

Surveillance et diagnostic d'évolutions temporelles par classifieur autoadaptatif

Monitoring and Diagnostic of time series evolutions by autoadaptive classifier

S. Lecoeuche^{1,2}, C. Lurette², G. Mercère^{2,3}, S. Lalot⁴, C. Vasseur²

¹ Département GIP, Ecole des Mines de Douai, 59508 Douai

² Laboratoire I3D, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq

³ EIPC, campus de la Malassise, 62967 Longuenesse Cedex

⁴ LME, Université de Valenciennes et du Hainaut Cambrésis, 59304 VALENCIENNES

lecoeuche@ensm-douai.fr

Résumé

Cet article propose une méthode de surveillance et de diagnostic de systèmes industriels complexes. L'approche présentée concerne la détection et la caractérisation de dérives de fonctionnement. On considère que la phase d'identification du procédé a déjà été effectuée. Les variations du procédé sont alors caractérisées par des évolutions dans le temps des paramètres d'identification. Ainsi, il est possible de surveiller le système en analysant les séries temporelles de ces paramètres par une modélisation et une classification. L'architecture neuronale proposée pour ces tâches dispose de capacités d'apprentissage non supervisé et des propriétés d'auto-adaptation. Les règles d'apprentissage non supervisé ont été conçues de manière à permettre la création de nouveaux prototypes et de nouvelles classes lorsque les données présentées ne sont pas connues. De même, les capacités d'auto-adaptation du réseau permettent de prendre en compte une variation des classes et ainsi effectuer une classification de données non stationnaires. L'application à une base de données de séries temporelles montre de bons résultats : 96% des données sont correctement classées en utilisant un faible nombre de prototypes. Appliqué à un réchauffeur électrique de fluide en circulation puis à un thermorégulateur, notre classifieur effectue la détection et la classification des différents types de dérives liées à l'encrassement et/ou au changement de viscosité.

Mots Clef

Surveillance, diagnostic, classifieur neuronal, apprentissage non supervisé, auto-adaptation, données non stationnaires.

Abstract

This article proposes a method of monitoring and diagnosis of complex industrial systems. The presented approach concerns the detection and the characterization of functioning drifts. It is considered that the phase of identification

of the process has been already carried out. The variations of the process are then characterized by modifications in the time of the parameters of identification. Thus, it is possible to supervise the system by analyzing the time series of these parameters by a modelization and a classification. The proposed neural architecture has unsupervised learning capabilities and auto-adaptation properties. The unsupervised learning rules were conceived so as to allow the creation of new prototypes as well as new classes when the presented data are not known. In the same way, the capabilities of auto-adaptation of the network make it possible to take into account a variation of the classes and thus to carry out a classification of non-stationary data. The application to a time series database shows good results : 96% of the data are correctly classified by using a low number of prototypes. Applied to an electric heater of fluid in circulation then to a "thermoregulator", our classifier carries out the detection and the classification of the various types of drifts related to fouling and/or to the change of viscosity.

Keywords

Monitoring, diagnosis, neural classifier, unsupervised learning, autoadaptation, non-stationary data.

1 Introduction

La surveillance et le diagnostic des systèmes industriels sont délicats et complexes lorsque le procédé à surveiller est influencé par un grand nombre de paramètres. Les dérives et les défauts sont alors difficiles à détecter et à localiser sans utiliser de modèles représentatifs du système réel. Pour ces applications avancées, des modèles analytiques sont donc indispensables. L'obtention de tels modèles peut se faire par le biais de différentes approches. Si les lois physiques du processus étudié sont connues, il est aisé de les utiliser pour construire le modèle. Lorsque la modélisation du procédé est impossible à réaliser, notamment lorsque les lois physiques décrivant le système sont trop complexes ou incomplètes, les méthodes pro-

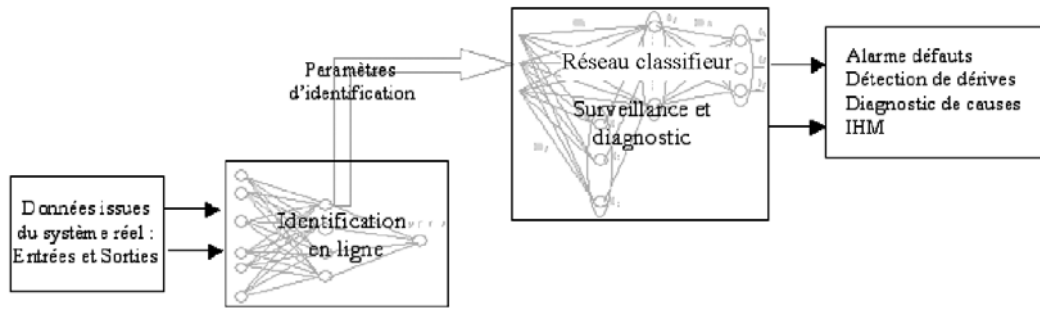


FIG. 1 – Architecture proposé pour la surveillance et le diagnostic de systèmes industriels

posées sont basées sur une représentation paramétrique du système obtenue par analyse des données d'entrée-sortie du procédé. Ces méthodes identifient les paramètres représentatifs du système permettant ainsi de reproduire au mieux son comportement (on parle alors de modèle "boîte noire") [1][2][3].

Dans le cadre d'une application à la surveillance, les techniques d'identification doivent être appliquées durant le fonctionnement du système et les paramètres caractérisant le système doivent être identifiés régulièrement afin de pouvoir suivre et détecter les évolutions du système en cours de surveillance. Les techniques d'identification récursive sont particulièrement adaptées à ce type de problématique. Qu'elles soient basées sur une approche entrée-sortie [4], une approche d'état [5][6] ou utilisant des réseaux de neurones [7], elles permettent de mettre à jour le modèle à chaque nouvelle acquisition. Des applications de surveillance de systèmes thermiques ont montré les capacités des réseaux de neurones à suivre les évolutions du système en utilisant une fenêtre d'apprentissage glissante [8][9]. Les capacités à suivre et à diagnostiquer les dérives du système réel seront fonction du choix de la taille de la fenêtre glissante ainsi que de l'offset utilisé entre chaque fenêtre d'apprentissage.

2 Principe

L'architecture proposée pour la surveillance et le diagnostic de systèmes industriels est représentée figure 1. Elle utilise les paramètres de l'identification du système effectuée en fonctionnement à partir de ses entrées et sorties. La surveillance et le diagnostic sont réalisés par une technique neuronale de classification qui utilise les paramètres d'identification dont les variations caractérisent les évolutions du système réel. Le vecteur d'entrée de ce classifieur est donc construit à partir de ces données. Pour cela, on utilise soit les paramètres identifiés bruts, soit les paramètres physiques déterminés à partir de techniques d'estimation utilisant les paramètres identifiés. Dans tous les cas, les données utilisées pour le réseau sont considérées comme des séries temporelles à surveiller. En effet, les paramètres représentatifs du système pourront dériver, légèrement ou fortement ; pourront se décaler, en positif ou en négatif ; pourront osciller. Comme les ampli-

tudes ne sont pas définies et ne seront jamais connues a priori, il est nécessaire d'utiliser un réseau disposant de capacités d'apprentissage en ligne et non supervisé ainsi que d'une architecture dynamique lui permettant de s'adapter à de nouvelles dérives pas encore rencontrées.

Dans de nombreuses applications de classification, les données à classer sont généralement prétraitées afin d'extraire une population représentative du problème de classification à résoudre pour ensuite effectuer une classification supervisée. Les réseaux de neurones ont montré depuis de nombreuses années de grandes capacités à résoudre ce type de problématique [10]. Un grand nombre d'améliorations ont été développées afin de labelliser l'espace de représentation des données à l'aide de partitions très précises pouvant décrire des formes connexes [11], radiales [12] et elliptiques [13]. Dans le cas de la classification des séries temporelles, l'apparition de nouvelles classes ne peut pas être prédite et il est nécessaire de pouvoir créer de nouvelles partitions de l'espace des données en cours de classification. Des capacités d'apprentissage non supervisé et d'auto-adaptation ont donc été introduites dans les techniques de classification [14][15]. Un grand nombre de réseaux de neurones ont été développés avec des capacités d'apprentissage non supervisé ou avec des capacités d'auto-organisation mais très peu d'entre eux disposent des deux capacités ; c'est-à-dire ayant la capacité d'adapter leur structure en fonction de l'évolution des données d'entrée par la création de nouveaux prototypes et/ou adaptation des prototypes existants mais également la possibilité de créer de nouvelles classes et de modifier la forme de celles existantes. Pour ce dernier type de réseau, les formes complexes des classes sont obtenues en créant un grand nombre de prototype par classe, chacun des prototypes décrivant une petite partition circulaire. Parmi les structures existantes, il est possible de mentionner le réseau CDL, (Cluster Detection and Labeling Network) développé en 1998 par Eltoft et deFigueiredo [16] et ses améliorations présentées par Lurette et Lecoeuche [17]. Plus récemment, le réseau AUDyC (Auto-adaptive and Dynamical Clustering Neural Network) a été proposé [18].

Nous proposons ici l'introduction du réseau AUDyC pour effectuer la surveillance des données temporelles. L'ap-

proche développée consiste à utiliser un classifieur ayant des capacités d'apprentissage non supervisé, lui permettant de classer de nouvelles formes de dérives. Le réseau utilisé possède également des capacités d'auto-adaptation afin de permettre la prise en compte d'une modification possible des classes de fonctionnement durant la vie du système.

3 Présentation du réseau classifieur AUDyC

L'architecture du réseau proposé est inspirée du réseau CDL qui est un réseau feedforward [16]. Le principal avantage de ce réseau est une architecture rendue complètement auto-adaptative grâce à ses règles d'apprentissage utilisées pour la création des prototypes et des classes [18].

3.1 Structure du réseau

La structure de notre réseau de neurones, illustrée en figure 2, consiste en trois couches.

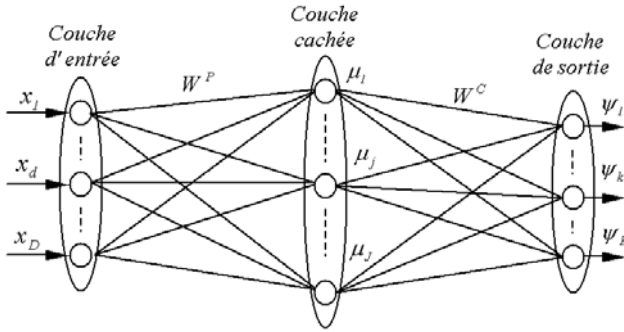


FIG. 2 – Structure du réseau de diagnostic

1. La couche d'entrée comporte autant de neurones qu'il y a de composantes dans le vecteur d'entrée appelé X_i .
2. Les neurones de la couche cachée sont totalement connectés à ceux de la couche d'entrée. Chacun des neurones cachés représente un prototype P_j appartenant à une classe. Leur sortie traduit le degré d'appartenance (eq. 1) du vecteur d'entrée X_i au prototype correspondant.

$$\mu(P_j, X_i) = \exp\left(-\frac{1}{2}((X_i - M_{P_j})^T \Sigma_{P_j}^{-1} (X_i - M_{P_j}))^2\right) \quad (1)$$

La distance de Mahalanobis a été choisie afin de donner au prototype une forme hyper elliptique pour une meilleure définition de leur voisinage.

Ainsi chaque prototype P_j est déterminé selon un vecteur moyen M_{P_j} (position du prototype) et une matrice de covariance pleine Σ_{P_j} (forme du voisinage du prototype).

3. La couche de sortie comporte autant de neurones qu'il y a de classes trouvées. Les connexions entre neurones de couche de sortie et ceux de la couche cachée définissent les appartenances des prototypes aux classes. La sortie de

chaque neurone de cette couche donne le degré d'appartenance (eq. 2) du vecteur d'entrée X_i à la classe correspondante C_k .

$$\Psi(C_k, X_i) = \min(1, \sum_{P_j \in C_k} \mu(P_j, X_i)) \quad (2)$$

3.2 Règles d'apprentissage du réseau

Le principal avantage de ce réseau de neurones est que sa structure, c'est-à-dire le nombre de neurones, n'est pas fixée a priori ou à l'issue d'une phase d'apprentissage supervisé comme pour la plupart des classifieurs. Une phase d'apprentissage spécifique a été conçue permettant ainsi l'adaptation de la structure du réseau et ce, pas seulement en modifiant les poids de connexion entre les neurones mais également en modifiant la taille de la couche cachée et celle de la couche de sortie. La phase d'apprentissage est décomposée en trois principaux étages (Figure 3), celles-ci se succédant de manière itérative.

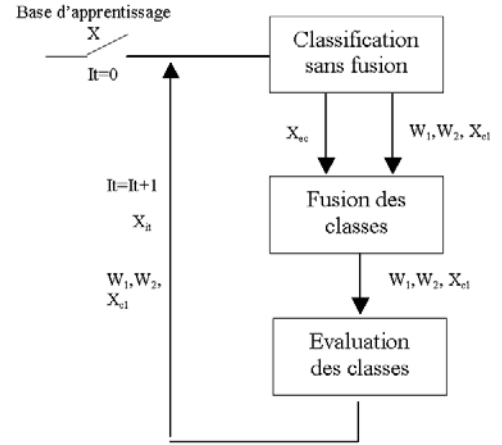


FIG. 3 – Étapes d'apprentissage

Étage de classification sans fusion. Cette première étape est la plus importante en terme d'apprentissage et d'adaptation de l'architecture du réseau. En effet, lors de cette étape, de nouveaux prototypes et de nouvelles classes vont pouvoir être créés lorsque les vecteurs d'entrée sont trop éloignés des exemples connus. Pour chaque vecteur présenté à l'entrée du réseau, son degré d'appartenance est calculé pour chaque prototype connu. Les sorties de neurones de la couche cachée donnent ce degré. En comparant cette valeur avec deux seuils fixés, plusieurs cas peuvent apparaître qui permettront d'aboutir soit à la création d'une nouvelle classe et/ou d'un nouveau prototype, soit à l'adaptation d'un prototype déjà existant. Le tableau 1 détaille les différentes règles d'adaptation.

La figure 4 donne un exemple des différents cas possibles. Sont déjà connus quatre prototypes P_1, P_2, P_3 et P_4 définissant trois classes C_1, C_2, C_3 (P_j^k traduit le prototype j appartenant à la classe k). Quatre nouveaux vec-

Cas 1	Le degré de similarité entre un exemple et n'importe quel prototype connu est inférieur au seuil minimum $\Rightarrow \mu(P_j, X_i) < S_{min}$ pour tous les P_j	L'exemple n'est proche d'aucun prototype. Il est nécessaire de créer un nouveau prototype (un nouveau neurone sur la couche cachée) et une nouvelle classe (un nouveau neurone sur la couche de sortie).
Cas 2	Le degré d'appartenance est supérieur au premier seuil mais inférieur au seuil max pour plusieurs prototypes appartenant à la même classe. $\Rightarrow S_{min} < \mu(P_j, X_i) < S_{max}$ pour $P_j \in C_k$	L'exemple est proche de plusieurs prototypes de la même classe mais pas suffisamment près. Il est nécessaire de créer un nouveau prototype. Ce prototype et les prototypes P_j appartiennent à la même classe k .
Cas 3	Le degré d'appartenance est supérieur aux deux seuils pour plusieurs prototypes appartenant à la même classe. $\Rightarrow S_{max} < (\mu(P_j, X_i))$ pour différents $P_j \in C_k$	L'exemple est associé au prototype P_j le plus proche et à sa classe. Le prototype est adapté en modifiant son vecteur moyen et sa matrice de covariance
Cas 4	Le degré d'appartenance est supérieur au premier seuil S_{min} pour plusieurs prototypes appartenant à différentes classes.	Cette ambiguïté va être analysée lors de l'étape de fusion de la procédure d'apprentissage.

TAB. 1 – Les différents cas traités par l'étape de "Classification sans fusion"

teurs X_1, X_2, X_3 et X_4 sont alors présentés successivement au réseau et en fonction de leur degré d'appartenance différents cas représentés tableau 2 sont possibles.

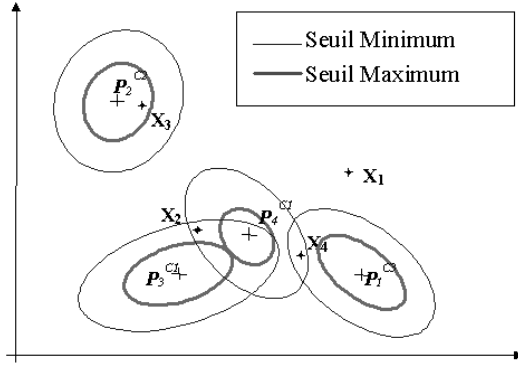


FIG. 4 – Illustration du degré d'appartenance d'un vecteur d'entrée à des prototypes connus

Vecteur d'entrée	Degré d'appartenance à				Alors
	P_1^{C3}	P_2^{C2}	P_3^{C1}	P_4^{C1}	
X_1	ϵ	ϵ	ϵ	ϵ	Nouveau prototype $P_5 = X_1$; Nouvelle classe C_4 ; $X_1 \in X_{cl}$
X_2	ϵ	ϵ	$> \mu_{min}$	$> \mu_{max}$	Nouveau prototype $P_6 = X_2$; P_6 appartient à la classe C_1 ; $X_2 \in X_{cl}$
X_3	ϵ	$> \mu_{max}$	ϵ	ϵ	Adaptation de P_2^{C2} ; X_3 appartient à C_2 ; $X_3 \in X_{cl}$
X_4	$> \mu_{min}$	ϵ	ϵ	$> \mu_{max}$	Nouveau prototype $P_7 = X_4$; C_2 et C_3 fusionnent ; $X_4 \in X_{ec}$

TAB. 2 – Les différents cas illustrés Fig. 4

Dans le cas 3, le prototype auquel a été associé le nouveau vecteur d'entrée sera adapté par cette nouvelle donnée. Cette adaptation concerne le vecteur moyen et la matrice

de covariance du prototype et se fait de manière récursive en utilisant les équations suivantes (eq. 3 et 4).

$$M_{P_j}^i = M_{P_j}^{i-1} + \frac{1}{N}(X_i - X_{i-N+1}) \quad (3)$$

$$\Sigma_{P_j}^i = \Sigma_{P_j}^{i-1} + \Delta X \begin{pmatrix} \frac{1}{N} & \frac{1}{N(N-1)} \\ \frac{1}{N(N-1)} & \frac{N(N-1)}{N(N-1)} \end{pmatrix} \Delta X^T \quad (4)$$

$$\text{avec } \Delta X = [X_i - M_{P_j}^{i-1} \quad X_{i-N+1} - M_{P_j}^{i-1}]$$

Un nombre minimum (N) d'exemples assignés aux prototypes doit être atteint avant d'utiliser ces relations. En deçà de ce seuil, le centre et la matrice de covariance sont adaptés récursivement uniquement par l'ajout de nouvelles données X_i [18].

La représentativité de chacun des prototypes est définie par l'importance de la connexion W_{jk}^C avec sa classe d'appartenance. Ainsi, ce poids est défini par :

$$W_{jk}^C = \frac{\text{Card}(P_j^{C_k})}{N} \quad (5)$$

A la fin de cette première étape, les données classées sont stockées dans χ_{Cl} , les autres données correspondant aux cas ambigus (cas 4) sont stockées dans χ_{amb} .

Étape de fusion des classes. Cet étage regroupe les prototypes et les classes qui sont rapprochés dans l'espace de représentation. Si une ambiguïté a été détectée (cas 4) dans l'étape de classification sans fusion, elle sera résolue par la fusion des différents prototypes ambigus dans une classe unique. Afin d'être insensible au bruit, un seuil n_{amb} est fixé. En fait, la fusion entre 2 classes n'est réalisée que lorsque :

$$\text{Card}(\{X_i \in \chi_{amb} / X_i \in C_{k1} \text{ AND } X_i \in C_{k2}\}) \geq n_{amb} \quad (6)$$

Ainsi la couche de sortie du réseau est modifiée par l'élimination des neurones représentant les classes ambiguës dans le but de n'en laisser qu'un seul par ambiguïté détectée.

Étape d'évaluation des classes. Cette dernière étape est utilisée pour caractériser les prototypes et les classes trouvés durant les étapes précédentes. Les classes ainsi que les prototypes représentant un nombre trop faible ($N_{P_{min}}$ et $N_{C_{min}}$) d'exemples X_i sont éliminés. Ces exemples sont à nouveau identifiés comme non classés (stockés dans χ_{app}) et peuvent être réintroduits à l'entrée du réseau à la prochaine itération.

3.3 Description du cycle d'utilisation

Le réseau AUDyC peut être utilisé sur des données stationnaires ou sur des données non stationnaires en modifiant son cycle d'utilisation.

Dans le cadre de la classification sur des données stationnaires, les données non classées après l'étape d'évaluation (stockées dans χ_{app} et χ_{amb}), sont de nouveau présentées à l'entrée du réseau. Le processus est alors cyclique et le critère d'arrêt est atteint lorsque toutes les données présentées sont classées (ou un pourcentage fixé) ou lorsque les ensembles χ_{app} et de χ_{amb} sont devenus stables.

Dans le cadre de la classification en ligne ou la classification de données non-stationnaires, le processus d'apprentissage n'est plus cyclique. L'apprentissage s'effectue de manière continue : un nouvel ensemble χ_{s+1}^{nswn} issu d'une fenêtre glissante est présenté à l'entrée du réseau à l'issue de la phase d'évaluation (Fig 5).

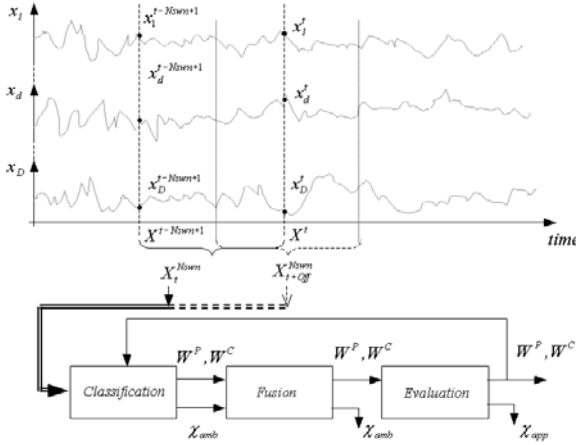


FIG. 5 – Illustrations du cycle d'apprentissage en ligne

4 Applications

4.1 Application à la surveillance de séries temporelles

Pour valider le développement proposé et pour s'assurer de ses capacités à surveiller les paramètres d'un système, le

réseau a été testé sur un benchmark : la base de données "Synthetic Control Chart Time Series" [19]. Cette base de données, décrite figure 6 est construite à partir de différentes courbes (600) représentant 6 classes de dérives. Elle a été choisie car chaque classe peut être associée à une forme type d'évolution de paramètres du modèle et ainsi à ses différentes dérives. Chaque classe est définie par 100 courbes de 60 échantillons.

Normal (zoom)		Cyclique	
Tendance à la hausse		Tendance à la baisse	
Décalage négatif		Décalage positif	

FIG. 6 – Exemples de séries temporelles utilisées

Choix des caractéristiques. Pour permettre leur classification, chaque série temporelle de la base a été caractérisée par un vecteur de représentation à trois composantes : son écart type, la pente de la régression linéaire et l'écart maximum entre les coefficients de modèles ARX locaux. Bien que la moyenne aurait également pu être un moyen aisé de discriminer les 600 séries, il a été choisi de ne pas prendre en compte cette caractéristique. En effet, ceci correspond au fait qu'une évolution à long terme du point de bon fonctionnement est un phénomène normal pour certains systèmes réels.

Résultats. Dans cette première application, et s'agissant de données stationnaires, l'apprentissage a été fait de manière cyclique (voir 3.3). La figure 7 montre le résultat de la classification de la base en utilisant les seuils suivants : $\mu_{min} = 0.3$, $\mu_{max} = 0.6$, $\Sigma_{ini} = 0.05$, $n_{amb} = 3$, $NC_{min} = 7$, $NP_{min} = 10$ et $N_{obs} = \infty$.

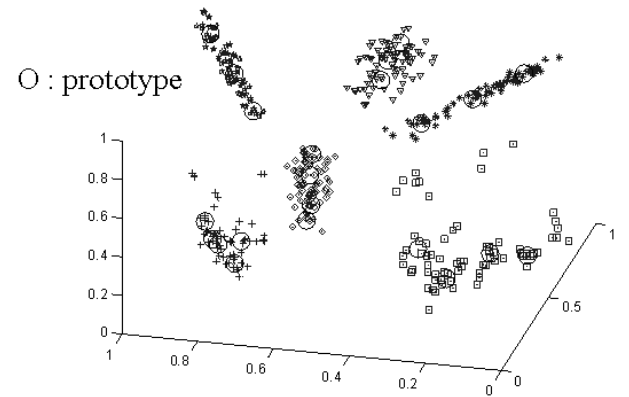


FIG. 7 – Résultat de la classification

Pour avoir une idée de la qualité de la classification, il est nécessaire de donner le pourcentage de données correc-

tement classées et le nombre de prototypes utilisés. Dans le cas présenté, 95.44% des vecteurs d'entrée ont été regroupés correctement dans 6 classes et 22 prototypes ont été obtenus. Tous les exemples mal classés sont des courbes de type "décalage" reconnus comme "tendance". La figure 8 donne quelques exemples.

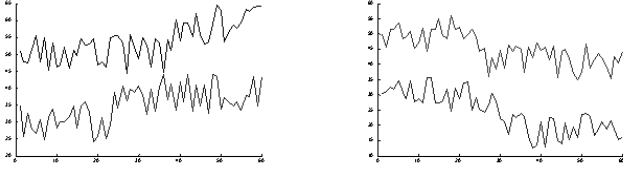


FIG. 8 – Exemples d'erreur de classification

La figure 9 illustre les bénéfices de notre réseau en terme de nombre de prototypes utilisés et de la partition de l'espace obtenue (17 prototypes après avoir simplifié la classification à un problème 2D). Par comparaison, un algorithme de classification supervisé éprouvé et efficace : l'algorithme de "Gaussian RBF Support Vector Machine" [20] donne un nombre plus important de prototypes (31) pour une bonne séparation en 6 classes des données initiales (avec une taille des fonctions radiales = 1.1). A titre indicatif, pour le même type de classification, le réseau CDL classique avait donné 97 prototypes [17].

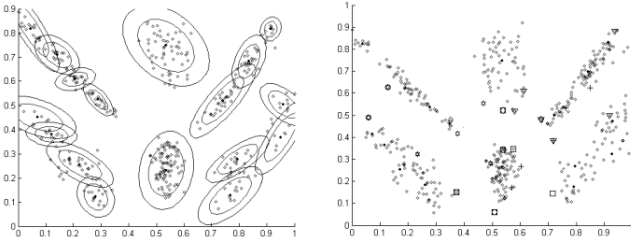


FIG. 9 – Illustrations 2D des prototypes obtenus par notre méthode et par un algorithme SVM

4.2 Surveillance et diagnostic de l'encrassement d'un réchauffeur électrique de fluides en circulation

Ces réchauffeurs sont utilisés dans l'industrie agroalimentaire (chauffage de lait, d'huile) où des dépôts encrassants peuvent se produire sur la surface intérieure du tube. D'une part, ces dépôts peuvent entraîner une altération du goût du produit traité, d'autre part, ils conduisent à une augmentation de la résistance thermique entre la paroi du tube et le fluide. Enfin, une modification de cette résistance thermique peut provenir d'une variation des caractéristiques du fluide et notamment de sa viscosité. La technique de surveillance présentée ci-avant va permettre de discerner la cause de la variation de la résistance thermique. La fi-

gure 10 donne une représentation schématique du système expérimental.

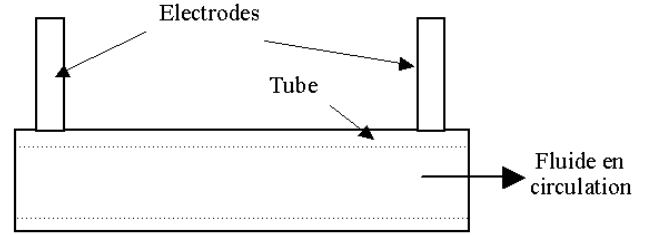


FIG. 10 – Schéma de principe du réchauffeur étudié

L'étude est basée sur un réchauffeur expérimental dont les caractéristiques sont résumées dans le tableau 3.

Tube		Fluide	
ø Intérieur	14 mm	Vitesse nominale	0.8 m/s
ø Extérieur	16 mm	Température l'entrée à	20° C
Longueur	1 m	Coefficient d'échange	Corrélation de Colburn

TAB. 3 – Principales caractéristiques du réchauffeur

Le fonctionnement en régime transitoire est régi par une équation de fonctionnement qui peut s'écrire sous une forme adimensionnée (eq.7). On trouvera plus de détails dans [9][21].

$$(1 + R) \frac{\partial \theta}{\partial t} + \frac{\partial \theta}{\partial x^+} + S. \frac{\partial^2 \theta}{\partial t^2} + S. \frac{\partial^2 \theta}{\partial x^+ \partial t^+} = 1 \quad (7)$$

La phase d'identification est réalisée selon un modèle OE (les entrées sont la puissance dissipée et le débit du fluide, la sortie est la température de sortie du fluide). Pour permettre la surveillance du système les paramètres optimaux de la fenêtre glissante d'identification sont les suivants : largeur de 90 échantillons, période d'échantillonnage d'une seconde, offset d'une largeur de fenêtre. En utilisant de telles fenêtres, on obtient, pour les 6 paramètres identifiés, des séries temporelles semblables à celles étudiées à la section 4.1 ci-dessus. Les quatre dérives étudiées sont : un encrassement limité, une légère fluctuation de la viscosité autour du point nominal, deux combinaisons d'encrassement et de viscosité (faible encrassement et forte variation de viscosité ou fort encrassement et faible variation de viscosité). Les deux premières dérives doivent pouvoir être considérées comme faisant partie de la classe "bon fonctionnement", les deux autres, aux effets cumulatifs, doivent pouvoir appartenir à des classes différentes. En choisissant correctement les seuils μ_{min} et μ_{max} du réseau classifieur ainsi que la taille initiale des prototypes, on montre que l'architecture est capable de trouver les classes de mauvais fonctionnement tout en tolérant des variations autour du point nominal de fonctionnement (figure 11).

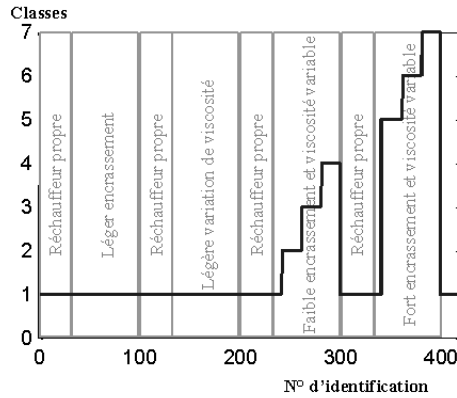


FIG. 11 – Résultats de la surveillance du réchauffeur

4.3 Détection de dérives de fonctionnement d'un thermorégulateur

Cette étude est extraite d'un projet, financé par l'ADEME (Agence de l'Environnement et de la Maîtrise de l'Energie), visant à proposer de nouveaux moyens pour la surveillance de thermorégulateurs industriels. Chaque thermorégulateur est composé principalement d'un réchauffeur électrique et d'un échangeur thermique lui permettant de générer ou de récupérer de la chaleur (Fig. 12). Grâce à une pompe qui assure la circulation du fluide, le réchauffeur électrique assure la montée en température de ce fluide à l'aide d'une résistance chauffante blindée. L'échangeur est utilisé pour assurer un refroidissement possible grâce à un circuit d'eau secondaire. Pour les autres composants, le filtre assure que les particules ne passent pas à travers la pompe tandis que le vase d'expansion permet de compenser l'augmentation du volume et la dilatation subie par le fluide entre la température ambiante et la température d'utilisation.

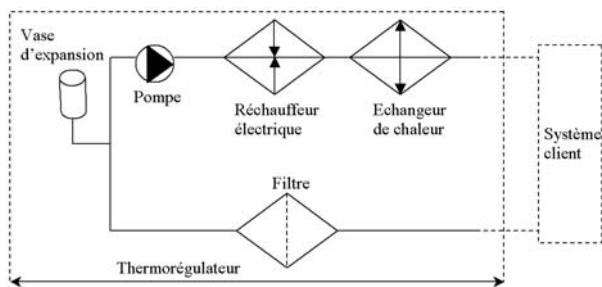


FIG. 12 – Principe de l'installation

L'objet du projet est de permettre la détection des principales dérives de fonctionnement des thermorégulateurs : encrassement de l'échangeur et du réchauffeur électrique, fuites de canalisation, problèmes liés à la pompe (cavitation), etc... Nous ne présenterons ici que la première partie de l'étude qui a consisté à détecter des dérives de fonction-

nement subies par le réchauffeur notamment celles liées à une variation du débit du circuit liée soit à une dérive de fonctionnement de la pompe soit à une diminution du débit liée à un encrassement.

Les différentes simulations suivantes ont pour but d'évaluer les performances de notre système de surveillance basé sur l'AUDyC : capacités à suivre les modifications apportées par les différents états de fonctionnement et d'outils de détection : aptitudes à détecter d'éventuelles anomalies. Il s'agit de simulation dans le sens où une panne a été provoquée sur le système réel. Les données sont quant à elles prélevées directement sur le système. Le réseau AUDyC a été utilisé avec le cycle dédié à la classification de données non-stationnaires.

Simulation n°1 : Dérive lente d'observations. On peut expliquer ce cas par exemple, lors de l'encrassement progressif de l'échangeur qui consiste à passer très lentement d'un état de bon fonctionnement à celui de mauvais fonctionnement. Cette simulation a été réalisée en considérant une diminution du débit du fluide chaud de 2 m³/h à 1m³/h, suivant une variation très lente sur une période de 60 heures.

Le résultat de cette variation du débit est donné sur la figure suivante (Fig.13), représentant l'évolution des données dans le temps. Les données présentées sont issues d'une analyse ACP permettant d'obtenir les axes discriminants à partir de l'ensemble des mesures disponibles sur le réchauffeur (débit, pression et température d'entrée, pression et température de sortie, puissance) et des paramètres identifiés.

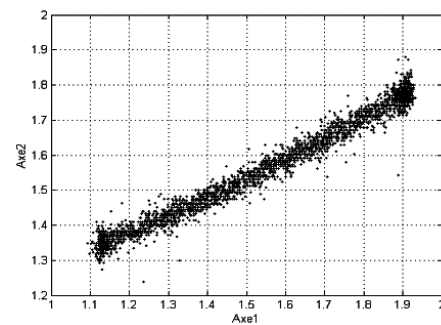


FIG. 13 – Diminution lente du débit du fluide chaud

Les données ont été extraites à l'aide d'une fenêtre glissante puis appliquées à l'entrée du réseau AUDyC initialisé avec les paramètres suivants : $\mu_{min} = 0.2$, $\mu_{max} = 0.3$, $\Sigma_{ini} = 0.05$, $n_{amb} = 3$, $NC_{min} = 15$, $NP_{min} = 15$, $N_{fen} = 1000$ et $N_{obs} = 800$.

Puisque les états de fonctionnement évoluent d'une façon très lente et avec une très petite dispersion par rapport aux anciens états, l'AUDyC adapte le mode de fonctionnement à chaque apparition d'une fenêtre. Avec le processus d'oubli, les anciennes observations sont négligées et les nouvelles enrichissent le mode de fonctionnement courant. En

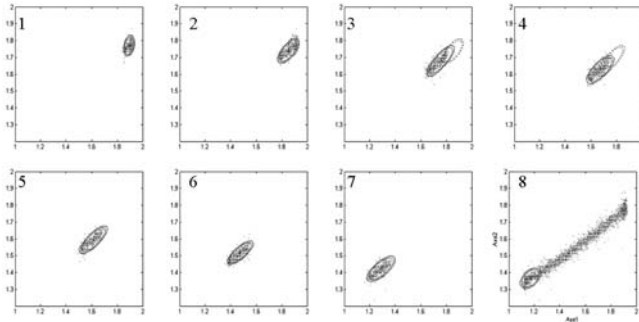


FIG. 14 – Présentation de l'évolution progressive du mode de fonctionnement par l'AUDyC. Instant d'apparition k correspondant aux numéros de fenêtres (1 ; 110 ; 129 ; 317 ; 370 ; 474 ; 620 et 731)

se basant sur l'analyse des évolutions des paramètres du modèle obtenu, la détection d'une dérive peut être aisément obtenue. Ainsi, et par exemple, l'outil EWMA appliqué sur la mesure de distance entre modèles successifs permet de détecter cette dérive.

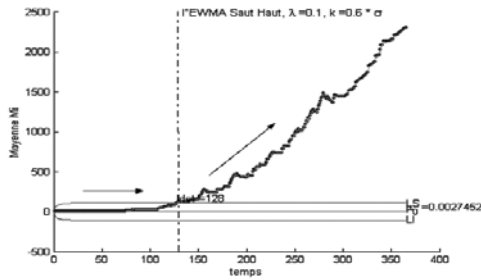


FIG. 15 – Application de l'EWMA sur la distance du centre évolutif

Simulation n°2 : Dérive rapide d'observations. Cette deuxième simulation concerne également la diminution du débit réalisée en agissant sur la pompe. Cette expérience est similaire à la première où une variation progressive de débit de 2 m³/h à 1 m³/h a été provoquée mais cette fois dans des délais beaucoup plus brefs. La figure suivante représente l'évolution des données sur deux heures successives.

Pour effectuer la modélisation du mode de fonctionnement normal du système et pour prendre en compte les différentes évolutions subies durant cette simulation, le réseau a été initialisé avec des paramètres identiques à ceux de la première simulation.

En analysant en ligne les fenêtres présentées par l'AUDyC (Fig.17), on remarque que les observations évoluent très "rapidement" à partir du mode de fonctionnement, dont les paramètres n'ont pas le temps d'être adaptés avec les nouvelles données. Des prototypes sont donc créés au fur et à mesure de l'arrivée d'états, à des instants successifs, entre le premier mode et une zone de l'espace de représentation

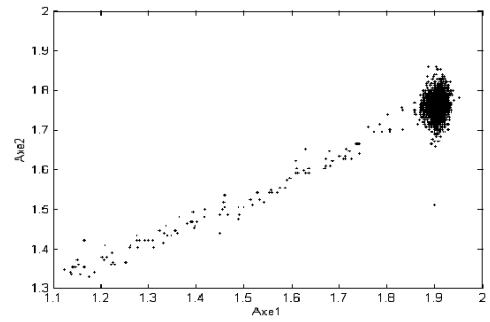


FIG. 16 – Dérive d'observations, débit du fluide chaud

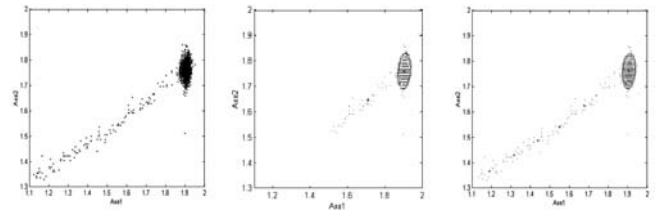


FIG. 17 – Présentation de quelques fenêtres d'observations par l'AUDyC

inconnue. Ces prototypes successifs n'ont pas le temps de modéliser d'autres modes de fonctionnement car ils sont éliminés au fur et à mesure par la phase d'évaluation (effectif ne dépassant pas Np_{min}).

A partir de la modélisation, la détection peut être envisagée via l'analyse des degrés d'appartenance des dernières observations de la fenêtre au mode de fonctionnement. La variation de ces degrés est illustrée par la figure 18. En utilisant le test du CuSum, le saut du degré d'appartenance a été détecté vers le bas à partir de la 28^{ème} observation. Ces dernières sont très rapprochées l'une de l'autre.

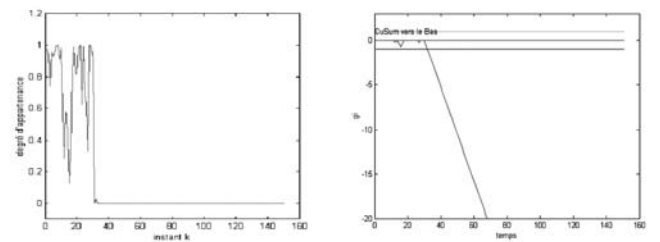


FIG. 18 – Variation du degré d'appartenance des observations dérivant du mode et application du CuSum sur le signal ($h=0.8$ et $k=0.5$).

Ainsi, ayant réalisé ces deux simulations, on met en évidence l'utilité de l'AUDyC et d'un couplage avec les outils classiques de détection. En effet, avec le suivi d'évolution dynamique des observations et l'apprentissage en ligne, nous avons pu extraire des connaissances sur le

comportement du système et appliquer des outils sur les signaux prélevés afin de juger la nature de l'évolution ou de l'anomalie.

5 Conclusion

Dans cet article, une nouvelle architecture neuronale a été appliquée à la surveillance et le diagnostic de séries temporelles. Les principaux avantages sont de permettre un apprentissage non supervisé et d'adapter la classification réalisée en fonction de l'évolution des données. Pour se faire, une nouvelle règle d'apprentissage a été présentée permettant de créer de nouveaux prototypes et de nouvelles classes mais également d'adapter ceux et celles existants. Les améliorations proposées permettent également une meilleure

partition de l'espace de représentation des données grâce à l'utilisation de prototypes définis à l'aide d'une matrice de covariance pleine. Il a été montré sur un exemple que le nombre de prototypes utilisés pour effectuer une classification est également faible en comparaison avec d'autres techniques performantes. Enfin, la capacité à adapter la structure (prototypes et classes) quand de nouvelles données apparaissent, autorise une utilisation en ligne. Aucune base de données initiale n'est nécessaire et le classifieur a la capacité d'effectuer une classification même avec l'apparition de données non stationnaires.

Références

- [1] P. Borne, G. Dauphin-Tanguy, J. Richard, F. Rotella, and I. Zambettakis, *Modélisation et identification des processus, tome 2*. Editions Technip, Paris, 1992.
- [2] L. Ljung, *System Identification : Theory for the User*, 2nd ed. Prentice Hall, 1999.
- [3] I. D. Landau and A. Besançon-Voda, *Identification des systèmes*. Systèmes automatisés IC2, Information Commande Communication, Hermès, 2001.
- [4] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. The MIT Press, 1983.
- [5] T. Gustafsson, "Recursive system identification using instrumental variable subspace tracking," in *11th IFAC Symposium on System Identification*, IFAC, Ed., Fukuoka, Japan, 1997.
- [6] G. Mercère and S. Lecoeuche, "A new recursive method for subspace identification of noisy systems : Eivpm," in *13th IFAC symposium on System Identification*, SYSID2003, Ed., Rooterdam, The Netherlands, August 27-29 2003.
- [7] J. Sjöberg, "Non-linear system identification with neural networks," Ph.D. dissertation, Dept. of Electrical Engineering Linköping University S-581 83 Linköping Sweden, 1995.
- [8] S. Lecoeuche, C. Lurette, and S. Lalot, "Online identification and diagnostic method applied to thermal systems," in *International conference on Engineering Applications of Neural Networks - EANN2000*, Kingston, England, July 17-19 2000, pp. 155-162.
- [9] S. Lalot and S. Lecoeuche, "Online fouling detection in electrical circulation heaters using neural networks," *International Journal of Heat and Mass Transfer*, vol. 46, pp. 2445-2457, juin 2003.
- [10] C. Looney, *Pattern recognition using neural networks : Theory and algorithms for engineers and scientists*. Oxford University Press, 1997.
- [11] L. Gupta, M. McAvoy, and J. Phegley, "Classification of temporal sequences via prediction using the simple recurrent neural network," *Pattern Recognition*, vol. 33, no. 10, pp. 1759-1770, 2000.
- [12] J. Gomm and D. Yu, "Selecting radial basis function network centers with recursive orthogonal least squares training," *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 306-314, 2000.
- [13] M. Mak and S. Kung, "Estimation of elliptical basis function parameters by the em algorithm with application to speaker verification," *IEEE Trans. on Neural Networks*, vol. 11, no. 4, pp. 961-969, 2000.
- [14] J. Mao and A. Jain, "A self organizing network for hyper ellipsoïdal clustering (hec)," *IEEE Transactions on Neural Networks*, vol. 11, no. 4, pp. 16-29, 1996.
- [15] N. Zheng, Z. Zhang, G. Shi, and Y. Qiao, "Self-creating and adaptive learning of RBF networks," in *Proc. of the Intern. Joint Confer. on Neural Networks*, July 1999.
- [16] T. Eltoft and R. deFigueiredo, "A new neural network for cluster-detection-and-labeling," *IEEE Trans. on Neural Networks*, vol. 9, no. 1, pp. 1021-1035, 1998.
- [17] C. Lurette and S. Lecoeuche, "Improvement of cluster detection and labeling neural network by introducing elliptical basis functions," in *International Conference on Artificial Neural Networks - ICANN 01*, G. Dorffner, H. Bischof, and K. Hornik, Eds. Vienna, Austria : Springer, August 21-25 2001, pp. 196-202.
- [18] S. Lecoeuche and C. Lurette, "Auto-adaptive and dynamical clustering neural network," in *13th International Conference on Artificial Neural Networks*, ICANN03, Ed., Istanbul, Turkey, June 26-29 2003.
- [19] D. T. Pham and S. J. Oh, "Identification of plant inverse dynamics using neural networks," *Artificial Intelligence in Engineering*, vol. 13, pp. 309-320, 1999.
- [20] S. Gunn, "Support Vector Machine for classification and regression," University of Southampton, 1998.
- [21] S. Lalot and S. Lecoeuche, "Identification en ligne et hors ligne de réchauffeurs électriques," *Transactions of the CSME*, vol. 25, no. 3-4, pp. 277-296, 2001.