19 défauts en utilisant le simulateur de DAMADICS pour créer une matrice des signatures. Les résultats de génération des résidus et ses analyses sont:

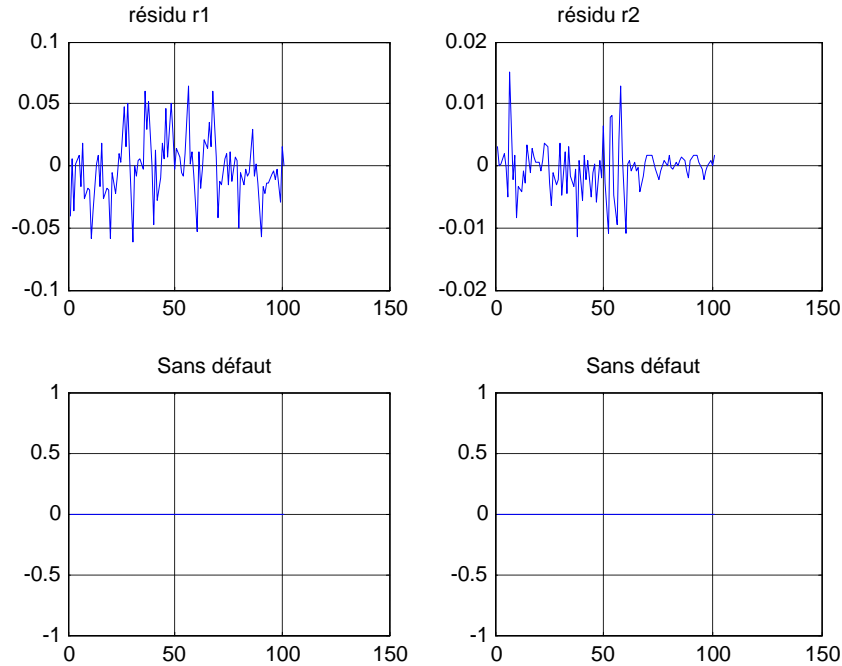


Figure IV.32: Génération des résidus sans défaut.

1) *Le choix du seuil* : La variance du résidu r_1 (Figure IV.32) dans le cas sans défaut est égale à $6.71 \cdot 10^{-4}$. On choisit alors le seuil pour le 1^{er} résidu égale à 5 fois l'écart type du résidu r_1 . L'écart type du résidu r_1 est $\sigma_1 = 0.0259$; La moyenne du résidu r_1 est $m_1 = 0.002$; le seuil $S_1 = 5 * \sigma_1 = \pm 0.1295$.

Pour le seuil du signal de r_2 on choisit le seuil suivant la variance du r_2 . La valeur moyenne du résidu $r_2 = -3.47 \cdot 10^{-4}$; L'écart type du résidu $r_2 = 0.004$; le seuil est $S_2 = 5 * \sigma_2 = \pm 0.0199$.

Nous avons utilisé le *DABLIB* (*simulateur de DAMADICS*) pour réaliser les simulations:

$$X = f(CV, P1, P2, T1)$$

$$F = f(X, P1, P2, T1)$$

(IV.16)

Nous avons modélisé l'actionneur par les réseaux de neurones *MLP* en utilisant les données générées par le modèle *DABLIB* (*Benchmark Datageneration*). Nous avons trouvé les résultats de simulation suivants :

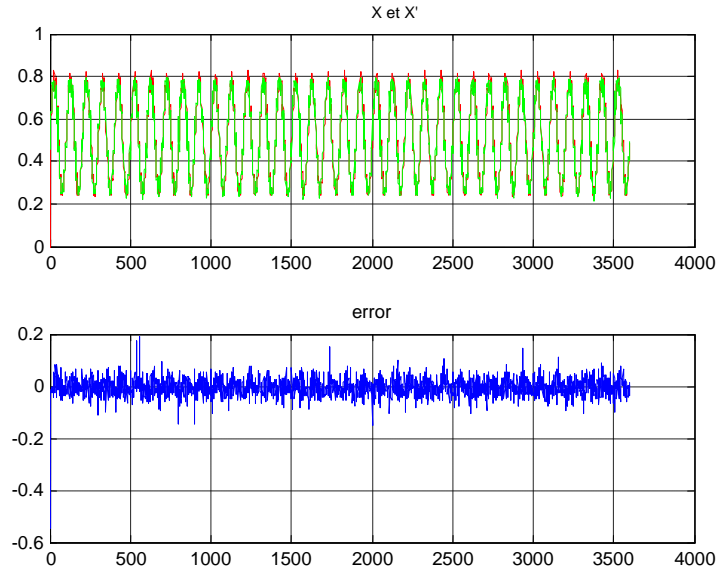


Figure IV.33: La sortie X du système et la sortie X' du modèle neuronal.

L'erreur $e_1 = 0,0027$ pour le réseau de la sortie X (Figure IV.33) et l'erreur $e_2 = 5,27 \times 10^{-5}$ pour le réseau de la sortie F (Figure IV.34).

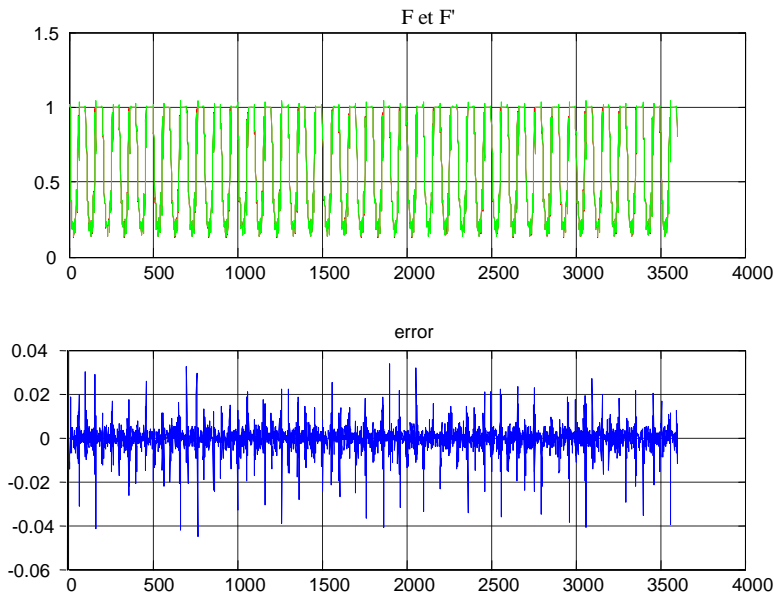


Figure IV.34: La sortie F du système et la sortie F' du modèle neuronal.

Les défauts artificiels de types abrupts sont simulés à partir de l'instant 20 jusqu'au l'instant 80. Nous avons utilisé les résidus:

$$\begin{aligned} r_1 &= X_{ess_NN} - X_{ess_f}; \\ r_2 &= F_{ess_NN} - F_{ess_f}; \end{aligned} \tag{IV.17}$$

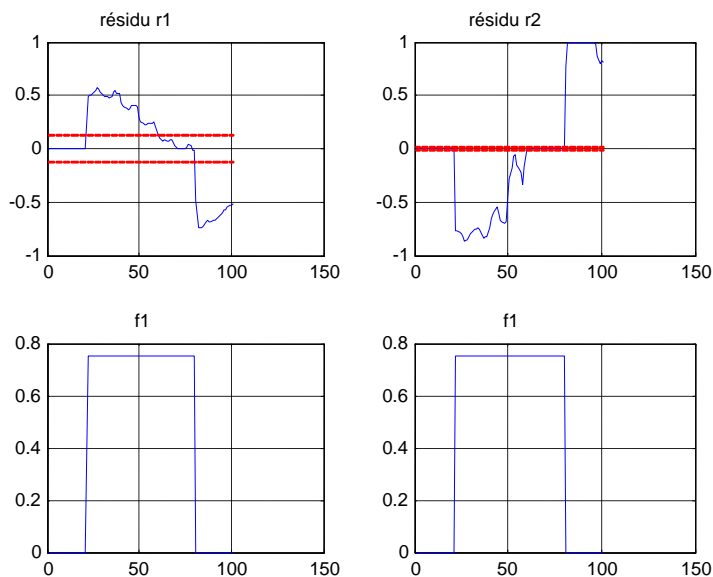


Figure IV.35: Génération des résidus en cas de défaut f_1 .

On observe sur cette figure que le défaut f_1 apparaît sur les deux sorties : il est détectable

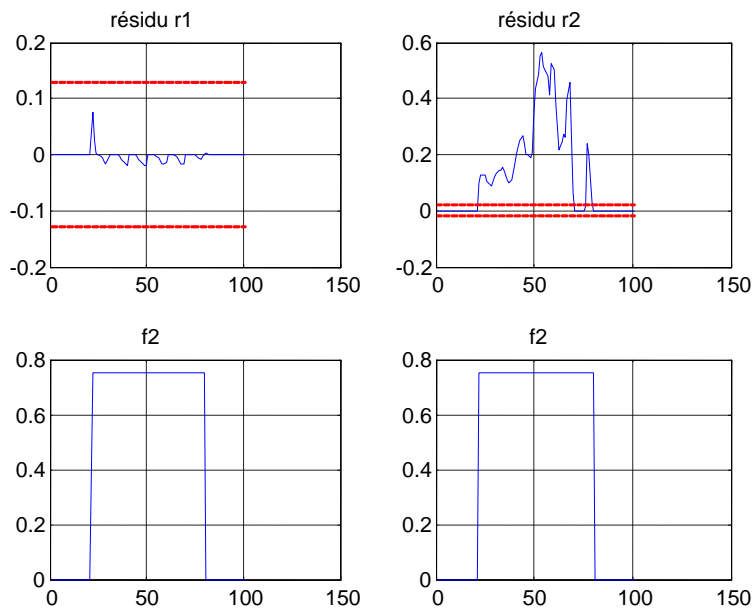


Figure IV.36: Génération des résidus en cas de défaut f_2 .

On observe sur cette figure que le défaut f_2 apparaît sur une seule sortie : il demeure détectable.

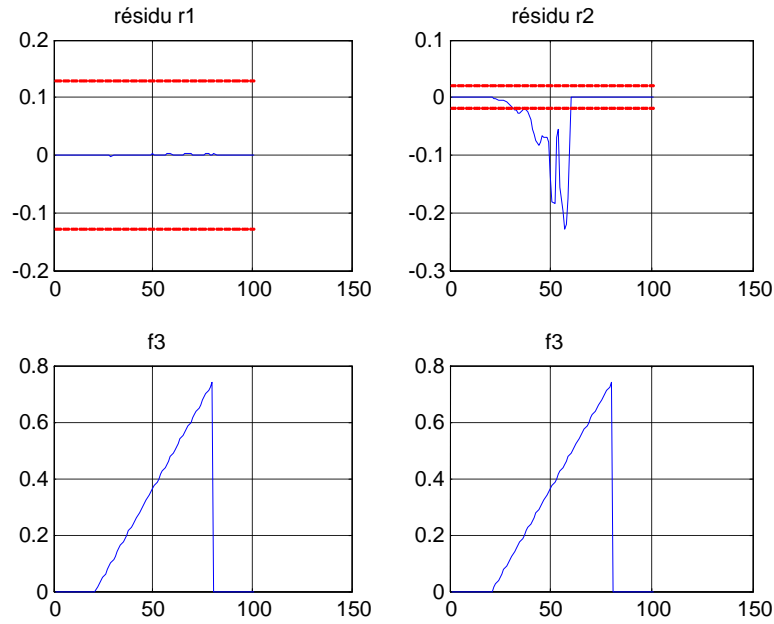


Figure IV.37: Génération des résidus en cas de défaut f_3 .

On observe sur cette figure que le défaut f_3 apparaît sur une seule sortie : il demeure détectable.

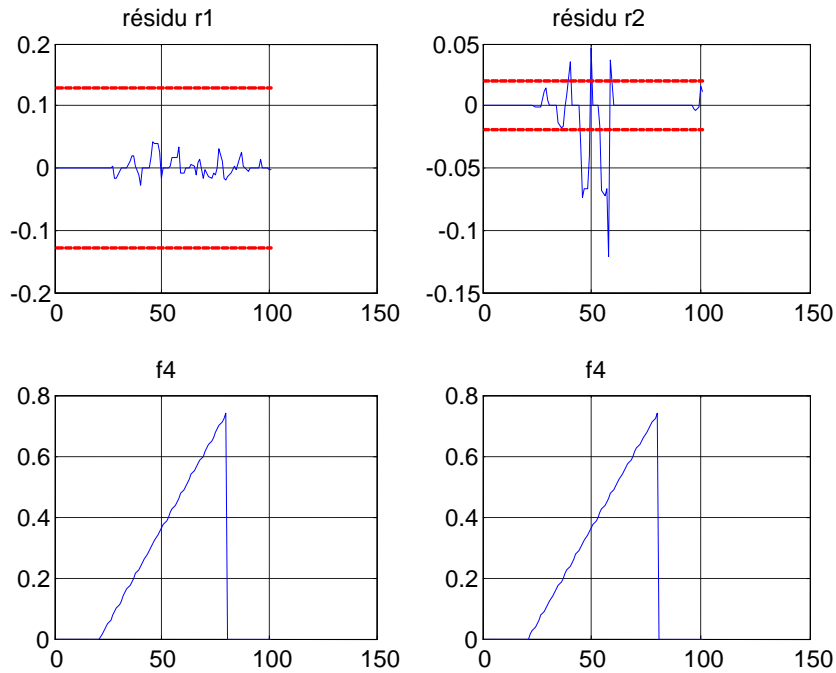


Figure IV.38: Génération des résidus en cas de défaut f_4 .

On observe sur cette figure que le défaut f_4 apparaît sur les deux sorties : il est détectable.

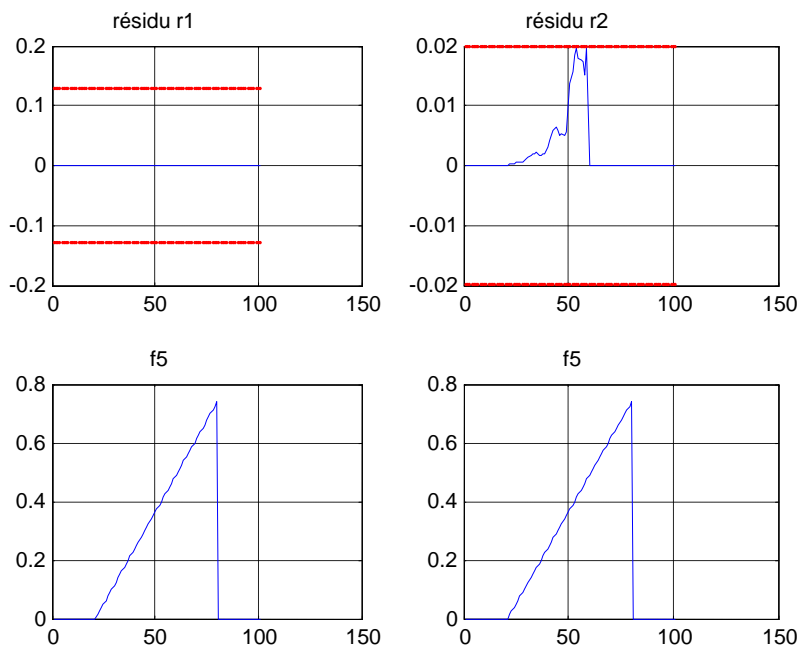


Figure IV.39: Génération des résidus en cas de défaut f_5 .

On observe sur cette figure que le défaut f_5 apparaît sur une seule sortie : il demeure détectable.

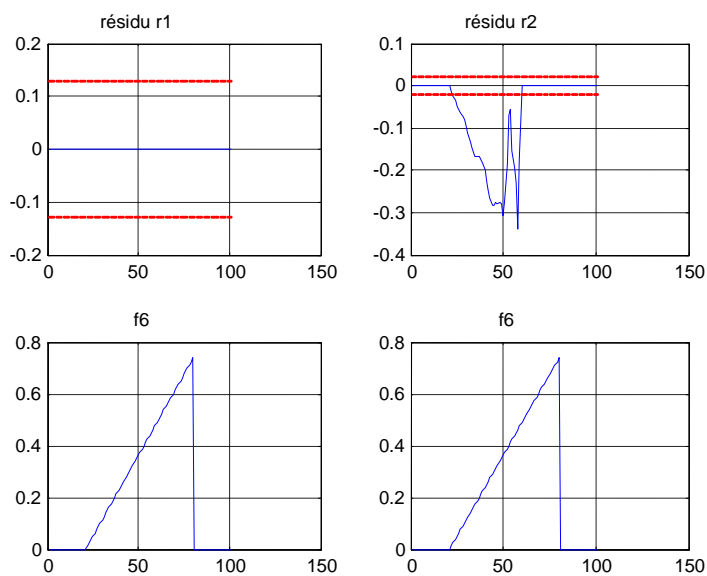


Figure IV.40: Génération des résidus en cas de défaut f_6 .

On observe sur cette figure que le défaut f_6 apparaît sur une seule sortie : il demeure détectable.

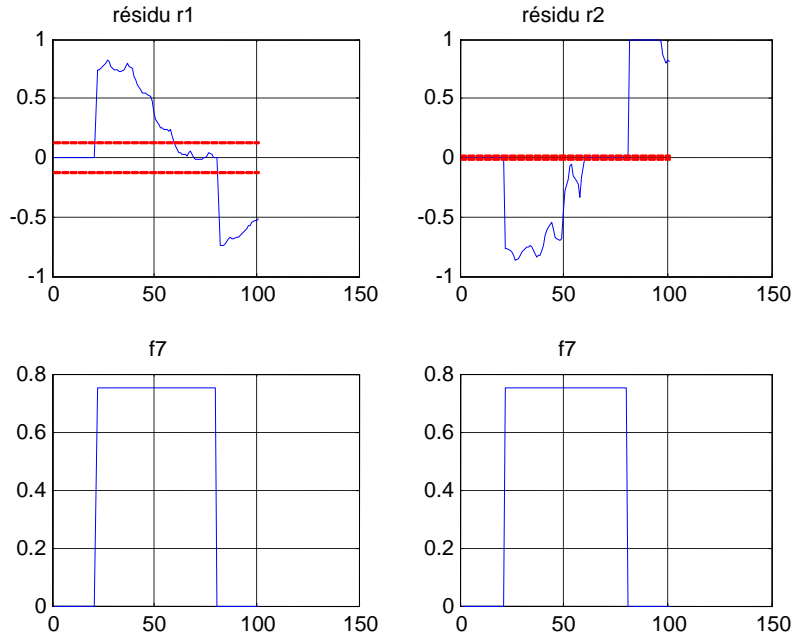


Figure IV.41: Génération des résidus en cas de défaut f_7 .

On observe sur cette figure que le défaut f_7 apparaît sur les deux sorties : il est détectable.

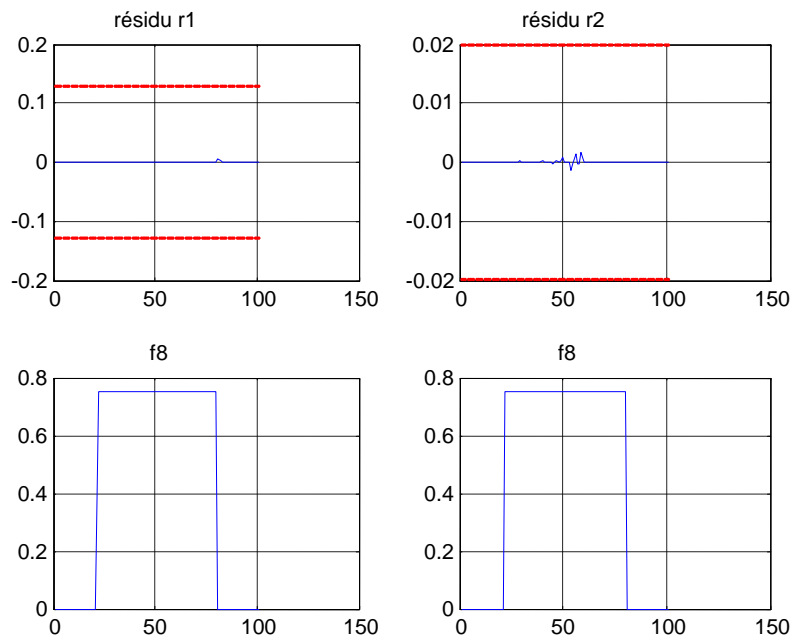


Figure IV.42: Génération des résidus en cas de défaut f_8 .

Par contre on observe sur cette figure que le défaut f_{14} n'apparaît pas sur les sorties : il est non détectable.

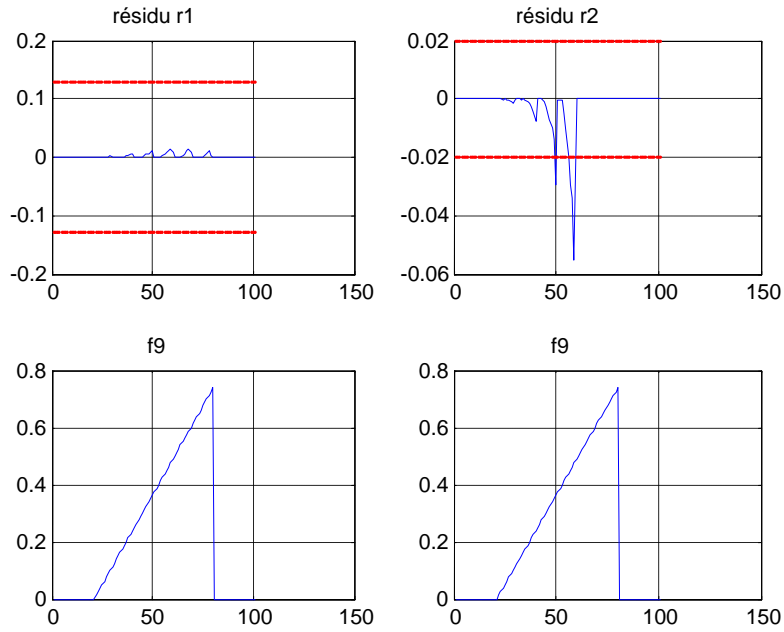


Figure IV.43: Génération des résidus en cas de défaut f_9 .

On observe sur cette figure que le défaut f_9 apparaît sur les deux sorties : il est détectable.

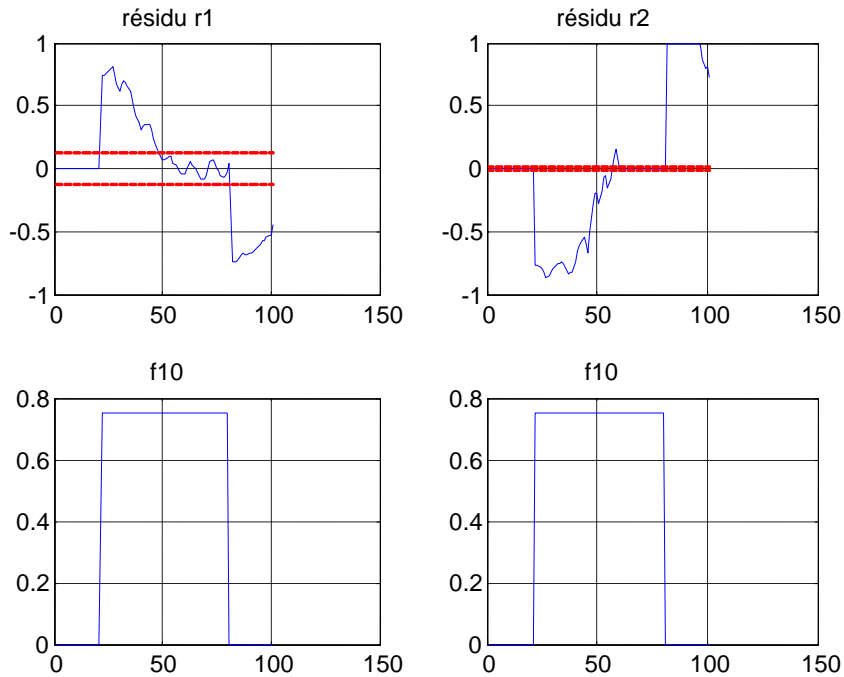


Figure IV.44: Génération des résidus en cas de défaut f_{10} .

On observe sur cette figure que le défaut f_{10} apparaît sur les deux sorties : il est détectable.

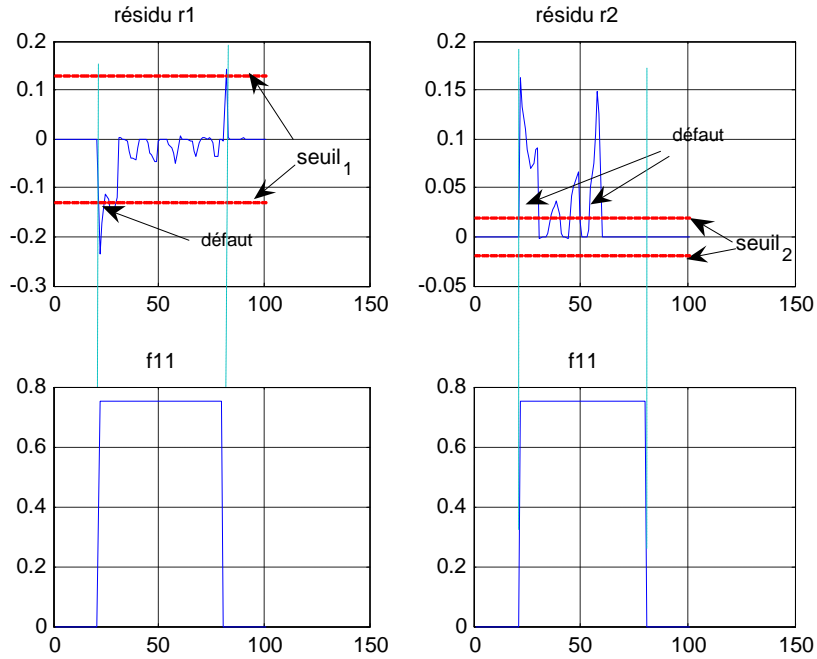


Figure IV.45: Génération des résidus en cas de défaut f_{11} .

On observe sur cette figure que le défaut f_{11} apparaît sur les deux sorties : il est détectable.

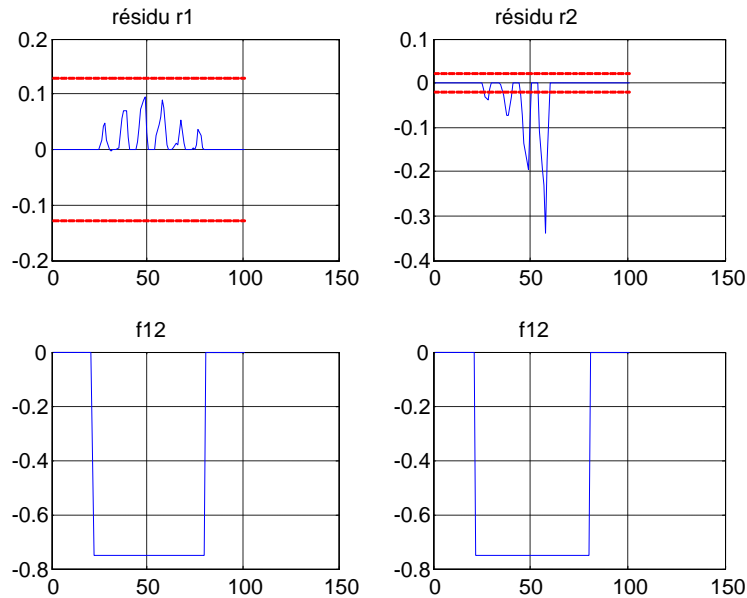


Figure IV.46: Génération des résidus si le défaut f_{12} est présent.

On observe sur cette figure que le défaut f_{12} apparaît sur les deux sorties : il est détectable.

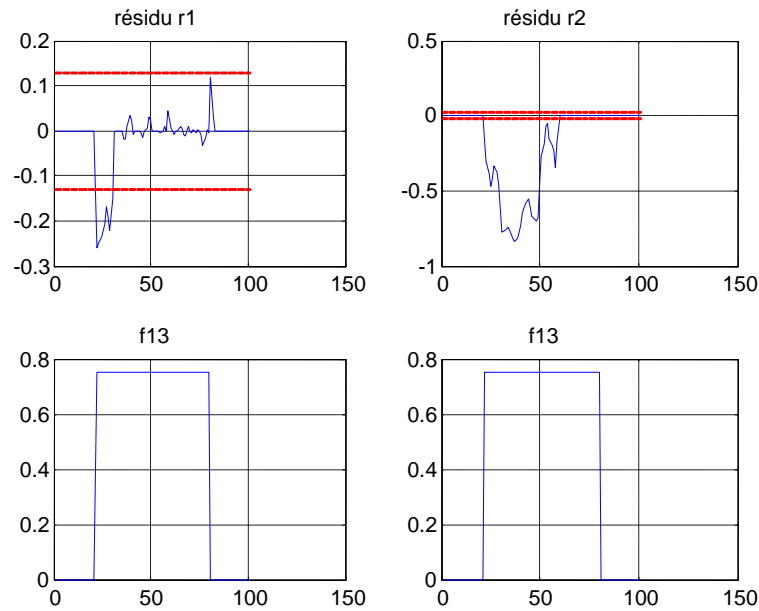


Figure IV.47: Génération des résidus en cas de défaut f_{13} .

On observe sur cette figure que le défaut f_{13} apparaît sur les deux sorties : il est détectable.

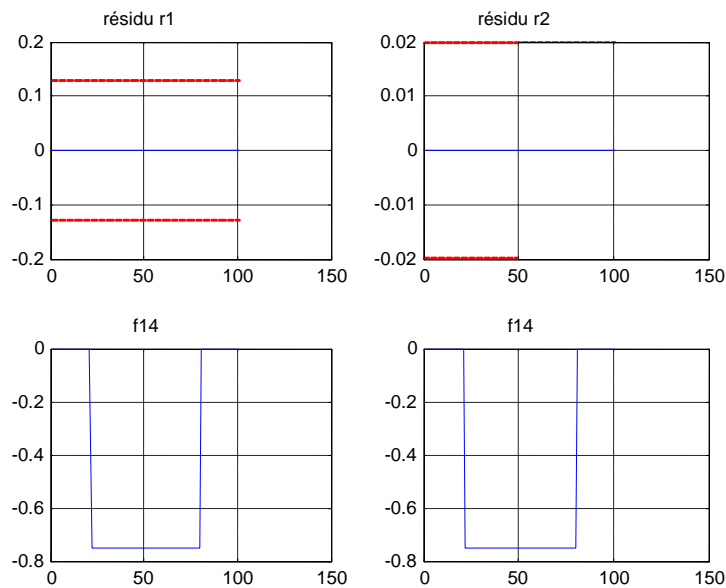


Figure IV.48: Génération des résidus en cas de défaut f_{14} .

Par contre on observe sur cette figure que le défaut f_{14} n'apparaît pas sur les sorties : il est non détectable.

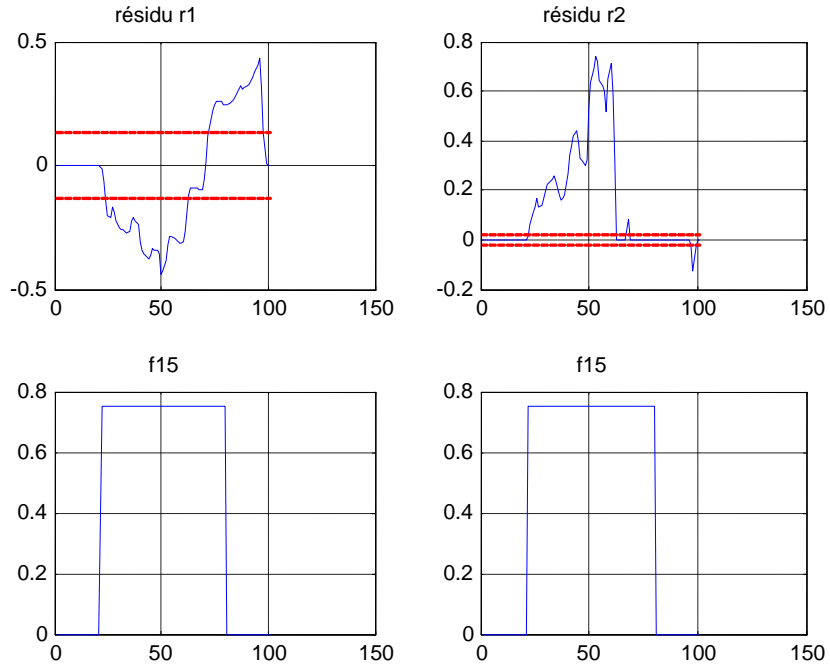


Figure IV.49: Génération des résidus en cas de défaut f_{15} .

On observe sur cette figure que le défaut f_{15} apparait sur les deux sorties : il est détectable.

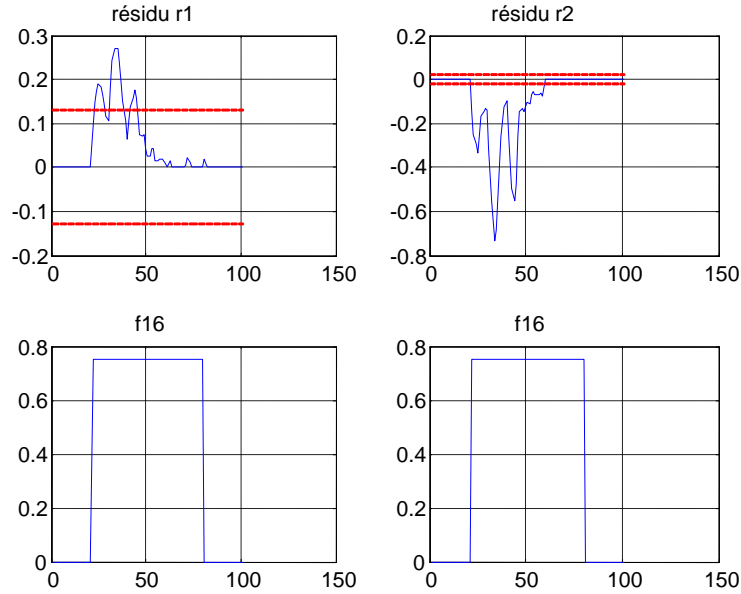


Figure IV.50: Génération des résidus en cas de défaut f_{16} .

On observe sur cette figure que le défaut f_{16} apparait sur les deux sorties : il est détectable.

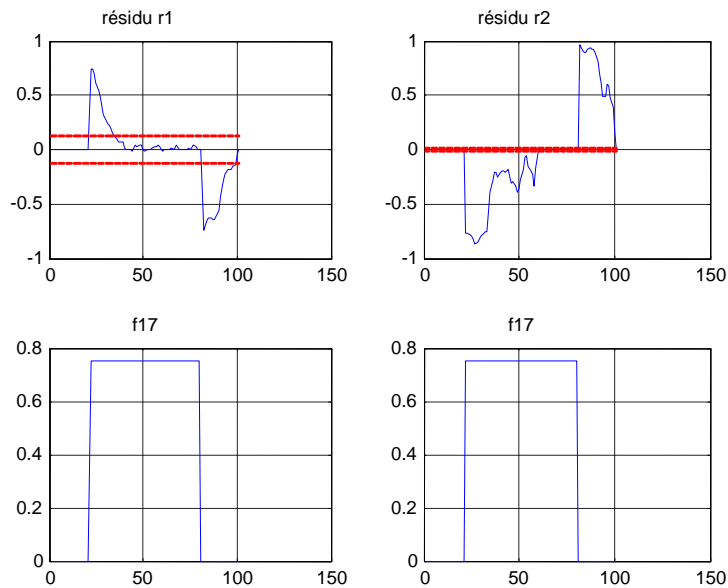


Figure IV.51: Génération des résidus en cas de défaut f_{17} .

On observe sur cette figure que le défaut f_{17} apparait sur les deux sorties : il est détectable.

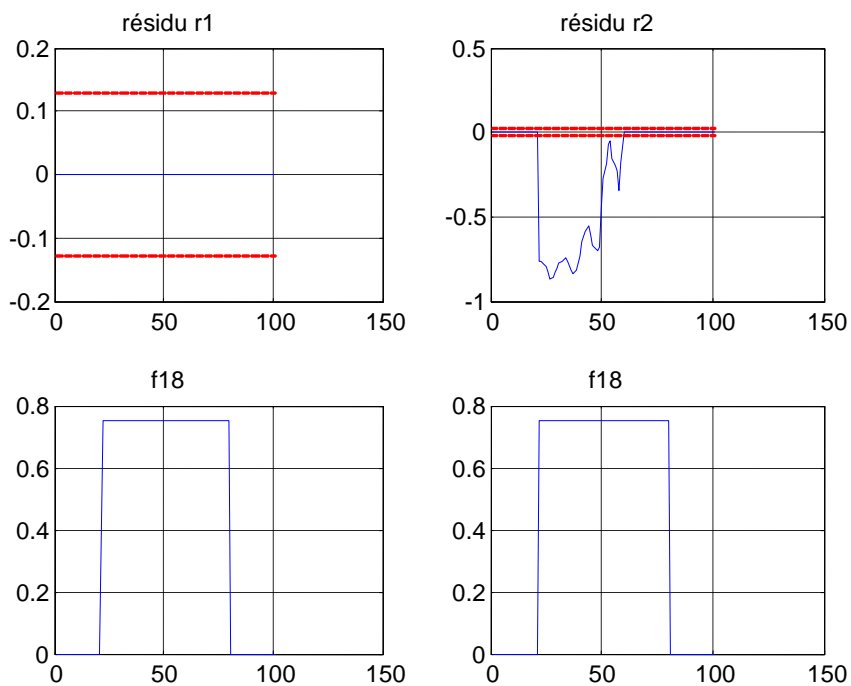


Figure IV.52: Génération des résidus en cas de défaut f_{18} .

On observe sur cette figure que le défaut f_{18} apparait sur une seule sortie : il demeure détectable.

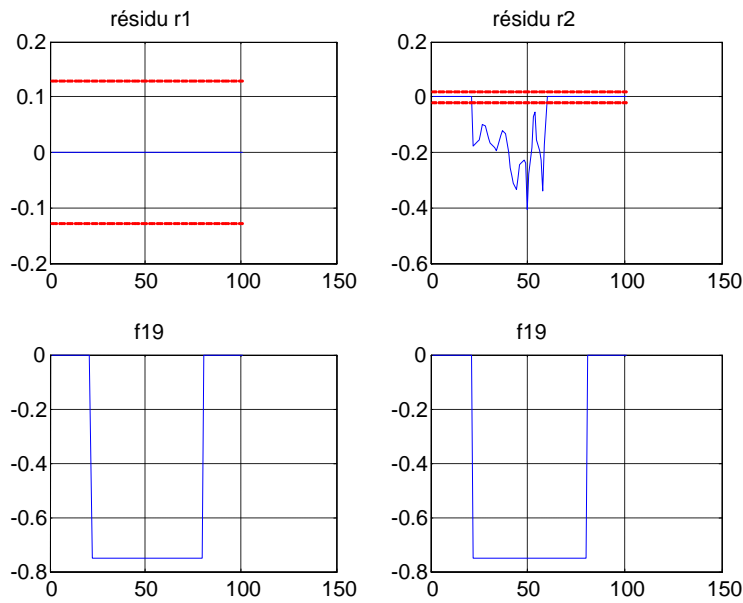


Figure IV.53: Génération des résidus en cas de défaut f_{19} .

On observe sur cette figure que le défaut f_{19} apparaît sur une seule sortie : il demeure détectable.

Le vecteur de résidu $R_0(t) = (r_{k0}(t))$, avec $k = 1, 2$ est d'abord examiné pour la détection des défauts selon l'équation (IV.17):

$$\begin{aligned} r_{10}(t) &= X(t) - X'(t) \\ r_{20}(t) &= F(t) - F'(t) \end{aligned} \quad (\text{IV.18})$$

Où X' et F' sont les sorties du modèle NN des comportements sans défaut.

La détection est obtenue selon la comparaison des résidus avec des seuils appropriés. Une triple évaluation des signaux : positifs, négatifs et zéro, est utilisée. Les seuils sont élaborés en fonction de l'écart-type des résidus dans le cas sans défaut [Kou, 11]. Notons que le choix des seuils constants ou adaptatifs influence fortement les performances du système de FDI (Tableau IV.8). Les seuils doivent être soigneusement choisis. Le tableau IV.7 résume les performances de détection pour les 19 types de défauts selon le signe des valeurs du vecteur des résidus R_0 .

L'analyse des résidus est une étape essentielle dans les procédures FDI. Le problème de la sélection du seuil est étroitement lié au comportement des résidus et aussi à des contraintes qui peuvent être imposées, comme la sécurité des marges de tolérance [Lef, 10]. Pour cette raison, nous étudions la variation du retard à la détection ($t_f - \tau$) en fonction de l'ampleur des seuils de détection S_{10} et S_{20} . Les résultats sont résumés dans le tableau IV.7.

Tableau IV.7: Les signatures des 19 défauts.

Résidus défauts	$r_{10} > 0$	$r_{10} < 0$	$r_{20} > 0$	$r_{20} < 0$	Type de Groupe
Sans-défaut	0	0	0	0	G ₃
f_1	1	1	1	1	G ₂
f_2	0	0	1	0	Isolé
f_3	0	0	0	1	G ₁
f_4	0	0	1	1	Isolé
f_5	0	0	0	0	G ₃
f_6	0	0	0	1	G ₁
f_7	1	1	1	1	G ₂
f_8	0	0	0	0	G ₃
f_9	0	0	0	1	G ₁
f_{10}	1	1	1	1	G ₂
f_{11}	1	1	1	0	Isolé
f_{12}	0	0	0	1	G ₁
f_{13}	0	1	0	1	Isolé
f_{14}	0	0	0	0	G ₃
f_{15}	1	1	1	1	G ₂
f_{16}	1	0	0	1	Isolé
f_{17}	1	1	1	1	G ₂
f_{18}	0	0	0	1	G ₁
f_{19}	0	0	0	1	G ₁

Tableau IV.8: Le retard à la détection en fonction de σ .

Les défauts	Résidus	σ	2σ	3σ	4σ	5σ	6σ	7σ
f_{12} $t_f=500s$	r_{10}	2s	3s	12s	39s	40s	60s	ND
	r_{20}	2s	2s	2s	2s	2s	2s	2s
f_{18} $t_f=231s$	r_{10}	FA	ND	ND	ND	ND	ND	ND
	r_{20}	1s	2s	2s	2s	2s	2s	2s
f_{11} $t_f=485s$	r_{10}	FA	2s	2s	2s	2s	3s	ND
	r_{20}	FA	2s	2s	2s	2s	2s	2s
f_{15} $t_f=444s$	r_{10}	7s	7s	8s	8s	9s	17s	51s
	r_{20}	FA	FA	7s	7s	7s	8s	8s
f_3 $t_f=405s$	r_{10}	FA	ND	ND	ND	ND	ND	ND
	r_{20}	FA	FA	29s	29s	35s	45s	47s

Selon le tableau IV.8, on remarque que si les seuils de détection augmentent, le délai de détection augmente également de manière exponentielle et peut aboutir à des non détections (ND). A l'inverse, les petites valeurs des seuils peuvent conduire à de fausses alarmes (FA). Il nécessite alors un choix judicieux des seuils pour chaque résidu. Dans notre travail les seuils choisis sont : $S_{10} = 5 * \sigma_1$ et $S_{20} = 5 * \sigma_2$ où σ_1 et σ_2 sont les écarts-types obtenus à partir du processus d'apprentissage. Pour ces choix des seuils, il apparaît trois groupes G_1 , G_2 , et G_3 montrés ci-dessous.

- $G_1 = \{f_3, f_6, f_9, f_{12}, f_{18}, f_{19}\}$
- $G_2 = \{f_1, f_7, f_{10}, f_{15}, f_{17}\}$
- $G_3 = \{\text{Sans-défaut}, f_5, f_8, f_{14}\}$

Les défauts ayant les mêmes signatures sont regroupés dans les trois groupes précédents. Les défauts du G_1 et du G_2 sont détectés mais pas isolés, car les signatures de résidus r_{10} et r_{20} sont similaires. On constate aussi que les défauts dans le groupe G_3 ont la même signature que les comportements *sans-défaut* et ne sont pas directement détectables avec les résidus r_{10} et r_{20} . A partir du tableau IV.7. Les défauts f_2, f_4, f_{11}, f_{13} et f_{16} ont chacun des signatures particulières, ils sont donc localisables.

A la différence de la première technique (modèle de connaissance de comportement sain et défectueux $FM(j)$), les sorties estimées $X'_j(t)$ et $F'_j(t)$ sont fournis par les modèles neuronaux des comportements pour le cas sain et défectueux ($NNFM$) de chacun des défauts candidats des 3 groupes à partir des données historiques fournies par benchmark DAMADICS. Pour chaque modèle $NNFM(j)$ les résidus $r_{1j}(t)$ et $r_{2j}(t)$ avec $j \in \{3, 6, 9, 12, 18, 19, 1, 7, 10, 15, 17, 5, 8, 14\}$ sont fournis en faisant la différence des sorties estimées ($X'_j(t), F'_j(t)$) et des sorties mesurées.

IV.6.2.2 Probabilité des modèles:

Les défauts dans les groupes G_1 et G_2 sont détectables, mais pas localisables avec les résidus r_{10} et r_{20} , car ils possèdent des signatures similaires. On peut également remarquer que les défauts du groupe G_3 ont la même signature que les comportements sans défaut. Ainsi les défauts dans le groupe G_3 ne sont pas directement détectables avec les mêmes résidus.

Pour illustrer notre contribution, nous avons choisis arbitrairement les défauts f_3, f_{15} et f_5 pour trois simulations distinctes. Le défaut f_3 du groupe G_1 est de type naissant (incipient) et simulé dans l'intervalle de temps $[487s \ 1000s]$. Le défaut f_{15} du groupe G_2 est de type abrupt et simulé dans l'intervalle de temps $[451s \ 1000s]$. Enfin le défaut f_5 est aussi de type naissant et simulé pendant l'intervalle de temps $[302s, \ 1000s]$. Les seuils de détection sont choisis de telle sorte que $S_{10} = 10 * \sigma_1$ et $S_{20} = 10 * \sigma_2$, avec $\sigma_1 = 7.047 * 10^{-6}$ et $\sigma_2 = 1.065 * 10^{-5}$. En fonction du stade de détection, le défaut f_3 est détecté à l'instant $\tau = 501s$ avec un retard de $14s$. Le défaut f_{15} est détecté à l'instant $\tau = 458s$ avec un retard de $7s$. Le défaut f_5 est non détectable avec la technique de seuillage. Dans chaque groupe, les défauts ne sont pas localisables. Pour cette raison, nous proposons d'utiliser la méthode décrite dans la section IV.4 afin de permettre le diagnostic rapide en ligne de tous ces défauts. A cet effet, nous avons introduit 19 modèles des comportements défectueux $FM(j)$ avec $j=1, \dots, 19$ issus de l'historique des données disponibles au niveau du simulateur $DABLIB$. Chaque modèle des comportements défectueux $FM(j)$ produit deux sorties estimés $X'(t, f_j, \tau)$ et $F'(t, f_j, \tau)$ comparées avec des données mesurées pour générer les résidus $r_{1j}(t, f_j, \tau)$ et $r_{2j}(t, f_j, \tau)$:

$$\begin{aligned} r_{1j}(t, f_j, \tau) &= X(t) - X'(t, f_j, \tau) \\ r_{2j}(t, f_j, \tau) &= F(t) - F'(t, f_j, \tau) \end{aligned} \quad (IV.19)$$

IV.6.2.3 Résultat de diagnostic de défauts:

Les résultats de diagnostic sont donnés dans le tableau IV.9, et illustrés par les figures IV.54 à IV.57. On définit les résidus cumulatifs $\rho_{1i}(t, T, \tau), \rho_{2i}(t, T, \tau)$, et la distance euclidienne $D_f(t, T, \tau)$, selon les équations IV.5 et IV.6. Le diagnostic hors ligne ou en ligne est obtenu à partir des équations (IV.8) ou (IV.11).

Par exemple, f_{15} est simulé dans un intervalle de temps $[444s \ 1000s]$ (figure IV.54). En fonction du stade de détection, un défaut est détecté à l'instant $\tau = 451s$, et les défauts du groupe G_2 sont isolés. Le diagnostic hors ligne est illustré avec un intervalle de temps large, avec $T = t = 1000s$ par exemple.

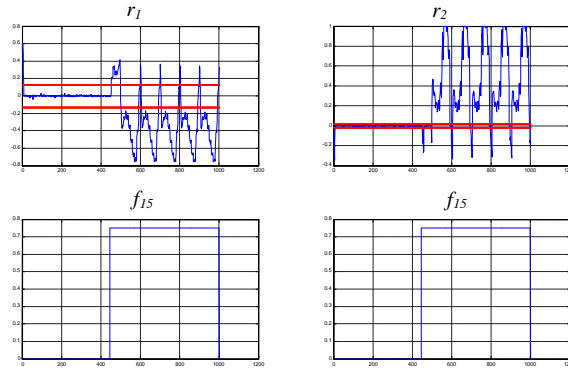


Figure IV.54: Résidus r_{10} , r_{20} lorsque le défaut f_{15} est simulé dans l'intervalle [444s 1000s].

Tableau IV.9: Diagnostic hors ligne, selon les modèles de connaissance défectueux.

FM(j) modèles de connaissance défectueux			
Défaut Candidat f_j	$\rho_{1j}(1000, 1000, 451)$	$\rho_{2j}(1000, 1000, 451)$	$D_j(1000, 1000, 451)$
f_1	14.96	20.02	25.00
f_2	9.94	11.23	15.00
f_3	10.03	14.79	17.88
f_4	9.99	14.29	17.44
f_5	10.01	13.47	16.78
f_6	10.01	18.13	20.71
f_7	14.15	20.02	24.52
f_8	10.01	13.57	16.86
f_9	10.08	13.65	16.97
f_{10}	11.33	16.65	20.14
f_{11}	9.93	13.59	16.83
f_{12}	8.12	11.01	13.68
f_{13}	10.09	20.01	22.41
f_{14}	10.01	13.57	16.86
f_{15}	0.53	0.24	0.58
f_{16}	11.30	16.42	19.93
f_{17}	10.19	15.29	18.37
f_{18}	10.01	20.02	22.38
f_{19}	10.01	8.65	13.23
Sans-défaut	10.01	13.57	16.86

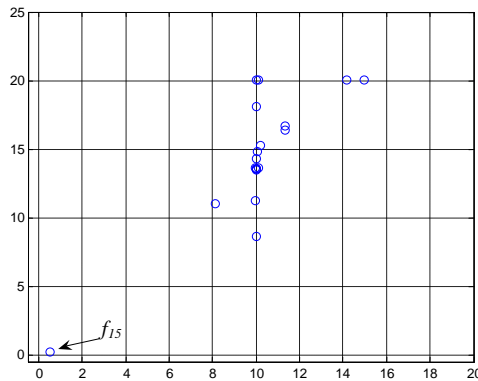


Figure IV.55: Diagnostic hors ligne: localisation du modèle de connaissance de défaut $FM(j)$ dans le plan (ρ_1, ρ_2) pour les défauts candidats f_1 à f_{19} .

Pour le diagnostic hors ligne, la figure IV.55 et le tableau IV.9 illustrent l'emplacement de chaque modèle de connaissance défectueux $FM(j)$ dans le plan (ρ_1, ρ_2) et aussi la distance $D_j(t, t, 451)$ à l'instant $t=1000s$. A partir des sorties mesurées et estimées, le modèle $FM(15)$ correspondant au défaut candidat f_{15} correspond à la plus petite distance euclidienne. On peut conclure que f_{15} est le défaut le plus probable suivant l'analyse des résidus dans l'intervalle de temps $[0, 1000s]$.

On peut conclure aussi des résultats similaires pour d'autres défauts candidats.

Un autre exemple est donné avec le défaut candidat f_{12} simulé dans l'intervalle de temps $[500s, 1000s]$. En fonction du stade de détection, le défaut est détecté à l'instant $\tau=502s$, et le groupe $G1 = \{f_3, f_6, f_9, f_{12}, f_{18}, f_{19}\}$ peut être isolé. Le diagnostic rapide en ligne est effectué dans un intervalle de temps plus petit (par exemple $T=50s$). Les figures IV.56 et IV.57 illustrent les trajectoires de la position de chaque modèle de connaissance du défaut $FM(j)$ dans le plan (ρ_1, ρ_2) pour $t \in [0, 700s]$. Chaque marque «+» correspond à la position actuelle des résidus cumulatifs $R_{1j}(t, 50, \tau)$ et $R_{2j}(t, 50, \tau)$ pour $t \in [0, 700s]$. Durant l'intervalle précédent la trajectoire de la distance euclidienne minimale à l'origine (valeur minimale de $D_j(t, 50, \tau)$) correspond au défaut le plus probable. La figure IV.57 met en évidence le défaut candidat f_{12} le plus probable.

On peut conclure aussi des résultats similaires pour d'autres défauts candidats.

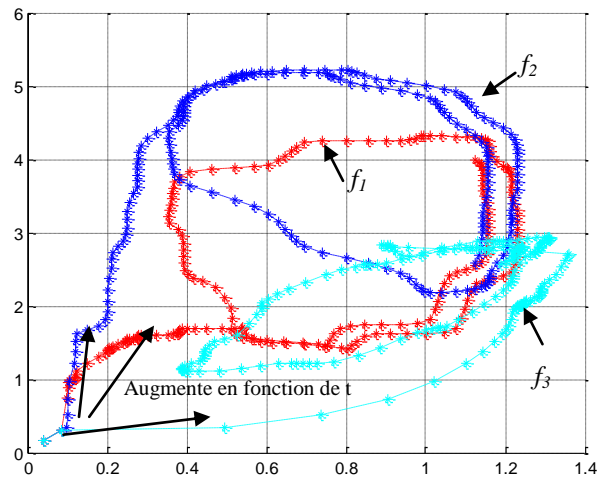


Figure IV.56: Diagnostic en ligne: localisation du modèle de connaissance de défaut $FM(j)$ dans le plan (ρ_1, ρ_2) pour les défauts candidats $\{f_1, f_2, f_3\}$.

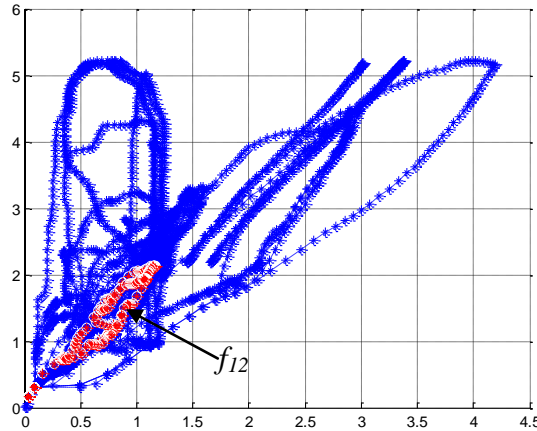


Figure IV.57: Diagnostic en ligne: localisation du modèle de connaissance de défaut $FM(j)$ dans le plan (ρ_1, ρ_2) pour les défauts candidats f_1 à f_{19} .

Suivant les trajectoires (Figure IV.57), les effets estimés des défauts sont faibles près de l'origine et augmentent au fur et à mesure qu'on s'éloigne de l'origine. La trajectoire correspondant au $FM(12)$ (c-à-d le défaut existant attendu) reste à proximité de l'origine par rapport aux autres trajectoires. On remarque également que les résidus cumulatifs $R_{1j}(t, T, \tau)$ et $R_{2j}(t, T, \tau)$ couvrent la partie positive du plan (ρ_1, ρ_2) . La répartition des résidus cumulatifs dans le plan (ρ_1, ρ_2) confirme l'importance des deux sorties $X(t)$ et $F(t)$ pour concevoir les résidus. L'évaluation des vecteurs de résidus R_0 conduit à une première étape de détection et de localisation.

Lorsque le défaut f_3 est simulé pendant l'intervalle de temps [487s 1000s], tous les modèles de comportements défectueux $NNFM(j)$ dans le groupe G1 sont évalués (Figure IV.58) et les résidus sont donnés par la Figure IV.59.

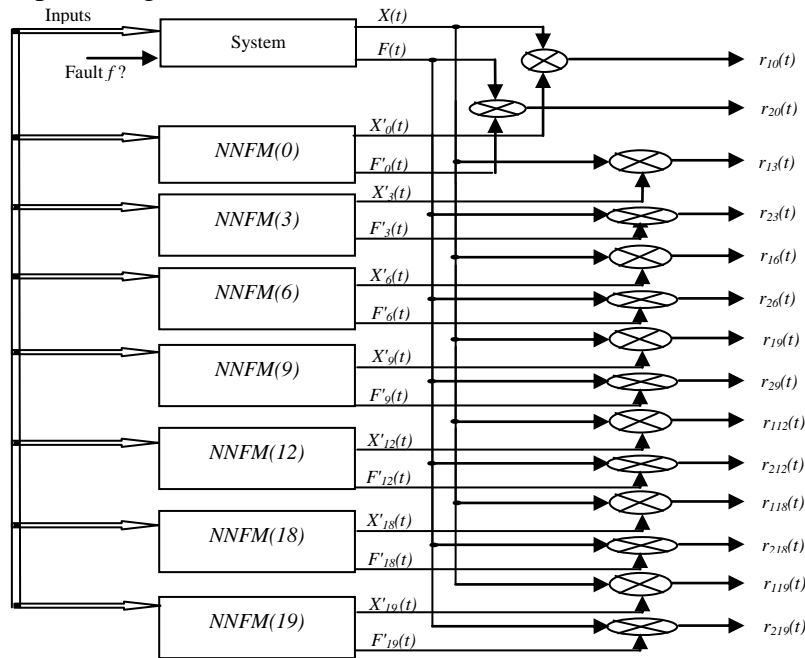


Figure IV.58: Localisation d'un défaut du groupe $G1$.

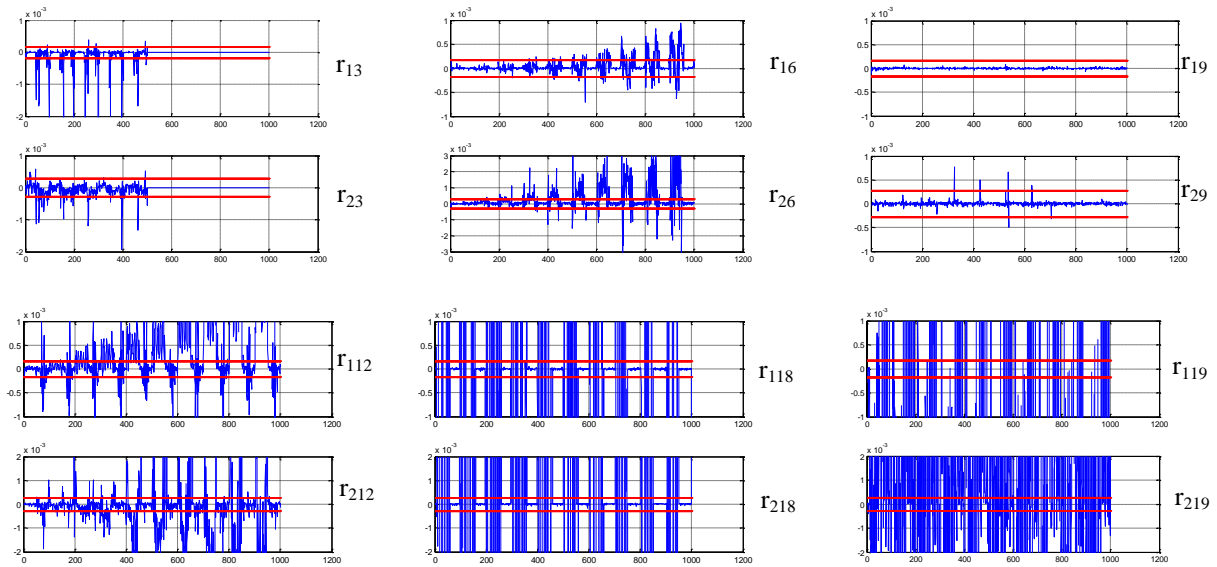


Figure IV.59: Résidus pour le groupe G1 (X-label: le temps(s), Y-label: l'amplitude des résidus) pour un défaut f_3 simulé à partir de $t=487s$.

L'application de la méthode de seuillage conduit d'habitude à une localisation partielle. Les résidus r_6 , r_{12} , r_{18} et r_{19} dépassent nettement les seuils pour $t > \tau$ ($\tau = 501s$) et donc les incidents potentiels f_6 , f_{12} , f_{18} et f_{19} sont localisés. Mais les défauts f_3 et f_9 sont difficiles à séparer.

La seconde méthode donne des meilleurs résultats. On définit les résidus cumulés $\rho_{li}(t, T, \tau)$, $\rho_{2i}(t, T, \tau)$, et la distance $D_i(t, T, \tau)$, selon les équations (IV.5) et (IV.6). L'application de la méthode décrite dans la section IV.4 conduit à des résultats donnés dans le tableau IV.10. Un retard de diagnostic avec une large fenêtre de temps ($t=1000s$) est obtenu selon l'équation (10).

Tableau IV.10: Diagnostic hors ligne pour le défaut f_3 avec $T = t = 1000s$.

Modèle de défauts $NNFM(i)$				
Défaut	$\rho_{li}(1000, 1000, 501)$	$\rho_{2i}(1000, 1000, 501)$	$D_i(1000, 1000, 501)$	$P_i(1000, 1000, 501)$
f_3	0.01	0.90	0.90	0.51
f_6	0.03	6.48	6.48	0.07
f_9	0.09	1.84	1.84	0.25
f_{12}	2.71	5.73	6.34	0.07
f_{18}	0.03	9.92	9.92	0.04
f_{19}	0.03	13.13	13.13	0.03

Les résultats du diagnostic sont présentés dans le tableau IV.10 pour $T = 1000s$. La 5^{ème} colonne du tableau IV.10 montre que la probabilité pour le défaut f_3 est d'environ 52% et le facteur de confiance pour le diagnostic est d'environ 51% selon l'équation (IV.15). On peut conclure que le défaut f_3 est le plus probable quand les résidus sont analysés dans l'intervalle de temps $[0, 1000s]$.

Un diagnostic précoce pour le défaut f_3 est également illustré par la sélection d'un petit intervalle de temps avec $T=50s$. Pour tout $t \in [0, 1000]$, le modèle avec la distance minimale à l'origine

(valeur minimale du $D_i(t, 50, \tau)$) correspond au défaut le plus probable. La Figure IV.60 (gauche) illustre les probabilités des défauts candidats à l'instant τ de détection en fonction du temps et le facteur de confiance de la décision FDI est donné par la Figure IV.61 (droite). On remarque que P_j et CF présentent une fréquence spécifique de 0,01 Hz qui correspond à la fréquence de l'entrée du système.

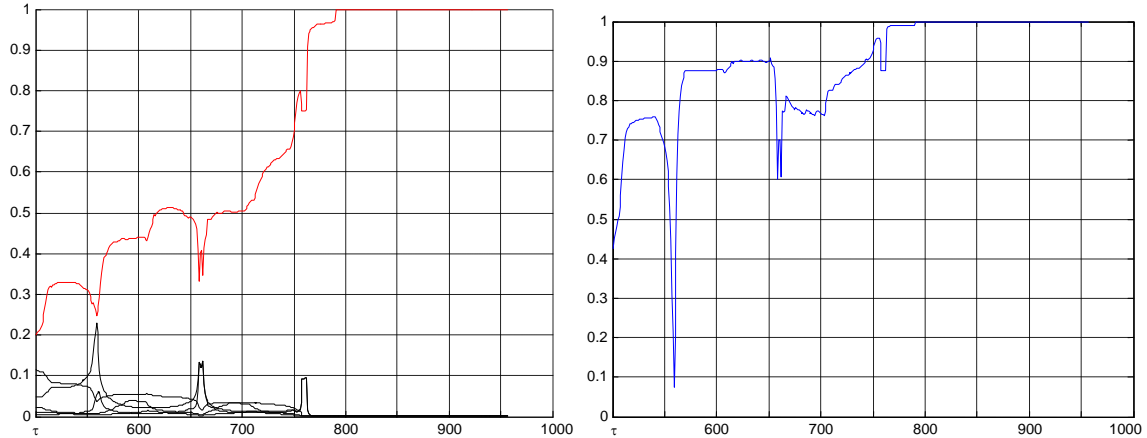
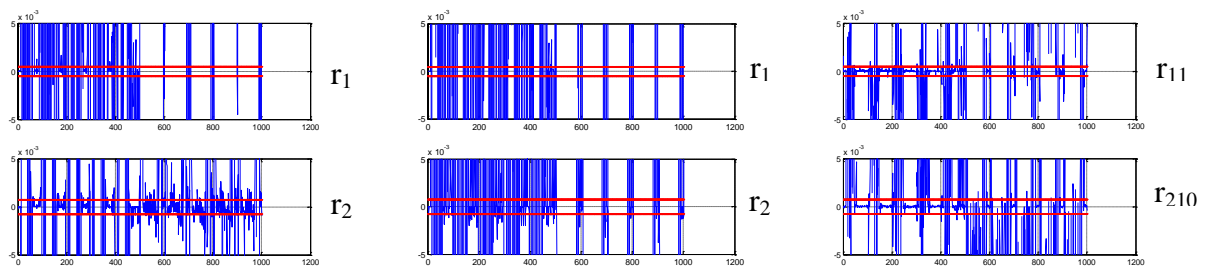


Figure IV.60: Evaluation des performances pour les FDI du défaut f_3 - (X-label: le temps(s), Y-label: la probabilité et le facteur de confiance) -Gauche) Probabilité $P_3(t, 50, 501)$; -Droit) facteur de confiance $CF(t, 50, 501)$.

Dans la figure IV.60 gauche, la courbe supérieure correspond à la probabilité du défaut f_3 . Cette probabilité augmente avec le temps et atteint la valeur 1 au temps $t = \tau + 290 = 791s$. Elle varie rapidement au cours de la phase de décision [500s 550s]. Ceci illustre la robustesse de notre méthode. La figure IV.60 droite montre les variations du facteur de confiance calculées par l'équation (IV.12) et confirme que le défaut f_3 est le défaut la plus probable.

Le défaut f_{15} est également simulé dans l'intervalle de temps [451s 1000s]. Ce défaut est détecté à l'instant $t=458s$ et les défauts du groupe 2 peuvent être localisés. Puis tous les modèles neuronaux des comportements défectueux dans le groupe G2 sont évalués et les résidus obtenus sont donnés par la figure IV.61.



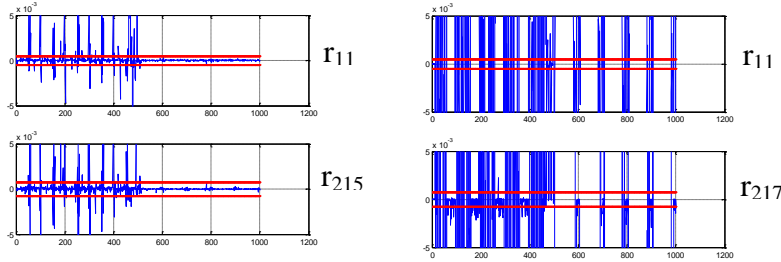


Figure IV.61: Résidus du groupe G2 (X-label: le temps (s), Y-label: l'amplitude des résidus) lorsque f_{15} simulé à partir de $t = 451$ s.

Notre méthode FDI est également appliquée pour localiser f_{15} . L'application de la méthode décrite dans la section IV.4 conduit aux résultats du tableau IV.11.

Tableau IV.11: Diagnostic hors ligne pour le défaut f_{15} avec $T = t = 1000$ s.

Modèle de défauts $NNFM(i)$				
défaut Candidat	$\rho_{i1}(1000, 1000, 458)$	$\rho_{i2}(1000, 1000, 458)$	$D_i(1000, 1000, 458)$	$P_i(1000, 1000, 458)$
f_1	8.93	4.09	9.82	0.007
f_7	10.38	12.52	16.27	0.004
f_{10}	1.23	1.44	1.90	0.037
f_{15}	0.06	0.04	0.07	0.935
f_{17}	1.69	4.50	4.81	0.015

Le tableau IV.11 indique l'emplacement de chaque modèle $NNFM(j)$ dans le plan (ρ_1, ρ_2) et la distance $D_j(t, t, 458)$ à l'instant $t = 1000$ s. La 5^{ème} colonne du tableau IV.11 indique également les probabilités de chaque défaut candidat selon l'équation (IV.12). A partir de cette colonne, on peut conclure que le défaut f_{15} est le plus probable. La probabilité du défaut f_{15} est d'environ 96%. De même, les probabilités de défauts des autres ne dépassent pas 3%. Ces indicateurs fournissent un facteur de confiance d'environ 96% conformément à l'équation (IV.12).

Le diagnostic précoce du défaut f_{15} est illustré par la sélection d'un petit intervalle de temps $T=50$ s. Pour tout $t \in [0, 1000]$ s, le modèle avec la distance minimale à l'origine correspond au défaut le plus probable. Les trajectoires de la figure IV.62 (gauche), représentent les modèles $NNFM$ en fonction du temps. La Figure IV.62 (droit) est un détail pour mieux voir la trajectoire pour le modèle $NNFM(15)$.

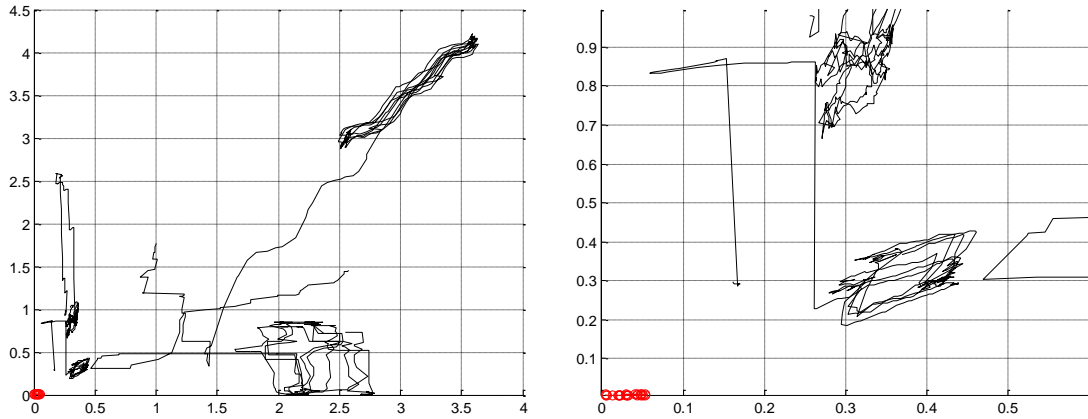


Figure IV.62: Diagnostic précoce dans le cas du défaut f_{15} (X-label: ρ_1 , Y-label: ρ_2): Gauche) l'emplacement des modèles $NNFM(j)$; à droite) Détails.

La trajectoire correspondant à $NNFM(15)$ reste à proximité de l'origine par rapport aux autres trajectoires. On peut conclure que le défaut candidat f_{15} est le défaut le plus probable. La répartition des résidus cumulés dans le plan (ρ_1, ρ_2) confirme l'importance de deux sorties $X(t)$ et $F(t)$ pour concevoir les résidus (on peut remarquer que les résidus cumulés $\rho_{1i}(t, T, \tau)$ et $\rho_{2i}(t, T, \tau)$ couvrent la partie positive du plan (ρ_1, ρ_2)). La Figure IV.63 expose les probabilités des défauts candidats à l'instant de détection τ en fonction du temps et aussi le facteur de confiance de décision FDI.

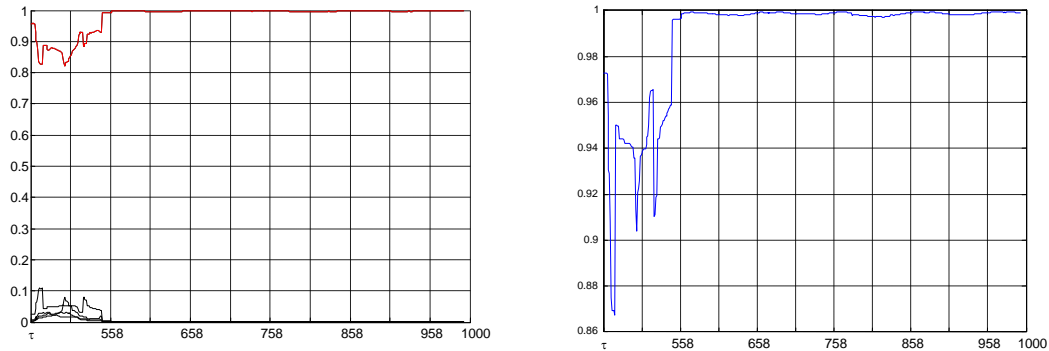


Figure IV.63: Evaluation des performances pour FDI du défaut f_{15} (X-label: le temps (s), Y-label probabilité et facteur de confiance).-Gauche) Probabilité $P_{15}(t, 50, 458)$; -Droite) facteur de confiance $CF(t, 50, 458)$.

Sur la Figure IV.63 (gauche), la courbe supérieure correspond à la probabilité du défaut f_{15} . Cette probabilité augmente très rapidement et atteint la valeur 1 au temps $t = \tau + 100 = 558$ s. La figure IV.63 (droite) montre les variations du facteur de confiance calculé par l'équation (15) et confirme que le défaut f_{15} est le défaut le plus probable. On remarque que le facteur de confiance pour localiser le défaut f_{15} atteint rapidement la valeur 1 en comparaison avec le défaut f_3 : la raison est que f_{15} est un défaut brutal alors f_3 est un défaut de type naissant.

Le défaut f_3 est aussi simulé pendant l'intervalle de temps [302s 1000s]. Ce défaut ne peut pas être détecté avec la technique de seuillage. Les résidus obtenus sont illustrés dans la figure IV.64 et on remarque qu'aucun résidu du groupe G3 ne dépasse les seuils définis précédemment.

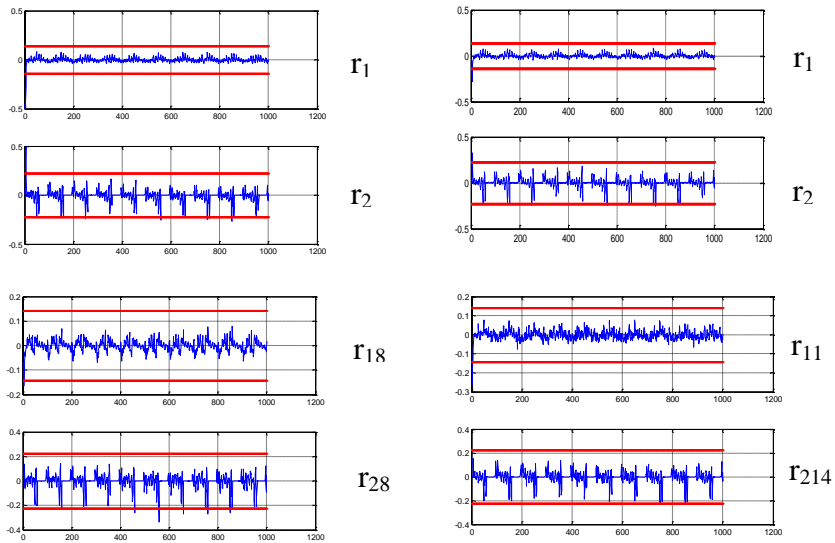


Figure IV.64: Résidus du groupe $G3$ (X-label : time (s), Y-label : Amplitude des résidus) où f_5 est simulé entre $t = 302s$ à $1000s$.

Dans ce cas, la détection et la localisation sont obtenues en une seule étape en considérant simultanément tous les résidus des modèles dans le groupe 3 (c.-à-d. R_0, R_5, R_8 et R_{14}). Les probabilités des modèles $NNFM(0)$, $NNFM(5)$, $NNFM(8)$ et $NNFM(14)$ sont présentés sur la Figure IV.65 (gauche). Dans cette figure, on peut remarquer que la probabilité du modèle $NNFM(0)$ est nettement la plus grande durant les instants $t = 100s$ à $t = 300s$ (ligne grise), puis à partir des instants $t = 300s$ à $t = 500s$, les probabilités de tous les modèles sont très similaires et enfin la probabilité du modèle $NNFM(5)$ augmente à partir de l'instant $t = 500s$ (ligne noire). Le facteur de confiance dans la Figure IV.65 (droite) montre que les décisions prévues par le système FDI sont fiables dans les intervalles de temps $[100s\ 300s]$ et $[500s\ 1000s]$.

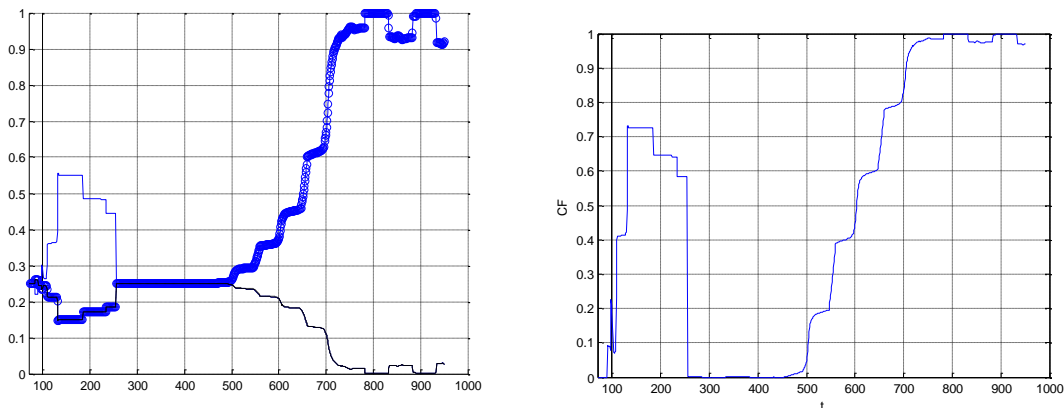


Figure IV.65: Evaluation des performances pour les FDI du défaut f_5 (X-label: le temps(s), Y-label: la probabilité et le facteur de confiance) -Gauche) Probabilité P_5 ; right) -Droit) Facteur de confiance CF .

IV.6.3 Structuration et discussion:

Le Tableau IV.12 résume les conclusions concernant la détection et le diagnostic des défauts pour le benchmark DAMADICS et selon la méthode considérée. Les résultats sont détaillés (1) pour la détection de défauts avec des seuils (d'après l'évaluation des résidus R_0), (2) pour la localisation des défauts par l'utilisation des seuils (d'après l'évaluation des résidus R_0 à R_{19}), (3) pour la détection de défauts et l'isolation avec une probabilité et le calcul des facteurs de confiance (d'après l'évaluation des résidus R_0 à R_{19}). 84% des défauts sont détectés par la méthode de seuillage. Le retard à la détection ne dépasse pas 30s. Mais les défauts dans le groupe 3 ne sont pas détectables avec la technique du seuillage.

La diminution des seuils de détection améliore les résultats de détection, mais conduit également à des fausses alarmes et le défaut f_{14} reste indétectable. Certains défauts sont détectés, mais ne peut pas être localisés avec la technique du seuillage (par exemple f_3 et f_9): la méthode permet de localiser 63% des défauts. En comparaison, notre méthode basée sur le calcul des probabilités défauts et le facteur de confiance conduit à la détection et la localisation de tous les défauts (pour l'exemple considéré). Dans quelques cas, le facteur de confiance est proche de 0,5 et la décision n'est pas considérée comme fiable. L'effort de calcul avec la méthode proposée consiste à exécuter plusieurs (jusqu'à 6) des modèles en parallèle.

Tableau IV.12: Comparaison des méthodes FDI (+: la décision est correcte, -: la décision est incorrecte).

Défauts Technique	Détection avec un seuil	localisation avec un seuil	FDI avec le facteur de confiance
f_1	+	-	+
f_2	+	+	+
f_3	+	-	+
f_4	+	+	+
f_5	-	-	+
f_6	+	-	+
f_7	+	+	+
f_8	-	-	+
f_9	+	-	+
f_{10}	+	+	+
f_{11}	+	+	+
f_{12}	+	+	+
f_{13}	+	+	+
f_{14}	-	-	+
f_{15}	+	+	+
f_{16}	+	+	+
f_{17}	+	+	+
f_{18}	+	+	+
f_{19}	+	+	+

IV.6.4 Résultat de la réduction de l'ensemble des résidus:

La méthode proposée nécessite la synthèse de 40 signaux de résidus obtenus à partir de 20 modèles sains et de dysfonctionnement. On a proposé une technique de réduction de l'ensemble des résidus par projection de ces résidus sur les axes comme nous avons vu précédemment dans le paragraphe IV.5.2. Pour illustrer cette technique nous l'avons appliqué sur les 40 résidus de notre système de diagnostic et nous avons obtenus 20 résidus (figures IV.66, IV.67 et IV.68) pour les 3 groupes de défauts $G1$, $G2$ et $G3$.

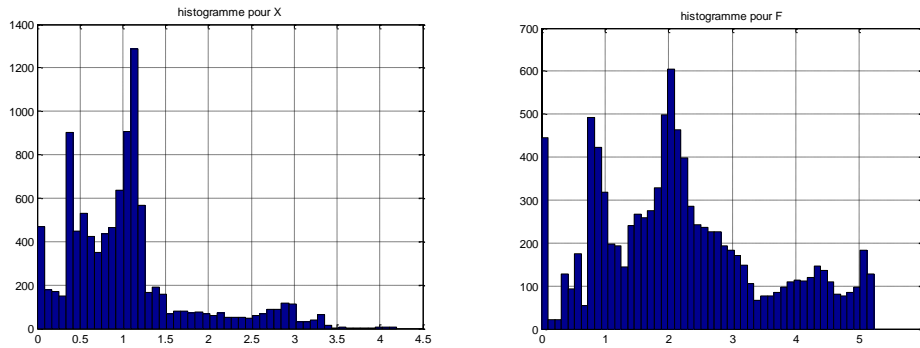


Figure IV.66: Histogramme pour la projection des résidus des modèles des comportements sain et défectueux ($NNFM(j)$) sur les deux axes X et Y dans le cas du défaut f_{12} du groupe $G1$.

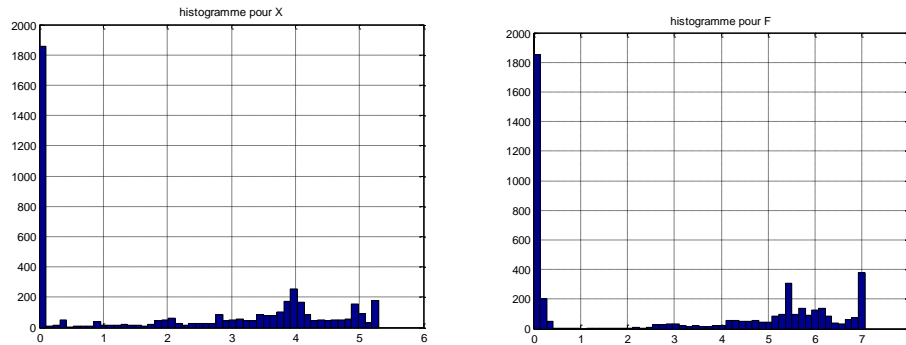


Figure IV.67: Histogramme pour la projection des résidus des modèles des comportements sain et défectueux ($NNFM(j)$) sur les deux axes X et Y dans le cas du défaut f_{15} du groupe $G2$.

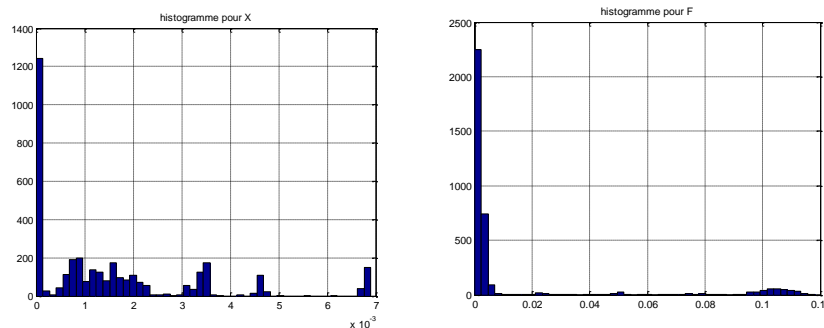


Figure IV.68: Histogramme pour la projection des résidus des modèles des comportements sain et défectueux ($NNFM(j)$) sur les deux axes X et Y dans le cas du défaut f_8 du groupe $G3$.

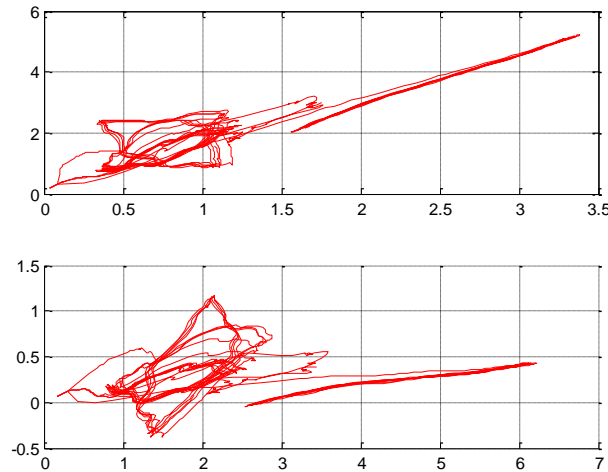


Figure IV.69: Résidus des modèles des comportements sain et défectueux ($NNFM(12)$) du groupe G2 (X-label: ρ_1 , Y-label: ρ_2) (haut). Résidus de $NNFM(12)$ après rotation des axes du repère (bas).

IV.7 Conclusion

Nous avons présentés un actionneur électropneumatique. Puis, un modèle neuronal permettant de reproduire le bon fonctionnement de cet actionneur a été présenté. Ensuite, en utilisant les méthodes développées dans les deux premiers chapitres, nous avons fait la génération des résidus pour la classification des défauts (f_7 , f_{10} , f_{15} et f_{17}).

Dans la première étape, le modèle neuronal donne des résultats satisfaisants, ce dernier a permis de reproduire les sorties de l'actionneur, la détection a donnée de bons résultats ainsi que le diagnostic avec les trois classifieur (neuronal, à base de logique floue, et neuro-flou). Notre système de diagnostic est capable de détecter et localiser les défauts f_7 , f_{10} , f_{15} et f_{17} .

Dans ce chapitre nous avons proposé des méthodes différentes pour la détection et le diagnostic des défauts : l'utilisation des réseaux de neurones, de la logique floue et le système neuro-flou. Cette approche nous a permis d'identifier les défauts du système DAMADICS. D'après les résultats de simulation de modélisation nous avons trouvé que le système neuro-flou donne des erreurs plus petites par rapport aux réseaux de neurone et à la logique floue. Dans l'étape de diagnostic en ligne nous avons utilisé les réseaux des neurones pour créer les modèles de comportements défectueux de l'actionneur DAMADICS. Cet algorithme de diagnostic permet de détecter, localiser et identifier les 19 défauts existant dans cet actionneur d'une manière hors lignes ou en ligne. Les résultats obtenus par ces trois méthodes on été jugés satisfaisants comparativement à ce qui a été trouvé jusqu'à nos jours par validation avec les travaux existant dans ce domaine, surtout les travaux de [Kor, 04], [Pat, 00].

La méthode proposée dans ce travail pour la détection et le diagnostic des défauts en ligne, combine la puissance de calcul et la robustesse des réseaux de neurones à la décision simple en temps réel en fonction de la distance euclidienne des résidus cumulés. La méthode conduit à la

détection et à la localisation des défauts. Les résultats obtenus avec le benchmark DAMADICS illustrent les performances de notre contribution.

La limitation de la méthode proposée est la nécessité de concevoir des modèles des comportements défectueux en fonction de chaque défaut candidat. La conception de ces modèles nécessite des ressources de calcul et l'historique des données du système.

Comme perspective, on étudiera la couverture de l'espace des résidus par les résidus afin de sélectionner les signaux les plus appropriés. On peut remarquer que certains résidus contiennent des informations utiles pour le FDI et les autres sont tout à fait inutiles. Basé sur l'évaluation d'un facteur de confiance pour chaque résidu, nous allons étudier une méthode pour sélectionner les résidus les plus fiables. Nos futurs travaux sont également de valider cette technique en l'appliquant sur d'autres systèmes avec différentes conditions de fonctionnement et des défauts différents.

CONCLUSION GENERALE ET PERSPECTIVES

Ce modeste travail nous a permis d'appliquer des méthodes de diagnostic utilisant l'intelligence artificielle sur un actionneur électropneumatique. Les réseaux de neurones constituent un excellent outil de modélisation et de diagnostic des processus non linéaires, malgré les problèmes que nous avons rencontrés qui se situent dans le choix d'une architecture optimale en fonction des entrées, ainsi que le nombre de couches, et le nombre de neurones dans chaque couche. Le choix d'un algorithme efficace pour assurer la convergence vers le minimum, et le choix des poids et biais initiaux, ont une grande influence sur la convergence de l'algorithme. Nous avons surtout montré que ce modèle permettait:

- De détecter un défaut et cela à partir de la comparaison entre la sortie du système réel et son estimation.
- De localiser un défaut et pour cela il suffit d'identifier le type de défaut.

Les modèles générés nous donnent des bons résultats prouvant ainsi leurs efficacités dans le domaine de la représentation des comportements sains et défectueux. Après avoir obtenu un modèle neuronale, nous avons identifié les défauts par trois types de classifieurs, le premier à base de réseaux de neurone, le second à base de logique floue et le troisième à base de neuro-flou dans le but de montrer leur efficacité dans cette étape qui est très importante pour le diagnostic. Pour diagnostiquer tous les défauts qui existent dans le système dynamique DAMADICS, on a développé une technique capable de détecter et d'isoler tous ces défauts d'une manière rapide en ligne en se basant sur la régénération automatique des résidus.

Notre contribution réside dans le développement d'une technique FDI qui utilise les modèles de comportements sain et défectueux combiné avec la méthode de seuillage pour isoler les défauts. Nous avons aussi proposé une nouvelle méthode qui évalue la probabilité de défauts et la confiance accordée à la décision de diagnostic. Les deux techniques donnent de bonnes décisions dans de nombreux cas. Cependant, les résultats obtenus avec la méthode basée sur le calcul des probabilités sont de meilleure qualité et la fiabilité de la décision est aussi explicitement évaluée. En particulier, la méthode proposée ne nécessite pas de calculer les seuils pour la détection et la localisation et par conséquent est plus robuste pour les défauts de types naissants.

De notre point de vue, la principale limitation de la méthode proposée est l'augmentation rapide de l'effort de calcul lorsque des nombreux défauts candidats et de nombreuses sorties sont pris en compte. Afin de réduire cet effort, on peut remarquer que certains résidus contiennent des informations utiles pour le FDI et les autres sont tout à fait inutiles. Sur la base de l'évaluation d'un facteur de confiance pour chaque résidu, nous allons étudier une méthode pour sélectionner les résidus les plus fiables. Nos futurs travaux prévoient également de valider cette technique en l'appliquant sur d'autres systèmes réels avec différentes conditions de fonctionnement et des défauts différents.



WEBSITE:

[1] <http://www.sztaki.hu/conferences/safeprocess2000/termfr.html>

[2] <http://www.sztaki.hu/conferences/safeprocess2000/index.html>

[3] <http://monet.aber.ac.uk:8080/monet/index.html>

[4] <http://homepages.laas.fr/combacau/SPSF/sursup.html>

[5] <http://www.batch-a.com/S8801.htm>

[6] <http://sites.polytech.univ-mrs.fr/flavien.peysson/docs/ens/c942.pdf>

[7] <http://perso.ensem.inpl-nancy.fr/Didier.Maquin/ps/Surveillance.pdf>

[8] DAMADICS, 2002, Website of DAMADICS: Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems. <http://diag.mchtr.pw.edu.pl/damadics/>.
<http://www.eng.hull.ac.uk/research/control/damadics1.htm>

- [Adj, 93] Adjallah K.H., (1993) "*Contribution au diagnostic de systèmes par observateurs d'état*", Thèse de doctorat de l'Institut National Polytechnique de Lorraine, France.
- [Akh, 96] Akhmetov D. and Dote Y. (1996) "Novel "on-line" Identification Procedure using Artificial Neural Network", In IFAC Workshop, *On-line Fault Detection and Supervision in the Chemical Process Industries*, Newcastle, U.K. 12-13 June 1995.
- [Ale, 98] Alessandri A. & Parisini T., (1998) "Direct Model-Based Fault Diagnosis Using Neural Filters", In IFAC Symposium, *Fault Detection, Supervision and Safety for Technical Processes*, Kingston Upon Hull, UK, 26-28 August 1997, pp. 343-348.
- [ANSI/ISA-S88.01, (1995)] Batch Control Part 1: *Models and Terminology*, Instrument Society of America, Research triangle Park, USA.
- [Åst, 00] Åström K. J., Albertos P., Blanke M., Isidori A., Santz R. & Schaufelberger W., (2000) "*Control of Complex Systems*", Springer Verlag.
- [Bar, 02] Bartyś M., (2002) "Specification Of Actuators Intended To Use For Benchmark Definition", For Internal Use Only, Manuscript V. 1.02, April.
- [Bar, 06] Bartyś M, Patton R., Syfert M., Heras S. de las, & Quevedo J., (2006) "Introduction to the DAMADICS actuator FDI benchmark study", *Control Engineering Practice*, vol. 14, no. 6, pp. 577–596.
- [Bas, 93] Basseville, M. & Nikiforov I.V. (1993) "*Detection of abrupt changes - Theory and application*". Information and System Sciences Series. Prentice Hall, Englewood Cliffs. New Jersey, USA.
- [Bea, 71] Beard R. V., (1971) "*Failure accommodation in linear systems through self-reorganisation*". Rept. MTV-71-1 Man Vehicle Laboratory, MIT, Cambridge 199, 1_20.
- [Beb, 07] Bebiolles A., (2007) "*Diagnostic de systèmes complexes à base de modèle interne, reconnaissance des formes et fusion d'informations*", thèse de doctorat ; Université de Technologie de Compiègne.
- [Ben, 98] Benkhedda H. & Patton R. (1998) "Information Fusion in Fault Diagnosis Based on BSpline Network", In IFAC Symposium, *Fault Detection, Supervision and Safety for Technical Processes*, Kingston Upon Hull, UK, 26-28 August 1997, pp. 669-674.
- [Ben, 04] Bentoumi M., (2004) "*Outils pour la détection et la classification. Application au diagnostic de défauts de surface de rail*" thèse de Doctorat de l'université Henri Poincaré, Nancy 1.
- [Bla, 03] Blanke M., Kinnaert M., Lunze J., & Staroswiecki M., (2003) "*Diagnosis and Fault Tolerant Control*", Springer, New York, NY, USA.
- [Bor, 90] Borne P., Dauphin-Tanguy G., Richard J.P., Rotella F. & Zambettakis I. (1990) "*Automatique: commande et optimisation des processus*", Editions Technip, Méthodes et Techniques de l'Ingénieur.
- [Bou, 04] Bouthiba T., (2004) "Fault location in ehv transmission lines using artificial neural networks", *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 1, pp. 69–78.
- [Car, 96] Carpentier T. & Litwak R., (1996) "Une approche structurelle pour le positionnement de capteurs en vue de la surveillance", AIG'96, *Automatique, Génie Informatique Image*, Tours, pp. 103-107,
- [Cha, 93] Chatain J.N., (1993) "*Diagnostic par système expert*". *Traité des Nouvelles Technologies, série Diagnostic et Maintenance*, Hermès, Paris.
- [Che, 98] Chen B., Shang Z., Yang S., Wang X. & McGreavy C. (1998) "Operational Support System for On-line Fault Monitoring in Chemical Manufacturing, In IFAC Symposium, *Fault Detection, Supervision and Safety for Technical Processes*, Kingston Upon Hull, UK, 26-28 August 1997, pp. 919-924.

-
- [Che, 99] Chen J. & Patton R., (1999) "*Robust Model-Based Fault Diagnosis for Dynamic Systems*". London : Kluwer Academic Publishers.
- [Che, 05] Chen W. & Saif M., (2005) "An Actuator Fault Isolation Strategy for Linear and Nonlinear Systems", Proceedings of the American Control Conference ACC'05, June 8-10, Portland, OR, USA.
- [Che, 90] Cheung J. T & Stephanopoulos G. (1990) "Representation of Process Trends", *Computers Chemical Engineering* 14, p.I and II, pp. 495-540.
- [Cho, 80] Chow E. Y. (1980) "*Failure detection system design methodology*", Thesis, Lab. Information and Decision system, M.I.T, Cambridge.
- [Cho, 84] Chow E. Y. & Willsky A. S., (1984) "Analytical redundancy and the design of robust failure detection systems", *IEEE Trans. Automatic Control*, Vol. 29, n°7, pp. 603-614, July 1984.
- [Cla, 75] Clark R.N., (1975) "Instrument fault detection", *IEEE Trans., Aerospace and Electronic System*, EAS-14, pp. 558-563.
- [Cla, 89] Clark R.N., Patton R.J. & Frank P.M., (1989) "*Fault diagnosis in dynamic systems : theory and application*". International Series in Systems and Control Engineering. Prentice Hall, Englewood Cliffs. New Jersey, USA.
- [Col, 02] Colomer J., Meléndez J. & Gamero F. I., (2002) "Pattern recognition based on episodes and DTW. Application to diagnosis of a level control systems". 16th International Workshop on Qualitative Reasoning (QR'2002), Barcelone, Espagne.
- [Cou, 97] Courtine S. (1997) "*Détection et localisation de défauts dans les entraînements électriques*", Thèse de l'Institut Polytechnique de Grenoble.
- [Das, 03] Dash S., Rengaswamy R. & Venkatasubramanian V., (2003) "Fuzzylogic based trend classification for fault diagnosis of chemical processes", *Computers and Chemical Engineering*, 27, pp.347-362.
- [Del, 01] Delaleau E., Louis J. P., & Ortega R., (2001) "Modeling and control of induction motors", *International Journal of Applied Mathematics and Computer Science*, vol. 11, no. 1, pp. 105-129.
- [Din, 91] Ding X. & Frank P.M., (1991) "Frequency domain approach and threshold selector for robust model-base fault detection and isolation", Proc. of IFAC Symp. SAFEPROCESS'91, Baden, Germany, vol. 1, pp.307-312.
- [Dre, 98] Dreyfus G., (1998) "*Les réseaux de neurones*", Mécanique Industrielle et Matériaux, n°51.
- [Dun, 96] Dunia R., Qin S., & Edgar T. F., (1996) "Identification of faulty sensors using principal component analysis". *American Institute Of Chemical Engineers Journal*, 42(10), 2797-2812.
- [Edw, 94] Edwards C. & Spurgeon S. (1994) "On the development of a discontinuous observer", *International Journal of Control*, 59, pp. 1211-1229.
- [Ema, 88] Emami-Naeini A. E. A., (1988) "Effect of model uncertainty of failure detection: The threshold selector", *IEEE Transaction on Automatic Control*, AC-33, pp. 1106-1115.
- [Fos, 93] Foss B.A., & Johansen, T.A (1993) "On local and fuzzy modelling". In *3rd Int. Conf. on Industrial Fuzzy Control and Intelligent Systems*, Houston, Texas.
- [Fra, 87] Frank P.M., (1987) "Fault diagnosis in dynamic systems via state estimation" - a survey, In Tzafestas S., Singh M., Schmidt G. (Eds.) *System Fault Diagnostics, Reliability and Related Knowledge-Based Approaches*, Vol.1, pp. 35-98. Reidel, Dordrecht.
- [Fra, 89] Frank P.M. & Wünnenberg J., (1989) "*Robust diagnosis using unknown input observer schemes, for Fault diagnosis in dynamic systems: theory and application*", Patton R.J, Frank P.M., Clark R.N., Prentice Hall, Englewood, Cliffs, pp. 47-98.
-

- [Fra, 90] Frank P. M., (1990) "Fault diagnosis in dynamic systems using analytical and knowledgebased redundancy : A survey and some new results". *Automatica* 26 (3), pp. 459–474.
- [Fra, 96] Frank P. M., (1996) "Analytical and Qualitative Model-based Fault Diagnosis - A Survey and Some New Results", *European Journal of Control*, 2(1), pp. 6–28.
- [Fra, 00] Frank P. M., Alcorta Garcia E., & Koppen-Seliger B., (2000) "Modelling for fault detection and isolation versus modelling for control". *Mathematics and Computers in Simulation*, 53(4-6):259–271.
- [Fuj, 96] Fujiwara T., Tsushi T. & Nishitani H., (1996) "Failure Detection by Auto-Associative Neural Networks", In IFAC Workshop, *On-line Fault Detection and Supervision in the Chemical Process Industries*, Newcastle, 12-13 June 1995, pp. 33-38.
- [Gar, 97] Garcia E. A. & Frank P. M., (1997) "Deterministic Nonlinear Observer-Based Approaches to Fault Diagnosis: A Survey", *Control Engineering Practice*, 5(5), pp. 663–760.
- [Ger, 91] Gertler J. (1991) "Analytical Redundancy Methods in Failure Detection and Isolation". SAFEPROCESS 91, Baden-Baden (Germany), pp. 9–21.
- [Ger, 97] Gertler J. & McAvoy T., (1997) "Principal component analysis and parity relations - a strong duality". IFAC conference SAFEPROCESS, Hull, UK. 837-842.
- [Ger, 98] Gertler J. (1998) "*Fault Detection and Diagnosis in Engineering Systems*". Marcel Dekker, New York.
- [Gom, 96] Gomm J. (1996) "Process Fault diagnosis using a Self-Adaptive Neural Network with On-Line Learning Capabilities", In IFAC Workshop, *On-line Fault Detection and Supervision in the Chemical Process Industries*, Newcastle, 12-13 June 1995.
- [Gom, 99] Gomm J. B., Yu D. L., & Williams D., (1999) "Sensor fault diagnosis in a chemical process via RBF neural networks", *Control Engineering Practice*, vol. 7, no. 1, pp. 49–55.
- [Gup, 03] Gupta M. M., (2003) "*Static and Dynamic Neural Networks*", JohnWiley & Sons, Hoboken, NJ, USA.
- [Ham, 99] Hammouri H., Kinnaert M. & Yaagoubi E. H., (1999) "Observer-Based Approach to Fault Detection and Isolation for Nonlinear Systems", *IEEE Transaction on Automatic Control*, 44(10), pp. 1879–1884.
- [Heb, 49] Hebb D. O., (1949) "*The Organization of Behavior: A Neuropsychological Theory*", Wiley, New York.
- [Hen, 00] Henry M., (2000) "Plant asset management via intelligent sensors digital, distributed and for free", *Computing and Control Engineering Journal*, vol. 11, no. 5, pp. 211–213, 2000.
- [Hes, 96] Hessel G., Schmitt W. & Weiss F.-P., (1996) "A Neural Network Approach for Acoustic Leak Monitoring at Pressurized Plants with Complicated Topology", In IFAC Workshop, *On-line Fault Detection and Supervision in the Chemical Process Industries*, Newcastle, 12-13 June 1995, pp. 83-88.
- [Hja, 95] Hjalmarsson H., Juditsky A., Sjöberg J., Zhang Q., Ljung L., Benveniste A. Belyon B. & Glorennec P.-Y., (1995) "Nonlinear blackbox modeling in system identification: a unified overview", *Automatica*, vol. 31, no. 12, pp. 1691–1724.
- [Hop, 80] Hopfield J., (1980) "Artificial neural Networks" *JEEE circuits and Devices Magazine* Sep, pp. 4-10.
- [Hop, 82] Hopfield J., (1982) "Neural networks and physical systems with emergent collective computational abilities". *Proceedings of the National Academy of Sciences, USA*, 79:2554-2558.

- [Hop, 84] Hopfield J., (1984) "Neurons with graded response have collective computational properties like those of two-state neurons". *Proceedings of the National Academy of Sciences*, USA, 81:3088–3092.
- [Hor, 89] Hornik K., Stinchcombe M., White H., (1989) "Multilayer feed-forward networks are universal approximators". *Neural Networks* 2: 359-66.
- [Hos, 88] Hoskins, J. C., & Himmelblau D. M., (1988) "Artificial Neural Network Models of Knowledge Representation in Chemical Engineering", *Comp. Chem. Eng.*, 12(9/10), 881.
- [Ise, 84] Isermann R., (1984) "Process fault detection based on modelling and estimation methods", *Automatica*, 20(4), 387-404.
- [Ise, 94] Isermann R., (1994) "On the applicability of Model-based fault detection for technical processes", *Control Engineering Practice*, 2, pp. 439-450.
- [Ise, 96] Isermann R. & Ballé P., (1996) "Trends in the application of model-based fault detection and diagnosis of technical processes". Proc. Of the IFAC, 13th Triennial World Congress, San Francisco, USA, vol. 7, pp. 1-12.
- [Ise, 97a] Isermann R., (1997) "Supervision, fault-detection and fault-diagnosis methods – Advanced methods and applications". Proc. Of the *IMEKO* world congress, New Measurements – Challenges and Visions, Tampere, Finland, vol. 1 pp 1-28.
- [Ise, 97b] Isermann, R. & Balle P., (1997) "Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes", *Control Engineering Practice*, 5(5), pp. 709-719.
- [Ise, 02] Isermann R., Schwarz R. & Stolzl S., (2002). Fault-tolerant drive-by-wire systems, *IEEE Control Systems Magazine*, 22(5), pp. 64-81.
- [Ise, 05] Isermann R., (2005) "Model-based fault-detection and diagnosis -status and applications", *Annual Reviews in Control*, vol. 29, no. 1, pp. 71–85.
- [Ise, 06] Isermann R., (2006) "*Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance*", Springer, Berlin, Germany.
- [Jan, 91] Janusz M. & Venkatasubramanian V., (1991) "Automatic generation of qualitative description of process trends for fault detection and diagnosis", *Engineering Applications of Artificial Intelligence* 4, pp. 329-339.
- [Jan, 97] Jang, J. S. R., C. T. Sun and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing*. Englewood Cliffs, NJ: Prentice-Hall.
- [Joh, 93] Johansen, T.A & Foss B. A., (1993) "Constructing NARMAX models using ARMAX models", *International Journal of Control*, Vol.58, pp.1125-1153.
- [Jon, 73] Jones H. L., (1973) "*Failure detection in linear systems*". PhD thesis. Massachusetts Institute of Technology. Department of Aeronautics and Astronautics, Cambridge.
- [Kas, 98] Kassidas A., MacGregor J.F. & Taylor P. A., (1998) "Synchronisation of Batch Trajectories Using Dynamic Time Warping", *AIChE Journal*, vol. 44, n. 4, pp. 864-875.
- [Kor, 04] Korbicz J., Koscielny J. M., Kowalczyk Z., & Cholewa W., (2004) "*Fault Diagnosis. Models, Artificial Intelligence, Applications*", Springer, Berlin, Germany.
- [Kos, 06] Koscielny J. M., Bartys M., Rzepiejewski P., & Costa J. Sa da, (2006) "Actuator fault distinguishability study for the DAMADICS benchmark problem", *Control Engineering Practice*, vol. 14, no. 6, pp. 645–652.
- [Kou, 08] Kourd Y., Guersi N., & Lefebvre D., (2008) "A two stages diagnosis method with neuronal networks", in *Proceedings of the International Conference on Electrical Engineering Design and Technologies (ICEEDT '08)*, Hammamet, Tunisie.
- [Kou, 10a] Kourd Y., Guersi N., & Lefebvre D., (2010) "A two stages diagnosis method with Neuro-fuzzy approach", in *Proceedings of the 6th Conférence Internationale Francophone d'Automatique*, Nancy, France.

- [Kou, 10b] Kourd Y., Guersi N., & Lefebvre D., (2010) "Neuro-fuzzy approach for fault diagnosis: application to the DAMADICS", in *Proceedings of the 7th international Conference on Informatics in Control, Automation and Robotics (ICINCO '10)*, Funchal, Madeira, Portugal.
- [Kou, 11] Kourd, Y., D. Lefebvre and N. Guersi (2011). "Early FDI Based on Residuals Design According to the Analysis of Models of Faults: Application to DAMADICS". *Advances in Artificial Neural Systems*. doi:10, 1155/2011/453169.
- [Kra, 90] Kramer M. A. & Leonard J. A., (1990) "Diagnosis using back-propagation neural networks: analysis and criticism", *Computers chem. Engng 14*, 1323-1338.
- [Lef, 10] Lefebvre D., Chafouk H., & Lebbal M., (2010) "*Modélisation et Diagnostic des Systèmes. Une Approche Hybride*", Editions universitaires européennes.
- [Lev, 44] Levenberg K., (1944) "A method for the solution of certain problems in least squares", *Quart. Appl. Math.*, Vol. 2, pp. 164–168.
- [Li, 06] Li Z., (2006) "Contribution a l'élaboration d'algorithmes d'isolation et d'identification de défauts dans les systèmes non linéaires", *PhD thèses*, INSA, Toulouse, France.
- [Mac, 43] Mac-culloch W. S., & Pitts W., (1943) "A logical calculus of the ideas immanent in nervous activity". *Butt. math. Biophysics*, S, 115-133.
- [Maq, 97] Maquin D., Cocquempot V., Cassar J.P., Staroswiecki M. & Ragot J., (1997) "Generation of analytical redundancy relations for FDI purposes", *IEEE Int. Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, SDEMPED'97*, Carry-le Rouet, France, pp. 270-276, September 1-3rd.
- [Mar, 63] Marquardt D. (1963). "An algorithm for least-squares estimation of nonlinear parameters". *SIAM J. Appl. Math*, Vol. 11, pp. 431–441.
- [Mas, 86] Massoumnia M.A., (1986) "A geometric approach to the synthesis of failure detection filters", *IEEE Trans. On Automatic Control*, vol. 31, no 9.
- [Mas, 89] Massoumnia, M. A., Verghese G. C. & Willsky A. S., (1989) "Failure detection and identification", *IEEE Transaction on Automatic Control*, 34(3), pp. 316-321.
- [Mel, 01] Meléndez J. & Colomer J., (2001) "Episodes representation for supervision. Application to diagnosis of a level control system", *Workshop on Principles of Diagnosis (DX'01)*, Sansicario, Italia.
- [McD, 90] McDuff, R.J., & Simpson P.K. (1990), "An adaptive resonance diagnostics system". *Journal of Neural Network Computing*, 2, pp. 19-29.
- [Min, 56] Minsky M. L, 1956 "Some universal elements for finite automata". In C. E. Shannon & J. McCarthy (Eds.), *Automata studies*. Princeton: Princeton Univer. Press. Pp. 117-128.
- [Min, 69] Minsky M., Papert S., (1969) "*Perceptrons*". MIT Press, Cambridge, MA.
- [Moh, 05] Mohamed D. M., (2005) "*Diagnostic et conduit des systèmes de production par approche a base de connaissances*", thèse de doctorat d'état en science, université El-hadj Lakhdar, Batna.
- [Mok, 08] Mok H. T. & Chan C. W., (2008) "Online fault detection and isolation of nonlinear systems based on neurofuzzy networks", *Engineering Applications of Artificial Intelligence*, vol. 21, no. 2, pp. 171–181.
- [Mou, 05] Mouss M. D., (2005) "*Diagnostic et conduit des systèmes de production par approche a base de connaissances*", *thèse de doctorat science*, université El-hadj Lakhdar -Batna.
- [Mru, 08] Mrugalski M., Witczak M., & Korbicz J., (2008) "Confidence estimation of the multi-layer perceptron and its application in fault detection systems", *Engineering Applications of Artificial Intelligence*, vol. 21, no. 6, pp. 895–906.

- [Nau, 94] Nauck D., (1994) " Fuzzy Neuro Systems: An Overview". In: R. Kruse, J. Gebhardt and R. Palm eds., *Fuzzy System in Computer Science*. Series Artificial intelligence, (Vieweg Verlag, Wiesbaden) 91-107. 2. Auflage 1996.
- [Nau, 95] Nauck D. & Kruse R., (1995) "NEFCLASS - A neuro-fuzzy approach for the classification of data", in: K.M. George et al eds., *Applied Computing, Proceeding of the 1995 ACM Symposium on Applied Computing, Nashville*.
- [Nel, 01] Nelles O., 2001 "*Non-Linear Systems Identification. From Classical Approaches to Neural Networks and Fuzzy Models*", Springer, Berlin, Germany.
- [Noc, 99] Nocedal J. & Wright S.J., (1999) "Numerical Optimization", Springer, New York.
- [Nyb, 97] Nyberg, M. & Nielsen L., (1997) "Parity functions as universal residual generators and tool for fault detectability analysis". In *Proceedings of the 36th IEEE Conference on Decision and Control*, San Diego, USA, pp. 4483 – 4489.
- [Par, 04] Parizeau M., (2004) "*Réseaux de Neurones GIF-21140 et GIF-64326*", Université de Laval.
- [Pat, 89] Patton, R.J., Frank P.M. & Clark R.N., (1989) "*Fault diagnosis in dynamic systems: theory and application*". International Series in Systems and Control Engineering. Prentice Hall, Englewood Cliffs. New Jersey, USA.
- [Pat, 91] Patton R.J., Chen J., (1991) "A re-examination of the relationship between parity space and observers-based approaches in fault diagnosis", *Revue Européenne Diagnostic et sûreté de Fonctionnement*, Vol. 1, pp. 183-200.
- [Pat, 97] Patton R.J., Chen J., (1997) "Observer-based Fault Detection and Isolation: Robustness and Applications", *Control Engineering Practice*, 5(5): 671-682.
- [Pat, 00] Patton R. J. & Chen J., (2000) "On eigenstructure assignment for robust fault diagnosis", *International Journal of Robust and Nonlinear Control*, vol. 10, no. 14, pp. 1193-1208.
- [Pat, 05] Patan K. & Parisini T., (2005) "Identification of neural dynamic models for fault detection and isolation: the case of a real sugar evaporation process", *Journal of Process Control*, vol. 15, no. 1, pp. 67–79.
- [Rag, 92] Ragot J. Maquin D., Bloch G., (1992) "Positionnement de capteurs pour des systèmes décrits par des équations bilinéaires", *Revue Européenne Diagnostic et Sûreté de Fonctionnement*, vol. 2, pp.115-121.
- [Ros, 58a] Rosenblatt, F., (1958) "*The perceptron: A theory of statistical separability in cognitive systems*". Buffalo: Cornell Aeronautical Laboratory, Inc. Rep. No. VG-1196-G-1.
- [Ros, 58b] Rosenblatt F., (1958) "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain", *psychological review* vol. 65, 6.
- [Ros, 62] Rosenblatt F., (1962) "*Principles of Neurodynamics*". Spartan, New York
- [Rum, 86a] Rumelhart D. E., Hinton G. E., Williams R. J. (1986) "Learning internal representations by error propagation". In: Rumelhart D E, McClelland J. L. et al. (eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, MA, Vol. 1, pp. 318-62.
- [Rum, 86b] Rumelhart, D. E., McClelland, J. L., and the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. MIT Press, Cambridge, MA.
- [Sau, 96] Sauter D., Hamelin F., Ghetie M.A., Fonte C., (1996) "Adaptive residual evaluation of model uncertainties: some methods and results", *Engineering Simulation*, vol. 13, pp. 773-788.
- [Sch, 98] Schubert M., Koppen-Seliger B., & Frank P. (1998) "Recurrent Neural Networks for Nonlinear System Modelling in Fault Detection", In IFAC Symposium, *Fault*

- Detection, Supervision and Safety for Technical Processes*, Kingston Upon Hull, UK, 26-28 August 1997, pp. 701-706.
- [Sco, 03] Scola H. R., Nikoukah R., & Delebecque F., (2003) "Test signal design for failure detection: a linear programming approach", *International Journal of Applied Mathematics and Computer Science*, vol. 13, no. 4, pp. 515–526.
- [Sha, 98] Shayler P., M. Goodman and Ma T. (1998) "Applications of Neural Networks in Automotive Engine Management Systems, In IFAC Symposium, *Fault Detection, Supervision and Safety for Technical Processes*, Kingston Upon Hull, UK, 26-28 August 1997, pp. 899-906.
- [Sod, 89] Soderstrom T. et al., (1989) "*System Identification*", Prentice-Hall International, Hemel Hempstead, Hertfordshire, UK.
- [Sre, 93] Sreedhar F., Fernandez B. & Masada G. Y. (1993) "Robust Fault Detection in Nonlinear Systems Using Sliding Mode Observers", IEEE Conference on Control Applications, September 13-16, Vancouver, Canada.
- [Tan, 07] Tan S. C., Lim C. P., & Rao M. V. C., (2007) "A hybrid neural network model for rule generation and its application to process fault detection and diagnosis", *Engineering Applications of Artificial Intelligence*, vol. 20, no. 2, pp. 203–213.
- [Tom, 02] Tombs M., (2002) "Intelligent and self-validating sensors and actuators", *Computing and Control Engineering Journal*, vol. 13, no. 5, pp. 218–220.
- [Ura, 07] Uraikul V., Chan C. W., and Tontiwachwuthikul P., (2007) "Artificial intelligence for monitoring and supervisory control of process systems", *Engineering Applications of Artificial Intelligence*, vol. 20, no. 2, pp. 115–131.
- [Van, 96] Van Schrik D., (1996) "estimator schemes for instrument fault detection and isolation", *IEEE, International Conf. On System, Man and Cybernetic*, pp. 406-411, 1996
- [Ven, 90] Venkatasubramanian, V., Vaidyanathan, R. & Yamamoto, Y. (1990) "Process Fault Diagnosis Using Neural Networks - I. Steady-State Processes", *Computers and Chem. Engng.*, 14, 699-712.
- [Ven, 03a] Venkatasubramanian V., Rengaswamy R., Yin K. & Kavuri S. N., (2003) "A review of process fault detection and diagnosis Part I: Quantitative model-based methods", *Computers and Chemical Engineering*, 27, pp. 293-311.
- [Ven, 03b] Venkatasubramanian V., Rengaswamy R., Kavuri S. N., Yin K, (2003) "A Review of Process Fault Detection and Diagnosis Part III: Process history based methods", *Computers and Chemical Engineering*, 27, pp. 327-346.
- [Vie, 04] Vieira J., Dias F. M., & Mota A., (2004) "Artificial neural networks and neuro-fuzzy systems for modelling and controlling real systems: a comparative study", *Engineering Applications of Artificial Intelligence*, vol. 17, no. 3, pp. 265–273.
- [Vil, 97] Villemeur A, (1997) "*Sûreté de fonctionnement des systèmes industriels, Fiabilité - Facteurs Humains-Informatisation*", Collection de la Direction des Études et Recherches d'Électricité de France, Édition Eyrolles, Paris.
- [Wal, 79] Walker B.K. & Gai E., (1979) "Fault detection threshold determination techniques using Markov theory", *Int. J Guidance, Control and Dynamics*, vol. 2, n°4, pp. 313-319.
- [Wei, 88] Weiss J.L., (1988) "Threshold computations for detection of failures in SISO systems with transfer function errors", *Proc. of the American Control Conf.*, pp. 2213-2218.
- [Wer, 77] Werbos P. (1977). *Advanced Forecasting Methods for Global Crisis Warning and Models of Intelligence. General Systems Yearbook.* (Appendix B).

- [Wer, 89a] Werbos P., (1989) "A Menu of Designs for Reinforcement Learning Over Time". In *Neural Networks for Robotics and Control*. Editors: Miller, W. T., Sutton, R. S., Werbos. Cambridge: MIT Press.
- [Wer, 89b] Werbos P., (1989) "Backpropagation and Neurocontrol: A Review and Prospectus". In *Proceedings of the International Joint Conference on Neural Networks* (IEEE, June).
- [Wer, 89c] Werbos P., (1989) "The consistency of HDP applied to a simple reinforcement learning problem". In *Neural Networks*, forthcoming.
- [Wer, 89d] Werbos P., (1989) "Maximizing long-term gas industry profits in two minutes in Lotus using neural network methods". *IEEE Transactions Systems Man and Cybernetics*.
- [Wil, 86] Williams B. C., (1986) "Doing Time: Putting qualitative reasoning on firmer ground", Proc. AAAI-S6, National Conference on Artificial Intelligence, pp. 105-112.
- [Wil, 76a] Willsky A.S., (1976) "A survey of design methods for failure detection in dynamic systems", *Automatica* 12, pp. 601-611.
- [Wil, 76b] Willsky A.S. & Jones H.L., (1976) "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems", *IEEE Trans. On Automatic Control*, pp. 108-112.
- [Wit, 07] Witczak M., (2007) "*Modeling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems. From Analytical to Soft Computing Approaches*", Springer, Berlin, Germany.
- [Xin, 05] Xing-Gang Y. & Edwards. C., (2005) "Robust sliding mode observer-based actuator fault detection and isolation for a class of nonlinear systems", *Proceedings of the 44th IEEE Conference on Decision and Control CDC 05 and the European Control Conference ECC 05*, December 12-15, Seville, Spain.
- [Yag, 91] Yager R.R., 1991 "Using fuzzy logic to build neural networks", in: R.Lowen and M.Roubens eds., *Proceedings of the Fourth IFSA Congress, Vol. Artificial intelligence*, Brussels, 210-213.
- [Yag, 92] Yager R.R., (1992) "Implementing fuzzy logic controllers using a neural network frame-work", *Fuzzy Sets and Systems*, 48. 53-64.
- [Yag, 94] Yager R.R. and Zadeh L.A. eds. (1994), "*Fuzzy Sets, Neural Networks, and Soft Computing*" Van Nostrand Reinhold, New York.
- [Yag, 08] Yaguol L., Zhengjia H., & Yahyang Z., (2008) "A new approach to intelligent fault diagnosis of rotating machinery", *Expert Systems with Applications*, vol. 35, no. 4, pp. 1593-1600.
- [Yam, 94] Yam Y. & Leung K.S. eds., (1994) "*Future Directions of Fuzzy Theory and Systems*" (World Scientific, Singapore).
- [Yam, 94] Yamakawa T., (1994) "A neo fuzzy neuron and its applications to system identification and prediction of chaotic behaviour", in: J.M. Zurada, R.J. Marks and C.J. Robinson eds., *Computational Intelligence: Imitating Life* (IEEE Press, New York) 383-395.
- [Yan, 94] Yan J., Ryan M. & Power J., (1994) "*Using Fuzzy Logic - Towards Intelligent Systems*" (Prentice-Hall, Englewood Cliffs).
- [Yan, 99] Yang J. C. & Clarke D. W., (1999) "The self-validating actuator", *Control Engineering Practice*, vol. 7, no. 2, pp. 249-260.
- [Yu, 98] Yu D., Gomm J. & Williams D., (1998) "Diagnosing Sensor Faults in a Chemical Process Via RBF Network", In *IFAC Symposium, Fault Detection, Supervision and Safety for Technical Processes*, Kingston Upon Hull, UK, 26-28 August 1997, pp. 893-898.

- [Zha, 94] Zhang J. and Morris A. (1994) "On-line process fault diagnosis using fuzzy neural networks". *Intelligent Systems Engineering*, 3. 37-47.
- [Zha, 00] Zhang Q., (2000) "A new residual generation and evaluation method for detection and isolation of faults in nonlinear systems". *International Journal of Adaptive Control and Signal Processing*, vol. 14, pp. 759-773.
- [Zha, 08] Zhao J., Huang J., & Sun W., (2008) "On-line early fault detection and diagnosis of municipal solid waste incinerators", *Waste Management*, vol. 28, no. 11, pp. 2406–2414.
- [Zwi, 95] Zwingelstein G., (1995) "*Diagnostic des Défaillances Théorie et Pratique pour les Systèmes Industriels*", Editions Hermès.