



Pronostic de défaillances : Maîtrise de l'erreur de prédiction.

Romain Adeline, Rafael Gouriveau, Noureddine Zerhouni

► To cite this version:

Romain Adeline, Rafael Gouriveau, Noureddine Zerhouni. Pronostic de défaillances : Maîtrise de l'erreur de prédiction.. LISMMA, CRAN, ENSTIB. 7ème Conférence Internationale de MObilisation et SIMulation, MOSIM'08., Mar 2008, Paris, France. 1 (sur CD ROM), 10 p., 2008. <hal-00270725>

HAL Id: hal-00270725

<https://hal.archives-ouvertes.fr/hal-00270725>

Submitted on 7 Apr 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PRONOSTIC DE DÉFAILLANCES : MAÎTRISE DE L'ERREUR DE PRÉDICTION

Romain ADELINÉ, Rafael GOURIVEAU, Nouredine ZERHOUNI

ENSMM-LAB Besançon
24 rue Alain Savary
25000 Besançon
romain.adeline@gmail.com, {rgourive, zerhouni}@ens2m.fr

RESUME : *Le travail rapporté ici traite globalement de la spécification et du développement d'un système de pronostic de défaillances. De ce point de vue, beaucoup de développements visant la proposition de méthodes de prévision existent dans la littérature. La majorité d'entre elles portent sur la construction de modèles capables de minimiser l'erreur de prédiction d'une situation future. Cependant, peu traitent de la maîtrise de cette erreur. C'est ce qui fait l'objet de ce papier et pour lequel nous proposons d'exploiter le système ANFIS (système d'inférence floue paramétré par apprentissage neuronal). Après avoir positionné l'activité de pronostic dans le cadre de la maintenance industrielle, nous présentons le réseau ANFIS. Nous étudions les pistes permettant de maîtriser l'erreur de prédiction d'un tel système, notamment lors de la phase d'apprentissage (optimisation des paramètres du réseau). Les éléments théoriques nécessaires à cette analyse sont décrits, une nouvelle fonction de coût est proposée et l'influence de celle-ci sur les performances du réseau est discutée. Nous illustrons l'ensemble sur un benchmark. La modification proposée permet de réduire la phase d'apprentissage du système de pronostic.*

MOTS-CLES : *Maintenance, pronostic, erreur de prédiction, ANFIS, fonction de coût.*

1. INTRODUCTION

Le maintien en condition opérationnelle des systèmes industriels à moindre coût est devenu un facteur critique quant à la performance des entreprises. Les concepts traditionnels de maintenance préventive et corrective sont peu à peu supplantés par une prise en compte plus réactive et proactive des défaillances (Iung *et al.*, 2003), (Muller *et al.*, 2007). Aussi, le pronostic se révèle être une activité tout à fait prometteuse pour ne pas engager de frais de maintenance inopportuns.

Ceci étant, relativement à la spécification d'un outil de pronostic, un problème central peut être soulevé : l'utilité d'un tel système est intimement liée à sa faculté à approximer et à prédire la dégradation d'un équipement. En d'autres termes, un outil de pronostic doit être à même de projeter dans le temps une situation courante. De ce point de vue, beaucoup de développements visant la proposition de méthodes de prévision existent dans la littérature (De Gooijer et Hyndman, 2006). L'essentiel de ces méthodes portent sur la construction de modèles capables de minimiser l'erreur de prédiction. Cependant, peu traitent de la maîtrise de cette erreur, c'est-à-dire de la connaissance et du contrôle de la distribution de cette erreur de prédiction. Or, cette problématique nous semble essentielle en regard du processus global du pronostic : les décideurs, certes nécessitent des indicateurs quant à l'évolution dans le temps de leurs équipements, mais doivent pouvoir y associer une confiance afin de mettre en œuvre des stratégies de

maintenance adéquates. C'est ce qui fait l'objet du travail présenté ici.

Les systèmes réels sont complexes et de comportement généralement non-stationnaire et non-linéaire ce qui rend difficile une étape de modélisation. Pourtant, l'implémentation d'un outil de prédiction doit s'accommoder de cela. Aussi, différentes techniques d'intelligence artificielle testées sur des problèmes de prédiction ont montrées de meilleures performances que celles de méthodes « conventionnelles » (Wang *et al.*, 2004), (Yam *et al.*, 2001), (Zhang *et al.*, 1998). Il ressort de ces travaux que les réseaux neuro-flous sont particulièrement appropriés. Dans cet ensemble, notre travail traite plus spécifiquement du système ANFIS (Adaptive Neuro-Fuzzy Inference System) proposé par (Jang, 1993). Parmi les différents paramètres influant sur les performances de prédiction d'un tel réseau, notre étude porte plus particulièrement sur l'influence de la fonction de coût utilisée lors de la phase d'apprentissage. Si, classiquement, celle-ci est la mesure d'erreur quadratique, nous avons étudié la proposition d'une fonction alternative dont le but est moins de minimiser l'erreur de prédiction que de maîtriser sa distribution à moyen terme.

Afin de rapporter l'ensemble des travaux, le cœur de ce papier est structuré en quatre parties. La première est consacrée à la description de la problématique de travail ; nous y positionnons également l'activité de pronostic dans le cadre de la maintenance industrielle.

Ensuite, nous présentons les systèmes neuro-flous généralement et le système ANFIS plus particulièrement. Dans la troisième partie, nous étudions les pistes permettant de maîtriser l'erreur de prédiction d'un système du type ANFIS. Les éléments théoriques nécessaires à cette analyse sont décrits et une nouvelle fonction de coût est proposée. L'influence de celle-ci sur les performances du réseau est discutée en quatrième partie. Nous synthétisons l'ensemble dans une conclusion générale et donnons quelques perspectives de travail.

2. FORMULATION DU PROBLÈME

2.1. Maintenance et pronostic

L'activité de maintenance fait appel à différents processus métiers visant d'abord, à « percevoir » certains phénomènes (détection), ensuite, à les « comprendre » (diagnostic), et enfin, à « agir » en conséquence (choix d'actions de maîtrise). Ceci étant, plutôt que de comprendre a posteriori un phénomène qui vient de se manifester (défaillance...), il convient d'en « anticiper » l'apparition afin de recourir à des actions protectrices en conséquence. C'est ce qui peut être entendu sous l'étiquette « pronostic de défaillances ». Aussi, cette thématique devient un cadre de travaux de recherche de premier plan (Ciarapica *et al.*, 2006), (Jardine *et al.*, 2006), (Vachtsevanos *et al.*, 2006) et le « pronostic » est aujourd'hui considéré comme un processus clef dans les stratégies de maintenance. Ce concept mérite ainsi d'être explicité pour situer notre travail.

2.2. Pronostic : prédiction et évaluation

Dans la littérature, le « pronostic » désigne selon les cas un processus visant à déterminer la durée de vie restante d'un système (Jardine *et al.*, 2006) ou la probabilité pour que le système fonctionne durant un certain temps (Lin et Makis, 2003). Ainsi, même s'il n'existe pas d'acceptation du terme « pronostic » totalement consensuelle, un trait marquant peut être identifié : il est souvent assimilé à un processus de prédiction (une situation future doit être appréhendée).

Le comité ISO définit le pronostic comme « l'estimation de la durée de fonctionnement avant défaillance d'un système et du risque d'existence ou d'apparition ultérieure d'un ou de plusieurs modes de défaillance » (ISO 13381-1, 2004). Plus généralement, les définitions du terme « pronostic » sont fondées sur les concepts de défaillance et de dégradation, ce qui laisse transparaître la notion d'acceptabilité. Aussi, le pronostic devrait être basé sur des critères d'évaluation dont les bornes dépendent du système surveillé et d'objectifs de performance. Considérons la figure 1 pour étayer ce point de vue (Dragomir *et al.*, 2007) : la situation « prédite » à « $t+\Delta t$ » ne peut être considérée comme critique que par l'existence d'une limite de dégradation. Sans ce seuil, il est impossible de conclure quant à la gravité de la situation et donc, impossible de mettre en

place une maintenance appropriée. Finalement, le pronostic peut ainsi être décomposé en deux sous-activités : une première visant à prédire l'évolution d'une situation, et une seconde permettant d'évaluer la situation prédite relativement à un référentiel.

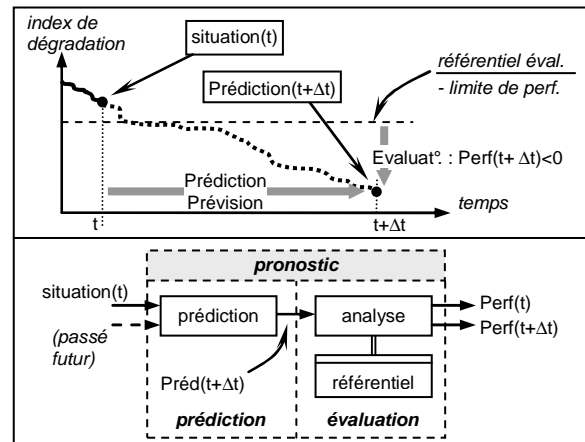


Figure 1. Processus de pronostic (Dragomir *et al.*, 2007)

2.3. Maîtrise de l'erreur de prédiction

Afin d'évaluer par anticipation les performances potentielles d'un système industriel, une grande attention doit être portée au modèle prédictif construit pour supporter (partiellement) le processus de pronostic. Deux aspects nous semblent d'intérêt.

En premier lieu, il est évident que l'erreur de prédiction entre une situation « réelle » et une situation « prédite » doit être le plus faible possible et, au minimum, rester inférieure à un certain seuil. En d'autres termes, la prédiction réalisée peut (certes) être imprécise (et donc incertaine), mais il doit être possible de prendre des décisions non ambiguës en conséquence. Aussi, afin de pouvoir accorder une certaine confiance au système prédictif, nous jugeons souhaitable d'être à même de contrôler l'erreur de prédiction au détriment, si nécessaire et dans une certaine mesure, de sa valeur. Une illustration de ce propos est proposée en figure 2 : bien que non centrée en zéro, la dispersion d'erreur de la courbe 2 laisse présager d'un modèle de prédiction plus fiable que celui de la courbe 1.

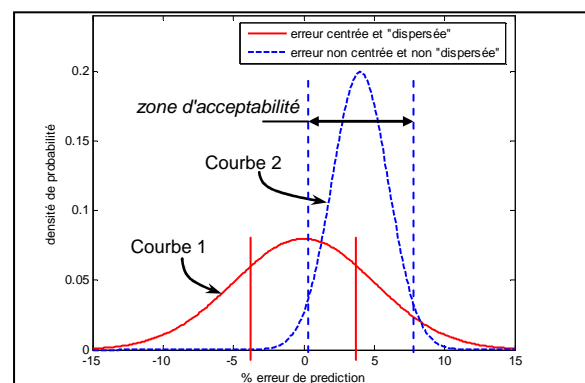


Figure 2. Deux distributions d'erreur de prédiction

Une autre problématique, liée à la précédente, est celle de la stabilité dans le temps de cette erreur de prédiction. Afin d'assurer une certaine généricité au système prédictif construit, sa « qualité » ne doit être que faiblement dépendante de l'horizon décisionnel pour lequel il est utilisé : les dégradations impactant les performances d'un système industriel sont par essence non déterministes et le modèle prédictif doit pouvoir être exploité à court ou moyen terme (figure 3).

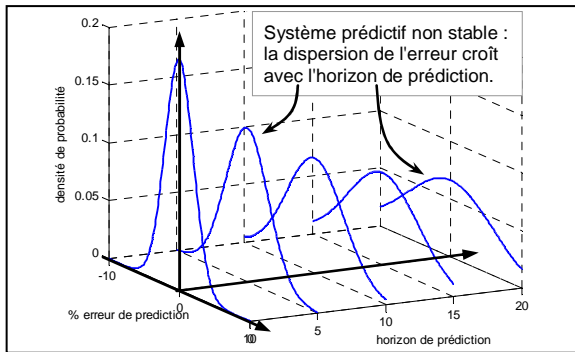


Figure 3. Stabilité dans le temps de l'erreur de prédiction

En définitive et compte tenu qu'il est difficile de garantir que minimiser l'erreur de prédiction permet d'en réduire la dispersion, nous proposons de tâcher de satisfaire ce double objectif. Cette maîtrise de l'erreur de prédiction nous semble essentielle dans le cadre du pronostic industriel. C'est ce que nous discutons dans cet article.

3. SYSTÈME PRÉDICTIF NEURO-FLOU

3.1. Réseaux Neuro-flous et prédiction

En des termes très généraux, (Nauck et Kruse, 1997) proposent une définition des systèmes neuro-flous (NF) : « les systèmes neuro-flous sont des systèmes flous formés par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones ». L'intérêt de construire des systèmes intégrant les réseaux de neurones et les systèmes d'inférence floue (SIF) réside ainsi dans leurs caractéristiques complémentaires. Les SIF exploitent des règles linguistiques du type Si-Alors traduisant une connaissance sur la dynamique d'un système. Cependant, un SIF n'est pas à même d'apprendre : les règles doivent être formulées ce qui est parfois difficile... Aussi, une solution consiste à utiliser les propriétés des réseaux de neurones afin « d'apprendre » la structure floue et d'en adapter les paramètres en conséquence.

Revenant à la problématique du pronostic de défaillances, les réseaux neuro-flous constituent une classe de réseaux adaptatifs particulièrement appropriés pour supporter l'activité de prédiction : ils permettent de capturer la dynamique des systèmes réels sans qu'une phase de modélisation (souvent impossible) ne soit nécessaire. Etant donné une situation caractérisée à la date « t » (entrée), un système neuro-flou est capable (après apprentissage) d'en prédire l'évolution à « $t+prév$ » (sortie). Des développements actuels sur des

problèmes de prédiction de séries temporelles non linéaires montrent d'ailleurs que les performances des systèmes neuro-flous dépassent celles d'autres méthodes aussi bien en terme de précision des résultats que d'efficacité d'apprentissage (Wang *et al.*, 2004). De plus, les règles d'un système NF sont transparentes, ce qui permet la validation et la manipulation par un expert. Enfin, les systèmes NF sont très prometteurs dans les cas où les données disponibles sont limitées. Cela ne constitue pas le cœur de cet article mais c'est l'ensemble de ces caractéristiques qui nous a conduit à utiliser de tels systèmes : les systèmes NF sont des outils adaptés pour supporter l'activité de prédiction du processus de pronostic. La section suivante est consacrée au système ANFIS (Adaptive Neuro Fuzzy Inference System).

3.2. ANFIS : Adaptive Neuro-Fuzzy Inference System

Le système ANFIS est une classe de réseau adaptatif proposé par (Jang, 1993). Il peut être vu comme un réseau de neurones non bouclé pour lequel chaque couche est un composant d'un système neuro-flou et, à ce titre, c'est un « approximateur » universel. Il est ainsi utilisé dans différentes applications de prédictions.

3.2.1. Principe

Considérons la figure 4 pour décrire l'architecture d'un système ANFIS et brièvement expliquer le mécanisme d'inférence d'un tel système. Le réseau proposé possède deux entrées et une sortie. Afin d'alléger la rédaction, ce système n'est pas totalement connexionniste (liens partiels entre les couches 1 et 2) mais les principes décrits ci-après sont généralisables.

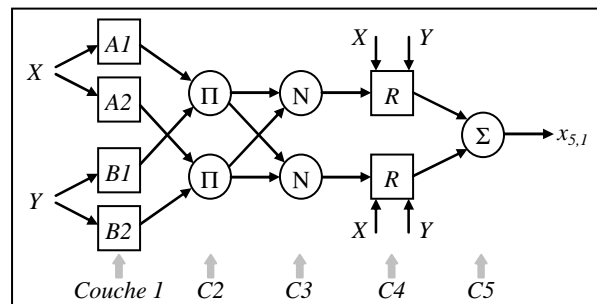


Figure 4. Système ANFIS à deux entrées, une sortie et deux fonctions d'appartenance par entrées

Un système ANFIS réalise une approximation linéaire de la variable de sortie en décomposant l'espace des entrées en différents espaces flous. Exception faite de la couche d'entrée, un système ANFIS est composé de 5 couches. Chaque couche peut comprendre différents nœuds. Nous notons $x_{i,k}$ la sortie du nœud k de la couche l .

Chaque nœud de la couche 1 (de « fuzzification ») se comporte comme une fonction d'appartenance floue (gaussienne, sigmoïde...). Pour le système de la figure 4, nous avons :

$$x_{1,k} = \mu_{A_k}(X) \text{ pour } k = 1 \text{ et } 2, \quad (1)$$

$$x_{1,k} = \mu_{B_{k-2}}(Y) \text{ pour } k = 3 \text{ et } 4. \quad (2)$$

Les sorties des nœuds de cette première couche représentent donc les degrés d'appartenance des entrées X et Y aux différentes ensembles flous A_k et B_{k-2} . Les paramètres inhérents à ces ensembles sont appelés paramètres « prémisses » du réseau (par exemple la moyenne et l'écart type d'une fonction gaussienne).

Dans la couche 2, chaque nœud réalise une T-norme floue. L'opérateur « produit » est généralement utilisé (contrainte de dérivabilité pour pouvoir déployer les algorithmes d'apprentissage).

$$x_{2,k} = \prod_{(l,j) \in \text{Ant}(2,k)} (x_{1,j}), \quad (3)$$

$\text{Ant}(2,k)$ désignant les nœuds antécédents du nœud $(2,k)$.

Dans la couche 3, les sorties des T-normes sont normalisées. Les résultats obtenus représentent le degré d'implication des différentes règles floues (couche 4) dans le résultat final.

$$x_{3,k} = \frac{x_{2,k}}{\sum_{(2,j) \in \text{Ant}(3,k)} x_{2,j}} \quad (4)$$

La couche 4 permet de réaliser une combinaison linéaire du premier ordre des variables d'entrées (approche de Takagi-Sugeno). Les nœuds de cette couche constituent ainsi les règles élémentaires du réseau.

$$x_{4,k} = x_{3,k} [m_{k0} + m_{k1} \cdot X + m_{k2} \cdot Y] \quad (5)$$

Les paramètres m_{k0} , m_{k1} et m_{k2} sont appelés paramètres « conséquents » du système neuro-flou.

La couche de sortie est constituée d'un unique nœud (dans notre cas). Si n désigne le nombre de règles floues (deux dans l'illustration de la figure 4), nous avons :

$$x_{5,l} = \sum_{i=1}^n x_{4,i} \cdot \quad (6)$$

Remarquons que la sortie « prédite » ($x_{5,l}$) est obtenue par moyenne pondérée des sorties des différentes règles.

3.2.2. Apprentissage

Dans les grandes lignes, l'apprentissage consiste en la correction des paramètres (prémisses et conséquents) du réseau afin de généraliser une fonction de transfert (inconnue a priori) entre les entrées et la sortie du réseau. Par exemple, en supposant des ensembles flous A_k et B_{k-2} du type gaussien, le réseau ANFIS décrit ci-avant comporte 14 paramètres devant être optimisés (8 inhérents aux fonctions gaussiennes et 6 à la linéarisation des sorties des règles). À cette fin, une base d'apprentissage est

nécessaire. Celle-ci est constituée d'un ensemble de couples « entrées / sortie » connus (relevé de données). Le déploiement des algorithmes d'apprentissage sur cette base de données permet de construire une fonction d'approximation (de prédiction dans notre cas) de la sortie d (sortie désirée) à partir des nouveaux vecteurs d'entrée.

En général, des algorithmes hybrides sont utilisés. Plus particulièrement, des travaux montrent que la combinaison de l'algorithme de descente du gradient et de l'algorithme LSE (Least Square Estimate – estimation des moindres carrés) permet de réduire la complexité de l'apprentissage et d'en améliorer l'efficacité (Li et Cheng, 2007), (Wang *et al.*, 2004). Nous en retraçons ci-après les grandes lignes. (Jang, 1993), (Jang et Sun, 1995) et (Yam *et al.*, 2001) en proposent une vision plus détaillée.

Première phase : propagation avant. Lorsque les paramètres prémisses du réseau sont fixés (ceux des fonctions d'appartenance floues de la couche 1), la sortie du système ANFIS est linéaire par rapport aux paramètres conséquents (cf. équations 5 et 6). De fait, pour chaque couple de la base d'apprentissage, une fois les entrées propagées jusqu'à la couche 4, il est possible d'utiliser l'algorithme de l'estimation des moindres carrés afin de déterminer les paramètres conséquents du réseau.

Deuxième phase : propagation arrière. Les paramètres conséquents optimaux étant connus, les prémisses peuvent être obtenus par l'algorithme de descente du gradient simple. À chaque itération d'apprentissage, ces paramètres sont corrigés de la forme suivante :

$$\Delta \alpha = -\eta \times \frac{\partial E}{\partial \alpha}, \quad (7)$$

où η désigne le pas d'apprentissage, α le paramètre à modifier, et $(\partial E / \partial \alpha)$ la composante du gradient de l'erreur de prédiction inhérente au paramètre α , calculé par l'algorithme de rétro-propagation du gradient.

Notons que les deux phases de l'algorithme hybride brièvement décrites ci-avant doivent être réalisées pour chaque ensemble de la base d'apprentissage (tableau 1). Aussi, le calcul du gradient total nécessite le calcul de tous les gradients partiels (en chaque nœud).

	Propagation Avant	Propagation Arrière
Paramètres prémisses	Fixés	Descente du gradient
Paramètres conséquents	Estimation des moindres carrés	Fixés

Tableau 1. Algorithme d'apprentissage hybride

Les grands principes d'inférence et d'apprentissage d'un système prédictif du type ANFIS étant posés, revenons à la problématique de la maîtrise de l'erreur de prédiction formulée dans la partie 2.

4. MAÎTRISE DE L'ERREUR DE PRÉDICTION

4.1. Améliorations potentielles d'un système ANFIS

Différents travaux montrent (cf. section précédente) que le système neuro-flou ANFIS est à même de « capturer » le comportement d'un système rapidement et précisément, et cela, mieux encore que d'autres méthodes, réseaux de neurones (RN) inclus. Ceci étant, dans la pratique, la construction d'un système ANFIS se heurte à des problèmes similaires à ceux rencontrés pour les RN. Aussi, (Rui et El-keib, 1995) permettent de faire un inventaire des « facteurs ayant une forte influence sur la précision et l'efficacité d'un réseau de neurones », inventaire qu'il est aisé d'adapter aux réseaux neuro-flous. Nous nous appuyons sur cela afin d'identifier une piste de travail quant à la maîtrise de l'erreur de prédiction.

4.1.1. Structure du réseau

D'après (Rui et El-keib, 1995), l'expérience montre qu'il n'y a pas de structure de réseau idéale a priori. Celle-ci est à définir au cas par cas selon le problème. Concernant le réseau neuro-flou ANFIS, définir une architecture consiste à choisir le nombre de fonctions d'appartenance, leur type (gaussienne, sigmoïde...) et le nombre de règles floues utilisées. Aussi, un compromis simplicité-généralisation doit être trouvé : il n'est pas concevable de construire un système ANFIS trop complexe, la base d'apprentissage étant, elle, limitée.

4.1.2. Sélection d'entrées

Le choix des variables d'entrées peut avoir une influence importante sur les performances d'un réseau ANFIS. Dans le cas de la prédiction de séries temporelles, (Jang, 1993) propose par exemple de faire une sélection des entrées dans l'historique des données afin de réaliser de bonnes prédictions. (Gouriveau *et al.*, 2007) discutent de l'intérêt d'un pré-traitement des données d'entrée du réseau.

4.1.3. Base d'apprentissage

L'ensemble d'apprentissage disponible impacte bien entendu la qualité de généralisation du système ANFIS. (Wang *et al.* 2004) rappellent qu'il est nécessaire de disposer d'une base d'apprentissage de taille 5 fois supérieure au nombre de paramètres à optimiser (prémises et conséquents). Cependant, au-delà de la taille de cet ensemble d'apprentissage, c'est également la pertinence des données qu'il contient qui se révèle critique.

4.1.4. Algorithme d'apprentissage

(Riedmiller et Braun, 1993) proposent de modifier l'algorithme d'apprentissage afin d'en accélérer la convergence. (Yam et Chow, 2000) et (Wang *et al.*, 2007) proposent d'optimiser l'initialisation des poids d'apprentissage. La modification des pas d'apprentissage peut aussi être envisagée (Sha et Bajie, 2002), (Yoo *et al.*, 2007).

4.1.5. Orientation de travail

De nombreux paramètres sont donc susceptibles d'influer sur les performances d'un réseau neuro-flou de type ANFIS. Parmi tous ceux-ci, nous nous intéresserons plus particulièrement à la modification de la fonction de coût de l'erreur de prédiction utilisée dans l'algorithme de rétro-propagation du gradient. La littérature se montre avare en exemples sur ce thème, en particulier dans le domaine de l'approximation de fonctions. Concernant les problèmes de classification, (Lemaire *et al.*, 2001), (Castellano *et al.*, 2000) ou (Benardos et Vosniakos, 2006) définissent de nouvelles fonctions de coût.

4.2. Éléments théoriques

Afin de cerner l'influence de la fonction de coût utilisée, revenons sur la deuxième phase de l'algorithme d'apprentissage hybride présentée en section 3.2.2. Pour ce faire, nous utilisons l'exemple proposé en figure 4.

4.2.1. Descente du gradient

L'algorithme de rétro-propagation du gradient vise le calcul du gradient de l'erreur totale E afin de corriger les paramètres du réseau. Si i désigne l'itération de calcul en cours, nous avons (extension de l'équation 7) :

$$\alpha_i - \alpha_{i-1} = \Delta\alpha = -\eta \times \nabla E . \quad (8)$$

Rappelons que :

- $x_{i,k}$ désigne la sortie du nœud k de la couche i ,
- α est un paramètre quelconque du réseau,

et adoptons les notations suivantes :

- L est le nombre de couches du réseau,
- P est le nombre total d'échantillons d'apprentissage (couples « entrées/sortie »),
- t désigne l'indice de l'échantillon d'apprentissage,
- d^t est la sortie réelle du réseau désirée, soit la « sortie » pour l'échantillon t .

La fonction de coût classique est la fonction d'erreur quadratique, définie par :

$$E^t = \left(d^t - x_{5,1}^t \right)^2 . \quad (9)$$

Calculons le gradient de l'erreur pour le nœud de sortie :

$$\frac{\partial E^t}{\partial x_{5,1}} = -2 \times \left(d^t - x_{5,1}^t \right) . \quad (10)$$

Pour chacun des nœuds des couches 1 à 4, nous obtenons le gradient de l'erreur par récursivité :

$$\frac{\partial E^t}{\partial x_{l,k}} = \sum_{(l+1,m) \in \text{Succ}(l,k)} \left(\frac{\partial E^t}{\partial x_{l+1,m}} \times \frac{\partial x_{l+1,m}}{\partial x_{l,k}} \right) \quad (11)$$

où $\text{Succ}(l,k)$ désigne l'ensemble des nœuds successeurs du nœud (l,k) .

A ce stade, l'apprentissage porte sur la minimisation du gradient de l'erreur par rapport aux paramètres à modifier. Ceci implique, pour chaque nœud et pour l'ensemble des paramètres α , le calcul du gradient partiel suivant :

$$\frac{\partial E^t}{\partial \alpha} = \frac{\partial E^t}{\partial x_{l,n}} \times \frac{\partial x_{l,n}}{\partial \alpha} \quad (12)$$

Le principe précédent peut ensuite être étendu à l'erreur totale (nécessaire pour l'équation 8). Cette erreur totale est obtenue en considérant l'ensemble des données d'apprentissage :

$$E = \sum_{t=1}^P E^t, \quad (13)$$

et chaque composante ($\partial E / \partial \alpha$) du gradient total (∇E) peut ainsi être déterminée :

$$\frac{\partial E}{\partial \alpha} = \sum_{t=1}^P \left(\frac{\partial E^t}{\partial \alpha} \right) \quad (14)$$

Rappelons que dans la deuxième phase de l'algorithme hybride, les paramètres consécutifs étant fixés (cf. tableau 1), les paramètres à optimiser sont les prémisses, c'est-à-dire les paramètres inhérents aux fonctions d'appartenance floues du réseau (couche 1).

4.2.2. Influence de la fonction de coût

Partant de l'équation 11 et compte tenu des équations 3 à 6, il est possible de démontrer que :

$$\forall x_{l,n}, \quad \frac{\partial E^t}{\partial x_{l,n}} = \frac{\partial E^t}{\partial x_{5,l}} \times g, \quad (15)$$

où g est une fonction ne dépendant pas de la fonction de coût. Il s'agit en fait d'un calcul en cascade de la couche l à la couche 5.

Ensuite, l'équation 12 devient :

$$\forall \alpha, \quad \frac{\partial E^t}{\partial \alpha} = \frac{\partial E^t}{\partial x_{5,l}} \times g \times \frac{\partial x_{l,n}}{\partial \alpha}. \quad (16)$$

Compte tenu que $\frac{\partial x_{l,n}}{\partial \alpha}$ ne dépend pas de la fonction de coût, mais simplement de la structure du réseau, l'équation 16 s'écrit :

$$\forall \alpha, \quad \frac{\partial E^t}{\partial \alpha} = \frac{\partial E^t}{\partial x_{5,l}} \times g_2, \quad (17)$$

avec g_2 indépendant de la fonction de coût.

Au final, dans la descente du gradient, $\frac{\partial E^t}{\partial x_{5,l}}$ est le seul terme dépendant de la fonction de coût.

Revenant à la correction des paramètres du réseau, il apparaît ainsi que cette modification ($\Delta \alpha$) est d'autant plus significative que le gradient partiel par rapport au nœud de sortie l'est : $\frac{\partial E^t}{\partial x_{5,l}}$ (équations 7 ou 8 avec 17).

Par suite, il est aisé de comparer l'influence de deux fonctions de coût dans la modification des poids du réseau et donc, d'agir sur l'efficacité de la phase d'apprentissage d'un système ANFIS.

4.3. Nouvelle fonction de coût

Précisons quelques impératifs auxquels doit répondre une fonction de coût lors de la phase d'apprentissage. Une illustration est proposée en figure 5.

En premier lieu, si l'erreur de prédiction est nulle, la fonction de coût doit l'être aussi, ainsi que le gradient partiel par rapport à la sortie (pas de correction des paramètres) : si $d^t = x_{5,l}^t$, alors $E^t = 0$ et $\frac{\partial E^t}{\partial x_{5,l}} = 0$.

Ensuite, il n'y a pas de raison a priori de distinguer deux erreurs de prédiction de signes opposés mais égales en valeur absolue. Ainsi, la fonction de coût doit avoir pour valeur moyenne 0 et être symétrique par rapport à \vec{O}_y .

La correction des paramètres va dans le sens inverse à la descente du gradient (composante « $-\eta$ » de l'équation 8). Or, le gradient total par rapport aux paramètres étant fonction du gradient partiel par rapport à la sortie (équation 17), il est nécessaire de distinguer les erreurs positives et négatives dans le calcul du gradient partiel par rapport à la sortie afin de modifier les paramètres de façon cohérente. En d'autres termes, il convient de construire une fonction de coût telle que la dérivée par rapport à la sortie soit symétrique par rapport à 0.

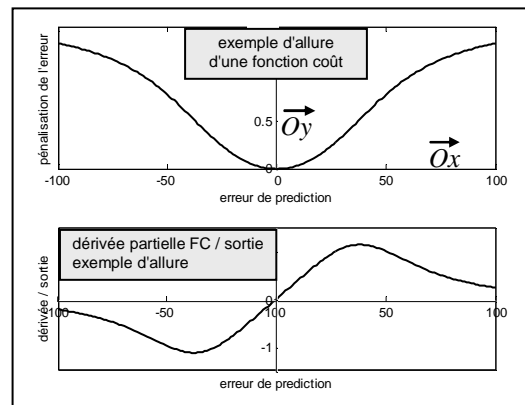


Figure 5. Exemple de fonction coût et du gradient partiel

Afin de stabiliser les erreurs de prédictions, les erreurs importantes doivent être davantage pénalisées que les erreurs faibles. Ceci étant, selon nous, la fonction de coût doit être indépendante de l'ordre de grandeur des données considérées. Ce n'est pas le cas de la fonction coût quadratique qui accorde la même importance à une erreur de 1 unité, quelle que soit l'ordre de grandeur de la sortie d^t recherchée (1, 10, 1000...). De ce point de vue, nous préférons travailler sur la base d'une erreur de prédiction exprimée en pourcentage de l'erreur par rapport à la sortie désirée.

Au final, nous proposons une nouvelle fonction de coût répondant à toutes les exigences répertoriées ci-avant :

$$E = \left[100 \times \frac{d - x_{5,t}}{d} \right]^2, \text{ et,} \quad (18)$$

$$\frac{\partial E}{\partial x_{5,t}} = -2 \times \left(\frac{100}{d} \right)^2 \times (d - x_{5,t}). \quad (19)$$

5. ÉTUDE EXPÉRIMENTALE

Afin d'évaluer l'influence de la nouvelle fonction de coût sur les performances du réseau, nous avons réalisé une campagne de tests. Les conditions de simulations et les principaux résultats de ces tests sont rapportés ci-après.

5.1. Benchmark

Nous avons utilisé la série temporelle de Mackey-Glass pour valider l'effet de la nouvelle fonction de coût proposée. Cette série de données est un benchmark largement utilisé. La série de Mackey-Glass est chaotique, donc non périodique et non convergente. Aussi, être à même de réaliser des prédictions sur un tel signal est de bonne augure en regard de notre objectif applicatif final (le pronostic de défaillances) : les systèmes réels sont complexes et de comportement généralement non-stationnaire et non-linéaire ce qui rend difficile une étape de modélisation. Travailler avec cette série de données constitue donc un premier pas vers la spécification d'un système de pronostic capable de prendre en compte la dynamique des équipements réels.

5.2. Simulations

5.2.1. Modèles de simulation

Nous avons réalisé les tests sur deux architectures de prédiction distinctes (figure 6).

Le premier modèle, nommé ici « ANFIS1 », correspond au système neuro-flou ANFIS classique. Les entrées de ce système sont directement extraites de la série de données. En notant $x(t)$ la valeur de la série de données à la date t , ANFIS1 se base sur 4 entrées $\{x(t-3), x(t-2), x(t-1), x(t)\}$ pour prédire la sortie $\hat{x}(t+r)$: prédiction à un horizon temporel r .

L'architecture du modèle de prédiction (ANFIS2) est plus élaborée : nous construisons un système prédictif inspiré des modèles en cascade proposés par (Sisman-Yilmaz *et al.*, 2004) composé de deux systèmes ANFIS en série, le second devant prendre en compte l'erreur réalisée par le premier. Le premier ANFIS de ce système a ainsi pour entrées $\{x(t-3), x(t-2), x(t-1), x(t)\}$ et réalise une prédiction à $t+r/2$, le second a pour entrées les sorties du premier et effectue une nouvelle prédiction à $t+r/2$. L'ensemble permet de travailler sur un horizon temporel r . L'apprentissage de chacun des modèles est indépendant (2 apprentissages). Notons que (Sisman-Yilmaz *et al.*, 2004) proposent d'utiliser ce type de structure afin de tenir compte des relations temporelles entre variables sans pour autant construire des modèles plus complexes tels que les réseaux récurrents.

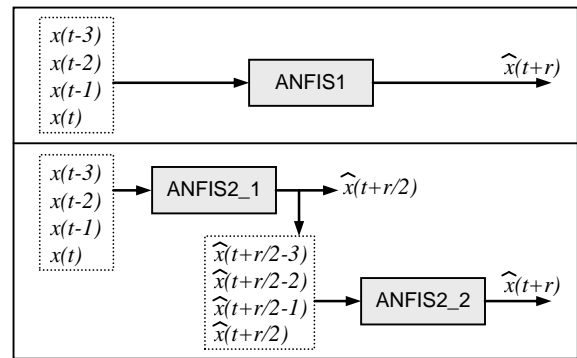


Figure 6. Deux modèles d'utilisation d'ANFIS

5.2.2. Conditions de test

Nous récapitulons dans le tableau 2 les conditions de tests retenues.

Système d'Inférence floue	
Nombre d'entrées	4
Fonctions d'appartenance	2 fonctions par entrée de type Gaussiennes
Nombre de paramètres	96 (16 prémisses et 80 conséquents)
Apprentissage et test	
Nombre de données	1000
Ensembles apprentissage / test	500 / 500, 250 / 750, 100 / 900
Algorithme d'apprentissage	hybride : LSE + Descente du Gradient
Horizons de prédiction (r)	« $t+10$ », « $t+20$ », « $t+50$ », « $t+100$ »

Tableau 2. Conditions de test

5.2.3. Indicateurs de prédiction

Pour juger des résultats, les indicateurs utilisés sont l'erreur quadratique moyenne (EQM) et la distribution de l'erreur de prédiction, exprimée en pourcentage afin de s'affranchir de l'ordre de grandeur de la série de données. Pour ce deuxième critère, nous distinguons l'erreur de prédiction moyenne et le maximum de l'erreur. Si K désigne le nombre de tests réalisés (cf. tableau 2), l'erreur quadratique moyenne est :

$$EQM = \sqrt{\frac{1}{K} \times \sum_{k=1}^{k=K} (x_k - \hat{x}_k)^2} \quad (20)$$

La distribution de l'erreur est obtenue en déterminant l'erreur de prédiction en pourcentage (*EPP*) pour chaque test.

$$\forall k \in [1, K], EPP_k = \frac{x_k - \hat{x}_k}{x_k} \times 100 \quad (21)$$

L'erreur de prédiction en pourcentage moyenne (*EPPM*) et maximum (*maxEPP*) découlent de ce calcul.

5.3. Résultats de simulation

Nous synthétisons dans le tableaux 3 les résultats de simulation sur les modèles ANFIS1 et ANFIS2.

P=500 / K=500		FC quadratique		Nouvelle FC	
		ANFIS1	ANFIS2	ANFIS1	ANFIS2
t+10	EQM	0,0512	0,0286	0,0528	0,0298
	EPPM	4,57 %	2,25 %	4,78 %	2,36 %
	maxEPP	37,62 %	30,17 %	41,27 %	29,61 %
t+20	EQM	0,0215	0,0957	0,0221	0,1021
	EPPM	18,9 %	8,33 %	1,96 %	9,03 %
	maxEPP	6,35 %	43,72 %	6,74 %	57,05 %
t+50	EQM	0,1024	0,0546	0,1068	0,0578
	EPPM	9,79 %	4,87 %	10,27 %	5,09 %
	maxEPP	57,62 %	30,16 %	59,92 %	31,99 %
t+100	EQM	0,1027	0,1035	0,1062	0,1083
	EPPM	10,29 %	9,85 %	10,64 %	10,33 %
	maxEPP	45,67 %	69,96 %	48,66 %	55,72 %
P=250 / K=750		FC quadratique		Nouvelle FC	
		ANFIS1	ANFIS2	ANFIS1	ANFIS2
t+10	EQM	0,0549	0,0321	0,0566	0,032
	EPPM	5,04 %	2,7 %	5,23 %	2,73 %
	maxEPP	27,23 %	25,42 %	25,53 %	27,15 %
t+20	EQM	0,0231	0,1365	0,0237	0,1303
	EPPM	2,06 %	1,187 %	2,13 %	12,06 %
	maxEPP	7,63 %	229,6 %	8,57 %	181,5 %
t+50	EQM	0,1084	0,0661	0,1127	0,0643
	EPPM	10,26 %	5,36 %	10,71 %	5,56 %
	maxEPP	59,19 %	47,74 %	58,88 %	35,87 %
t+100	EQM	0,1088	0,1214	0,1120	0,1188
	EPPM	10,46 %	11,25 %	10,82 %	11,5 %
	maxEPP	48,28 %	79,57 %	47,91 %	53,3 %
P=100 / K=900		FC quadratique		Nouvelle FC	
		ANFIS1	ANFIS2	ANFIS1	ANFIS2
t+10	EQM	0,1213	0,244	0,1236	0,0747
	EPPM	9,97 %	11,18 %	10,19 %	5,81 %
	maxEPP	53,69 %	255,4 %	54,31 %	73,64 %
t+20	EQM	0,0603	0,2907	0,0607	0,3604
	EPPM	5,13 %	21,71 %	5,17 %	25,03 %
	maxEPP	30,94 %	156,4 %	31,31 %	272,3 %
t+50	EQM	0,1677	0,1376	0,1667	0,1459
	EPPM	15,39 %	11,52 %	15,37 %	11,66 %
	maxEPP	116,2 %	136,3 %	114,4 %	60,98 %
t+100	EQM	0,1528	0,3152	0,1527	0,323
	EPPM	12,52 %	22,25 %	12,61 %	20,46 %
	maxEPP	70,93 %	197,5 %	71,73 %	266,3 %

Tableau 3. Résultats de simulation

5.4. Discussion

5.4.1. ANFIS1

Nous donnons en figure 7 les courbes comparatives entre la fonction coût classique et la nouvelle FC pour le modèle ANFIS1. Il apparaît que les performances de la nouvelle FC sont du même ordre de grandeur que celles de l'ancienne et cela, quels que soient l'horizon de prédiction et la taille de la base d'apprentissage. La nouvelle FC ne donne ainsi pas de meilleurs résultats avec le modèle ANFIS1. Ceci étant, elle ne dégrade pas non plus les performances de prédiction.

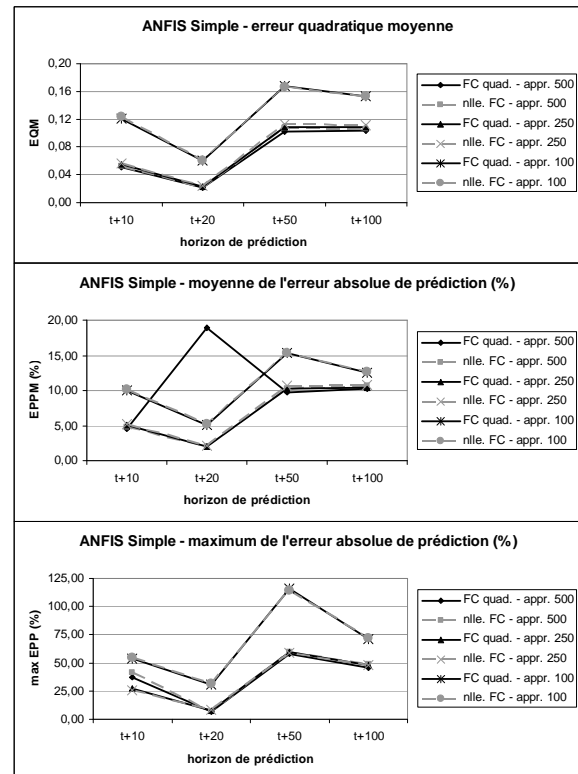


Figure 7. Performances des fonctions de coût quadratique et modifiée avec le modèle ANFIS1

5.4.2. ANFIS2

Si l'on s'en tient à l'écart quadratique moyen (*EQM*) et à l'erreur moyenne de prédiction en pourcentage (*EPPM*), la nouvelle fonction de coût ne semble pas donner de résultats meilleurs de manière significative (figure 8). Mais un examen plus fin de la distribution de l'erreur de prédiction permet de compléter ce constat.

Pour un ensemble d'apprentissage de taille 100, les effets de la nouvelle fonction de coût se font clairement sentir (figure 9). D'abord, son utilisation permet de concentrer les courbes de distribution de l'erreur en fonction de l'horizon de prédiction, notamment à moyen terme (figure 9 – partie basse). Ensuite, pour un horizon donné, la nouvelle FC permet de réduire la dispersion de l'erreur. Au final, les prédictions réalisées ne sont certes pas beaucoup plus précises qu'avec la fonction de coût quadratique mais la qualité (stabilité) du modèle prédictif tend à être indépendante de l'horizon de prédiction.

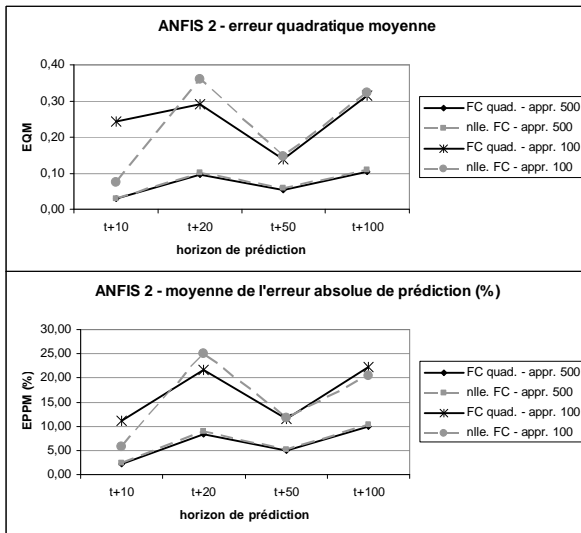


Figure 8. Performances des fonctions de coût quadratique et modifiée avec le modèle ANFIS2

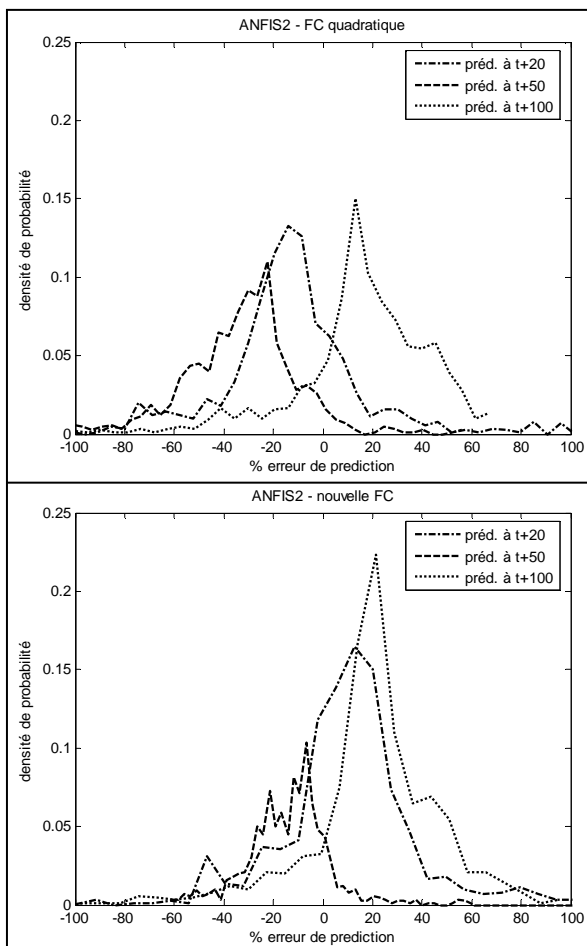


Figure 9. ANFIS2 - distribution de l'erreur de prédiction pour une base d'apprentissage de taille 100

5.4.3. Synthèse

Si la fonction de coût proposée ne montre pas encore tous les résultats attendus, certains « signes » se révèlent prometteurs quant à la problématique de la maîtrise de l'erreur de prédiction. Soulignons également que les aspects positifs obtenus l'ont été avec une base

d'apprentissage de taille 100. Ce résultat est très encourageant : normalement, la base d'apprentissage d'un système ANFIS doit être de taille 5 fois supérieure au nombre de paramètres à optimiser, élément souvent rédhibitoire dans les applications réelles. Pourtant nous avons ici 96 paramètres dans chacun des systèmes ANFIS, c'est-à-dire 192 pour le modèle 2 complet.

6. CONCLUSION

L'activité de pronostic devient un élément clef dans la recherche de stratégies de maintenance industrielle. Il existe cependant encore peu de systèmes réels permettant de supporter ce processus. Aussi, le travail partiellement rapporté ici traite globalement de la spécification et du développement d'un système de pronostic de défaillances. De ce point de vue, beaucoup de développements visant la proposition de méthodes de prévision existent dans la littérature. La majorité d'entre elles portent sur la construction de modèles capables de minimiser l'erreur de prédiction d'une situation future. Cependant, peu traitent de la maîtrise de cette erreur. C'est ce qui fait l'objet de ce papier.

Afin de restituer le travail, nous avons, dans un premier temps, positionné l'activité de pronostic dans le cadre de la maintenance industrielle, et formulé le problème de la maîtrise de l'erreur de prédiction. Sur cette base et en cohérence avec les contraintes inhérentes aux systèmes industriels (non linéarité, beaucoup de données...), nous avons proposé d'utiliser les réseaux neuro-flous pour supporter l'activité de prédiction et plus particulièrement l'architecture ANFIS. Nous avons ensuite étudié les pistes permettant de maîtriser l'erreur de prédiction d'un tel système, notamment lors de la phase d'apprentissage. A ce niveau, nous avons décrits les éléments théoriques nécessaires à cette analyse, et proposé une nouvelle fonction de coût visant l'obtention d'un compromis entre approximations et généralisation. Nous avons finalement réalisé une étude expérimentale afin d'évaluer l'influence de la nouvelle fonction de coût sur les performances du réseau prédictif. Sans être totalement satisfaisante en regard de la problématique de stabilité de l'erreur de prédiction, la modification proposée permet de diminuer la dispersion de l'erreur de prédiction à moyen terme, et cela, avec une base d'apprentissage réduite et sans pour autant augmenter la complexité du système prédictif.

Outre l'application à un cas réel, les travaux sont étendus suivant quatre axes complémentaires. D'abord, nous tâchons de conforter les fondements théoriques de notre proposition et de fournir un guide de choix de fonction de coût. Nous étudions également les passerelles pouvant être exploitées entre les réseaux de neurones récurrents et les systèmes neuro-flou du type ANFIS. Ensuite, nous visons la proposition d'un système prédictif pour lequel l'erreur de prédiction tolérée serait imposée par l'utilisateur. Enfin, nous tâchons de construire un modèle multi-variables (la surveillance des systèmes complexes ne peut se baser sur le suivi d'un unique critère).

Notons finalement que, d'un point de vue applicatif, les développements s'accordent avec les principales préoccupations technologiques des industriels (système temps réel, complexe et de comportement non linéaire, génération de connaissances pour un meilleur retour d'expériences...). Aussi, ces travaux sont menés dans la perspective de construire un module de pronostic adaptable aux cas par cas.

REFERENCES

- Benardos, P.G., and G.C. Vosniakos, 2006. Optimizing feedforward artificial neural network architecture. *Engineering Applications of Artificial Intelligence*, 20(3), p. 365-382.
- Castellano, G., A.M. Fanelli, and C. Mencar, 2000. A new Empirical Risk Functional for a Neuro-Fuzzy Classifier, *Proc. of the European Symp. of Intelligent Techniques*, Ahachen, Germany, p. 429-436.
- Ciarapica, F.E., and G. Giacchetta, 2006. Managing the condition-based maintenance of a combined-cycle power plant: an approach using soft computing techniques, *Journal of Loss Prevention in the Process Industries*, 19, p. 316-325.
- De Gooijer, J.G., and R.J. Hyndman, 2006. 25 years of time series forecasting. *International Journal of Forecasting*, 22, p. 443-473.
- Dragomir, O., R. Gouriveau, N. Zerhouni, and F. Dragomir 2007. Framework for a distributed and hybrid prognostic system, *Proceedings of the 4th IFAC Conference on Management and Control of Production and Logistics*, Sibiu, Roumanie.
- Gouriveau, R., M. El-koujok, and N. Zerhouni, 2007. Spécification d'un système neuro-flou de prédiction de défaillances à moyen terme. *LFA : Rencontres francophones sur la Logique Floue et ses Applications*, Nîmes.
- ISO 13381-1, 2004. *Condition monitoring and diagnostics of machines – prognostics – Part 1: General guidelines*. International Standard, ISO.
- Lung, B., G. Morel, and J.B. Leger, 2003. Proactive maintenance strategy for harbour crane operation improvement. *Robotica*, 21, p. 313-324.
- Jang, J.S.-R., 1993. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. Syst., Man, and Cybern.*, 23, p. 665-685.
- Jang, J.S.-R., and C.T. Sun, 1995. Neuro-fuzzy modeling and control. *Proceedings of IEEE*, 83(3), p. 378-406.
- Jardine, A.K.S., D. Lin, and D. Banjevic, 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechan. Systems and Signal Processing*, 20, p. 1483-1510.
- Lemaire, V., O. Bernier, D. Collobert, and F. Clerot, 2001. Une nouvelle fonction de coût régularisante dans les réseaux de neurones artificiels. *Revue d'Intelligence Artificielle*, X.
- Li, C., and K.H. Cheng, 2007. Recurrent neuro-fuzzy hybrid-learning approach to accurate system modeling. *Fuzzy Sets and Systems*, 158, p. 194-212.
- Lin, D., and V. Makis, 2003. Recursive filters for a partially observable system subject to random failure, *Advances in Applied Probability*, 35, p. 207-227.
- Muller, A., M.C. Suhner, and B. Lung, 2007. Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. *Reliability Engineering and System Safety*, in Press.
- Nauck, D., and R. Kruse, 1997. A neuro-fuzzy method to learn fuzzy classification rules from data, *Fuzzy Sets and Systems*, 89(3), p. 277-288.
- Riedmiller, M., and M. Braun, 1993. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm, *Proceedings of the IEEE International Conference on Neural Networks*, San Francisco, p. 586-591.
- Rui, Y., and A.A. El-keib, 1995. A Review of ANN-based Short-Term Load Forecasting Models. *Proceedings of the Twenty-Seventh South-eastern Symposium on System Theory*, Starkville, MS, USA, p. 78-82.
- Sha, D., V.B. Bajje, 2002. An on-line hybrid learning algorithm for multilayer perceptron in identification problems, *Computers & Electrical Engineering*, 28(6), p. 587-598.
- Sisman-Yilmaz, N.A., F.N. Alpaslan, and L. Jain, 2004. ANFIS unfolded in time for multivariate time series forecasting. *Neurocomputing*, 61, p. 139-168.
- Vachtsevanos, G., F.L. Lewis, M. Roen, A. Hess and R. Wu, 2006. *Intelligent Fault Diagnosis and Prognostic for Engineering System*, John Wiley and Sons Inc.
- Wang, W.Q., M.F. Goldnaraghi, and F. Ismail, 2004. Prognosis of machine health condition using neuro-fuzzy systems. *Mechanical Systems and Signal Processing*, 18, p. 813-83.
- Wang, C.H., C.H. Kao, and W.H. Lee, 2007. A new interactive model for improving the learning performance of back propagation neural network, *Automation in Construction*, 16(6), p. 745-758.
- Yam, J.Y.F., and T.W.S. Chow, 2000. A weight initialization method for improving training speed in feed forward neural network. *Department of Electronic Engineering*, City Univ. of Hong Kong.
- Yam, R.C.M., P.W. Tse, L. Li, and P. Tu, 2001. Intelligent predictive decision support system for condition-based maintenance. *International Journal of Adv. Manufacturing Technology*, 17, p. 383-391.
- Yoo, S.J., J.B. Park, and Y.H. Choi, 2007. Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates, *Information Sciences*, 177(15), p. 3074-3098.
- Zhang, G., B.E. Patuwo, and M.Y. Hu, 1998. Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting*, 14, p. 35-62.