

Diagnostic Industriel par Neuro-Floue

Application à Un Système de Production

R.Mahdaoui* H.L Mouss**
Laboratoire d'Automatique et Productique (LAP) Université de Batna
1, Rue Chahid Boukhrouf 05000 Batna
* mahdaouiraf@caramail.com ** hayet_mouss@yahoo.fr

Résumé- Dans cet article, nous proposons une approche de reconnaissance de formes neuro-floue (RNF) statistique pour le diagnostic d'un système de production. Les caractéristiques de l'approche neuro-floue pour le diagnostic industriel sont présentées en combinant les concepts des réseaux de neurones artificiels (RNA) et celles de la logique floue (LF). Le système de production choisie est l'atelier de clinkérisation de la SCIMAT, cimenterie de Ain Touta « Batna ».

Mots clefs - **Diagnostic, réseaux de neurones artificiels, logique floue, neuro-floue, reconnaissance de formes.**

I. INTRODUCTION

La fonction de diagnostic est une tâche très complexe et ne peut être qu'une partie résolue par la technique de reconnaissance de formes [8], le diagnostic par RdF peut être présenté comme une solution alternative à l'approche modèle puisque les modes de fonctionnement sont modélisés, pas d'une manière analytique, mais en utilisant uniquement un ensemble des mesures de ce modes [8].

Aussi l'expert humain dans sa mission de diagnostiquer la cause d'une défaillance de tout un système, utilise des informations quantitatives ou qualitatives. D'un autre côté, malgré les résultats grandement surprenants obtenus par les RNA en surveillance et précisément en diagnostic ils restent tout même assez loin d'égaliser les capacités sensorielles et de raisonnement de l'être humain. La logique floue fait un autre axe très efficace en diagnostic industriel.

Aussi, peut-on remplacer l'expert humain pour d'automatiser la tâche de diagnostic à 100%, en utilisant l'approche neuro-floue ? Et Comment fait l'expert humain pour rassembler toutes les informations lui permettant d'apprendre sa décision ?

Notre objectif consiste à faire une association (adaptation) de techniques de la logique floue avec les techniques neuronales (un système neuro-floue), choisir les types de réseaux de neurone, déterminer les règles floues et enfin la structure du système neuro-floue pour automatiser le maximum de la tâche diagnostic.

Afin d'atteindre cet objectif nous organisons ainsi cet article.

La première partie présente les principales architectures et principes de fonctionnement des systèmes neuro-flous et leurs applications. La deuxième partie est dédiée à l'atelier de clinkérisation au niveau de cimenterie. Enfin,

dans la troisième partie nous proposons un système neuro-floue pour le diagnostic du système de production.

II. LES SYSTÈMES NEURO-FLOUS

Les Systèmes hybrides qui combinent la logique floue, les réseaux neurones, les algorithmes génétiques, et les systèmes experts prouvent leur efficacité dans une variété de problèmes de monde réel et dans l'industrie.

Chaque technique intelligente a des propriétés particulières (par exemple capacité d'apprentissage, explication de décisions). Chaque technique convient à 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. Aussi pour les Systèmes de la logique floue qui peuvent raisonner avec l'information imprécise et expliquer leurs décisions mais ne peuvent cependant pas acquérir automatiquement les règles qu'ils l'utilisent pour prendre ces décisions.

Ces limites ont été une raison derrière la création de systèmes hybrides intelligents où ; deux ou plus de techniques sont combinées afin de vaincre les limitations d'une seule technique.

Afin de résumer l'apport du neuro-floue, le tableau 1 regroupe les avantages et les inconvénients de la logique floue et des RNAs.

TABLEAU I
COMPARAISON ENTRE RNA ET LF

RNA	SIF
la base de règle ne peut être utilisée	la base de règle peut être utilisée
L'apprentissage peut être démarré de zéro	Pas d'apprentissage (utilise la connaissance linguistique).
Boîte noire	Interprétable (la règle IF-THEN)
Complexion des algorithmes d'apprentissage	Universel et implémentation simple
Difficulté pour extraire la connaissance	La connaissance doit être disponible

Les applications montrent les avantages de la logique floue quand le modèle des systèmes est difficile à implémenter. Malheureusement, avec l'augmentation dans la complexité du modèle de l'existence du processus, nous avons rencontrés une difficulté pour développer des règles floues et des fonctions d'appartenance.

Cela a mené au développement d'une autre approche qui est principalement connu comme approche neuro-floue. L'intégration des réseaux de neurones et les systèmes d'inférence flous peut être formulé en trois principales catégories : coopérative, concurrents et neuro-flou hybride [5].

Aussi, on peut dire que les systèmes neuro-flous sont des modèles de connexion qui permettent l'apprentissage comme un RNA, mais leur structure peut être interprétée comme un ensemble de règles floues.

A. Définition

Un système neuro-flou hybride 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 [7].

B. Principe de fonctionnement

Les Réseaux de neuro-flous hybride apprennent des rapports et des modèles en utilisant un algorithme d'apprentissage supervisé qui examine les données dans un ensemble de la formation qui consiste en exemples d'entrées et leurs sorties associées. Pendant la phase d'apprentissage, un Réseau neuro-flou hybride 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 d'un Réseau neuro-flou est vérifiée après que le cycle de d'apprentissage soit complet en utilisant un ensemble séparé d'entrées et sorties appelé l'ensemble de la validation "Fig. 1,".

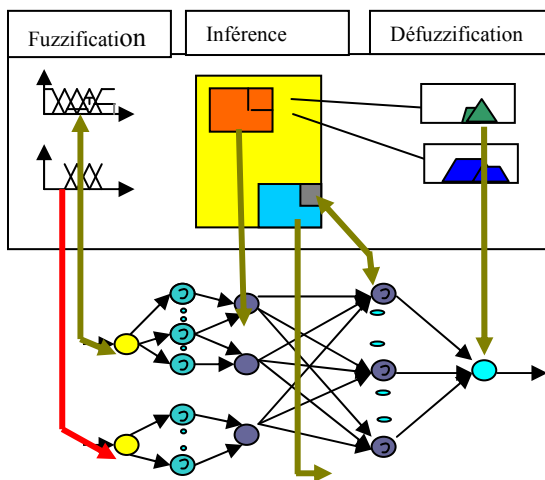


Fig. 1. Principe de fonctionnement de RNF.

L'apprentissage par renforcement est une addition récente aux algorithmes développés pour les réseaux neuro-flous hybrides.

Ce type d'apprentissage est utilisé quand les sorties désirées pour une entrée donnée n'est pas connue pendant une séquence d'actions qui ont lieu. Pendant le renforcement qui apprend le processus, la seule réaction donnée au système est une indicatrice rugueuse de performance, tel que « bon », « mauvais », « trop lent », ou « trop vite ».

C. Le perceptron flou

L'architecture du perceptron flou est identique à celle du perceptron multi-couche usuel, mais les poids sont modélisés par des ensembles flous. Les activations, les sorties, et les fonctions de propagation seront changées. Aussi l'intention de ce modèle est qu'il soit interprétable par des règles linguistiques et peut utiliser des bases de connaissance des règles à priori, donc l'apprentissage peut ne pas démarrer à zéro (la base de règles n'est pas vide).

La "Fig. 3" illustre un perceptron flou avec N entrées, M règles floues et P sorties.

Les neurones de la première couche réalisent la première phase de l'inférence floue. A chaque observation correspond des neurones de la première couche d'entrée, chargées de calculer les degrés d'appartenance des variables floues aux différents sous ensembles de termes linguistiques. La fonction d'appartenance utilisée est la fonction triangulaire symétrique.

Les neurones de la deuxième couche calculent le degré de vérité des antécédents des règles floues par l'intermédiaire de t-norme.

Le nombre de neurones de cette couche est égal à la taille de base de règles. La connexion entre la première couche et la deuxième n'est pas totale car définie par la structure de la règle linguistique.

Les valeurs de sortie de la troisième couche sont le maximum des valeurs d'activation de toutes les unités de règles qui sont associé à telle classe.

Afin d'implémenter ce type de perceptron flou et l'exploité pour diagnostiquer un système de production dédiée nous avons proposer un logiciel informatique NEFDIAG.

III. PRESENTATION DE NEFDIAG

A. Introduction

Le NEFDIAG est un programme informatique de simulation interactive, réalisé au sein de LAP (université de Batna) écrit sous DELPHI, consacré essentiellement à la création, l'apprentissage et au test d'un système neuro-flou de classification des pannes d'un procédé industriel dédié.

NEFDIAG modélise un classificateur flou Fr avec un ensemble de classes $C = \{c_1, c_2, \dots, c_m\}$.

B. structure et apprentissage

NEFDIAG peut être représenté comme un type spécial de perceptron floue, à trois couches utiliser pour classifier des défaillances.

NEFDIAG fait son apprentissage par un ensemble de formes, telle que chaque forme sera affectée (classée) vers une des classes prédéfinies. NEFDIAG génère les règles floues par un parcours des données ensuite optimise les règles par apprentissage les paramètres des sous ensembles flous qui sont utilisées pour partitionner les données « caractéristiques » des formes à classées et les paramètres des données.

NEFDIAG modélise un classificateur flou Fr avec un ensemble de classes $C = \{c_1, c_2, \dots, c_m\}$. Après l'apprentissage, NEFDIAG peut être utilisé pour classifier une nouvelle observation, le système peut être représenté sous forme de règles floues :

Si Variable1 est A_1
 Variable2 est A_2
 Variable3 est A_3
 Variable n est A_n

Alors la forme $(x_1, x_2, x_3, \dots, x_n)$ est classée dans la classe « mode de défaillance 1 ».

Telle que A_1, A_2, A_3, A_n sont des termes linguistiques représentés par des ensembles flous. Cette caractéristique permettra de connaître les analyses sur nos données, et utiliser ces connaissances pour les classifier.

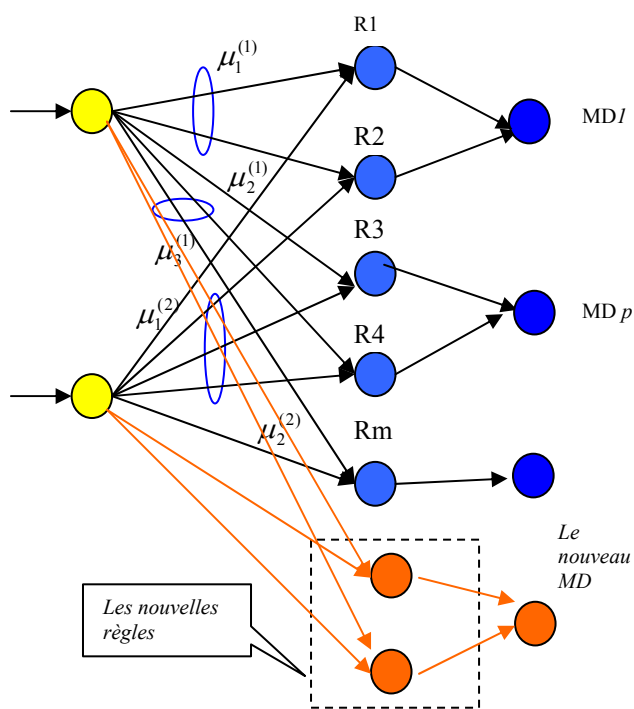


Fig. 2. Architecture d'un RNF.

L'apprentissage des réseaux de neuro-flous artificiels est une phase qui permet de déterminer ou de modifier les paramètres du réseau, afin d'adopter un comportement désiré.

L'étape d'apprentissage est basée sur la descente de gradient d'erreur quadratique moyenne commise par le réseau RNF.

Le système NEFDIAG peut démarre avec une base de connaissance partielle des formes, et peut ensuite la raffiner durant l'apprentissage, ou il peut démarrer avec une base de connaissance vide.

L'utilisateur définit le nombre initial des fonctions d'appartenance pour partitionner les domaines des données d'entrées. Et il faut aussi spécifier le nombre k , nombre maximum des neurones des règles qui seront créés dans la couche cachée.

Les principales étapes de notre algorithme d'apprentissage sont ainsi présentées.

Initialisation : pour chaque donnée issue par des capteurs il y a une unité d'entrée, pour chaque mode de défaillance il y a une unité de sortie. Pour chaque unité

d'entrée une partition floue initiale est spécifiée « exp. Un nombre des fonctions d'appartenance triangulaire ».

Apprentissage des règles : Le système NEFDIAG peut démarre avec une base de connaissance partielle de formes, et peut ensuite la raffiner durant l'apprentissage « Fig. 3 ».

La règle sera créée par la recherche (pour une forme donnée F) la combinaison des fonctions d'appartenance tel que chaque entrée produise la plus grande fonction d'appartenance « Fig. 4 ». Si cette combinaison n'est pas identique pour les règles existantes dans la base des règles, et le nombre de règles n'est pas maximum, alors une règle sera créée et ajoutée à la base des règles « Fig. 2 ».

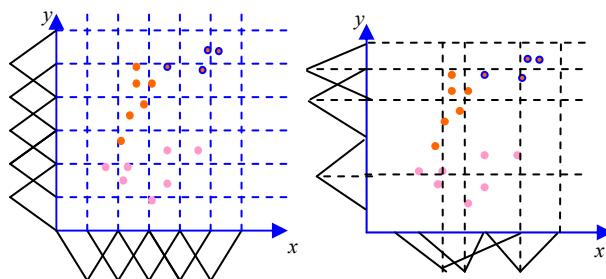


Fig. 3. Apprentissage des règles.

L'algorithme d'apprentissage détecte (calcule) tous les antécédents des règles et ensuite crée la liste des antécédents. En premier temps cette liste est vide, ou contient des antécédents de règles des connaissances a priori.

L'algorithme sélectionne ensuite un conséquent pour chercher l'antécédent A et créer la liste de base de règle candidates.

Les meilleures règles seront sélectionnées de la base des règles candidates, en base de mesure de performance [7].

$$P_R = \frac{1}{S} \sum_{(p,t) \in \Gamma} (-1)^c R(p) \quad (1)$$

$$c = \begin{cases} 0 & \text{si } classe(p) = con(R) \\ 1 & \text{si non} \end{cases} \quad (2)$$

Dans ce cas quelques classes (mode de défaillance) ne seraient pas représentés dans la base de règles, si les règles pour ce mode de défaillance à une valeur de performance très petite.

Apprentissage des Fonctions d'Appartenance : pour l'apprentissage des fonctions d'appartenance, une simple rétro propagation est utilisée. Elle dépend de l'erreur de sortie pour chaque unité de règles. Chaque règle change ses fonctions d'appartenance par le changement de leurs supports « Fig. 4 », « Fig. 5 ».

Il faut que l'erreur de chaque règle soit calculée [5].

$$\varepsilon_r = \tau_r (1 - \tau_r) \frac{1}{m} \sum_{j=1}^m (2\nu_r^{(j)}(t_j) - 1) |E_j| \quad (3) \text{ où}$$

τ_r est le degré d'accomplissement d'une règle r .

Après l'apparition d'un autre nouveau mode de défaillance dans la phase d'apprentissage notre système neuro-floue va faire une adaptation ou une restructuration du système pour être adapté à la nouvelle situation "Fig. 2,".

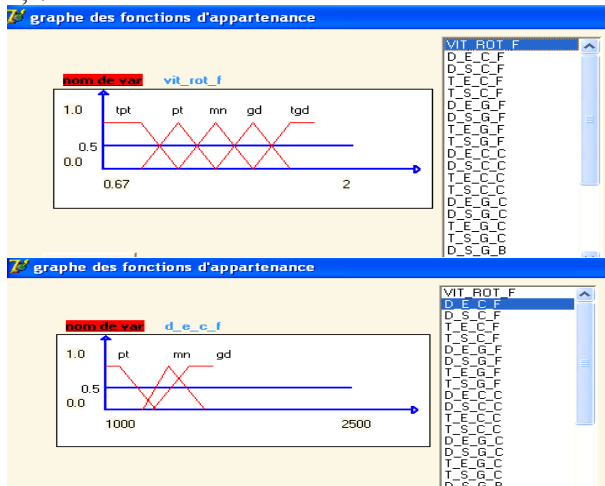


Fig. 4. Partition de vitesse rotation de four avant et après apprentissage.

D'abord les couches de règles (ou la base de règles) ajout tous les règles du mode de défaillance détecté. Ensuite dans la couche des modes de défaillance, un autre nœud sera connecté au réseau neuro-flou "Fig. 2,"

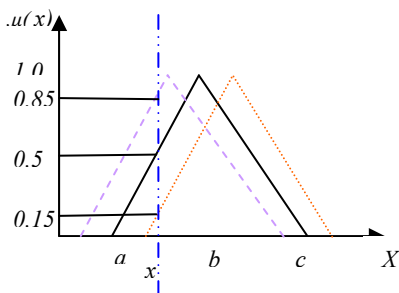


Fig. 5. apprentissage des FA

IV. L'ATELIER DE CLINKERISATION

Notre application est illustrée sur un procédé industriel de fabrication du ciment. Cette installation fait partie de cimenterie de Ain-Touta (SCIMAT).

Cette cimenterie d'une capacité de 2.500.000 t/an « Deux fours » est composée de plusieurs unités qui déterminent les différentes phases du processus de fabrication du ciment.

L'atelier de cuisson regroupe deux fours dont le débit clinker est de 1560 t/h. le broyage de ciment comprend deux broyeurs de 100t/h chacun. Les expéditions du

ciment s'effectuent à partir de deux stations, une pour les camions et une autre pour les wagons "Fig. 6,".

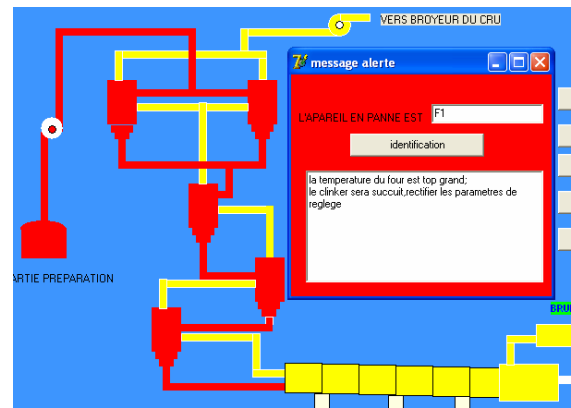


Fig. 6 Prototype d'atelier de clinkeristaion par NEFDIAG après l'apparition d'un défaut.

V. 4. DIAGNOSTIC PAR NEURO-FLOUE

A. Analyse de dysfonctionnements

Cette étape à pour objectif l'identification des dysfonctionnements pouvant affecter la mission du système. Cette analyse est grandement facilitée par la reconnaissance des modèles structurels et fonctionnels de l'installation.

Pour l'analyse des dysfonctionnements nous avons retenu la méthode d'analyse des modes de défaillance, de leurs effets et leur criticité (AMDEC).

En se basant sur l'étude réalisée par [6], sur l'atelier de cuisson, nous avons élaboré une AMDEC en ne considérant que les modes de défaillances les plus critiques (criticité>10) et ceci pour des raisons de simplicité [6].

Donc nous avons un système neuro-floue de 27 entrées et quatre sorties qui ont été créés pour faire un diagnostic de notre système. Les règles qui sont créés avec le système sont des connaissances a priori, c- a- d la base de règle a priori. Chaque variable ayant une partition initiale sera modifiée au long de la phase d'apprentissage (nombre de ensembles floue pour chaque variable).

Le raisonnement pour le diagnostic est décrit sous forme de règles floues à l'intérieur de notre système neuro-floue. L'avantage principal de l'utilisation de la base de règles floues réside dans sa modularité et sa facilité d'extension (suppression ou rajout d'autres règles).

La base de règles initiale pour établir le diagnostic des défaillances est construite en exploitant le modèle élaboré en phases dysfonctionnement de notre système (AMDEC).

En effet, cette analyse permet d'établir les liens de causes à effets entre les composantes défaillants et les symptômes observés. Ces liens seront représentés sous formes de règles floues construisant la base de connaissances qui sera apprenies a plus tard et ensuite testés, pour effectuer les raisonnements floues nécessaire et

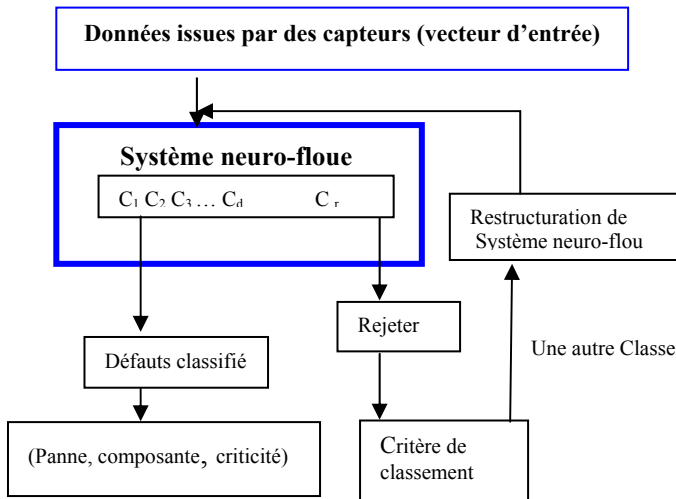


Fig. 7. Le diagnostic par RDFS NF.

Ensuite la détection des anomalies est représentée sous forme de message d’alarme destiné à signaler à l’opérateur (utilisateur) l’apparition d’une anomalie (ou des anomalies) et permet d’identifier le composant responsable à l’aide d’une base de donnée qui stock toutes les informations fournies par l’AMDEC (mode de défaillance, causes possibles, équipement, effets sur le système).

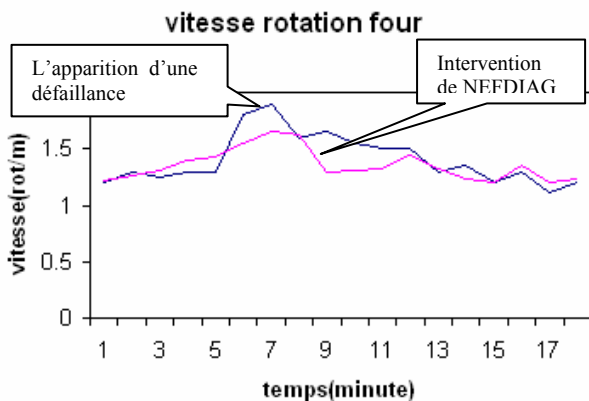


Fig. 8. la vitesse de rotation après l’apparition d’une anomalie ensuit l’intervention de NEFDIAG.

Après apparition d’une anomalie, un message d’alarme permet à l’opérateur de détecter le dysfonctionnement et aussi de localiser le composant responsable. La “Fig. 2,” illustre le système de cuisson avec la présence d’un dysfonctionnement.

Notons que dans cette étude, les anomalies ou dysfonctionnements désignent des anomalies fonctionnelles.

Après l’affichage du message, l’opérateur peut consulter ce dernier pour plus d’information ou bien le supprimer “Fig. 6,”. Ensuite NEFDIAG fait des interventions pour réguler les variables qui sont l’origine de la défaillance courante “Fig. 8,”.

Dans cet article, nous avons présenté un nouvel outil de diagnostic neuro-flou en suivant l’approche AMDEC, nous avons détaillé la mise en œuvre d’un exemple d’application industrielle par l’outil de développement NEFDIAG.

Nous avons illustré d’utilisation de notre outil d’aide au prédiction diagnostic sous forme d’un prototype « NEFDIAG » installer sur un PC, ce développement a été réalisé au « LAP ».

Nous avons abordé les différentes étapes à suivre pour l’élaboration d’un système d’aide au diagnostic à partir des méthodes de classification et des reconnaissances floues des formes.

NEFDIAG est représenté comme un type spécial de perceptron flou, à trois couches utilisé pour classifier des défaillances, en utilisant le système neuro-flou de type 3.

NEFDIAG fait un apprentissage à deux phases. Un apprentissage de règles, et génère les règles floues par le parcours de données et optimise les règles par apprentissage des paramètres des ensembles flous qui sont utilisés pour partitionner les données des formes à classer et les paramètres des données.

REFERENCES

- [1] H.K. Kwan and Y.Cai, "A fuzzy neural network and its application to pattern recognition" IEEE Transactions on Fuzzy Systems, 3 pp. 185-193. 1994
- [2] J.-S. Roger Jang "ANFIS: Adaptive-network-based fuzzy inference system " IEEE Trans. Syst., Man, and Cybernetics, 23(1993) 665-685.
- [3] J.M. Keller and H.Tahani "Implementation of conjunctive and disjunctive fuzzy logic rules with neural networks" International Journal of Approximate Reasoning,6(1992) 221-240.
- [4] G. Zweingelstein "Diagnostic des défaillances, théories et pratique pour les systèmes industriels" Col. Traité des nouvelles technologies, séries diagnostic et maintenance, Hermès, 1995.
- [5] L.-X wang and J.M Mendel "Generation fuzzy rules by learning from examples" IEEE trans. Syst., Man, and Cybernetics, (1992) 22(6):1414-1427.
- [6] D. Mouss "Diagnostic et conduite des systèmes de production par approche a base de connaissances" Thèse de doctorat Université de Batna, 2006.
- [7] D. Nauck "Neuro-fuzzy systems: review and prospects"European congress on intelligent technique and sift computing (EUFIT'97), Aachen, sep.8.11, (1997), pp.1044-1053.
- [8] D. Racoceanu "contribution à la surveillance des systèmes de production en utilisant les techniques de l'intelligence artificielle" habilitation à diriger des recherches, université de Franche Comté de Besançon, 2006.