



HAL
open science

Suivi de systèmes non stationnaires basé sur une approche de reconnaissances de forme

Guillaume Mercère, Laurent Bako, Habiboulaye Amadou-Boubacar, Stéphane
Lecoeuche

► **To cite this version:**

Guillaume Mercère, Laurent Bako, Habiboulaye Amadou-Boubacar, Stéphane Lecoeuche. Suivi de systèmes non stationnaires basé sur une approche de reconnaissances de forme. Conférence Internationale Francophone d'Automatique, CIFA'06, Jun 2006, Bordeaux, France. pp.CDRom. hal-00211823

HAL Id: hal-00211823

<https://hal.science/hal-00211823>

Submitted on 21 Jan 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.

Suivi de systèmes non-stationnaires basé sur une approche de reconnaissance des formes

GUILLAUME MERCERE¹, LAURENT BAKO^{2,3}, HABIB AMADOU BOUBACAR^{2,3}, STEPHANE LECŒUCHE^{2,3}

¹ Laboratoire LAII, 40 avenue du Recteur Pineau, 86022 Poitiers, France

² Centre de Recherche Ecole des Mines de Douai, 59508 Douai Cedex, France

³ Laboratoire LAGIS, Bâtiment P2, USTL LILLE 1 - 59655 Villeneuve d'Ascq, France
guillaume.mercere@univ-poitiers.fr, {laurent, amadou,lecoeuche}@ensm-douai.fr

Résumé — Cet article, en lien avec la surveillance de systèmes non-stationnaires, présente une nouvelle approche de suivi basée sur la reconnaissance de formes. Le principe retenu consiste à coupler un algorithme d'identification récursive à une technique de classification dynamique. Dans un premier temps, la technique d'identification en ligne est décrite. Cette estimation récursive d'un modèle linéaire permet d'extraire un vecteur caractéristique de l'état de fonctionnement du procédé. Dans un second temps, ce vecteur est utilisé par un algorithme de classification pour déterminer le mode de fonctionnement courant. Ses capacités à adapter en continu la modélisation de l'espace de décision (modes connus) et à suivre ses évolutions sont brièvement détaillées. Des résultats issus d'exemples de simulation sont finalement proposés.

Mots clés — Identification en ligne, Classification de données évolutives, Modélisation et Suivi de Modes.

I. INTRODUCTION

Les systèmes de supervision se heurtent généralement à l'évolution dans le temps des caractéristiques du procédé. Ceci peut être source d'alarmes intempestives et impose une « maintenance » régulière des références sur lesquelles se fondent les fonctions surveillance et diagnostic. Cette mise à jour des critères de décision doit se faire après avoir déterminé le mode courant de fonctionnement et les tolérances par rapport à cette référence. Bien que la supervision soit un domaine de recherche très étendu, il n'y a que peu de travaux qui envisagent l'actualisation des références. Notre papier s'inscrit dans cette problématique. Plus particulièrement, le suivi de systèmes non-stationnaires est abordé. Les non-stationnarités affectant le comportement statique ou dynamique du système sont définies par des évolutions de la structure et des paramètres du modèle. Une façon de les représenter consiste à considérer deux aspects :

- soit des changements de modes de fonctionnement ;
- soit des dérives des modes de fonctionnement.

L'approche retenue réside dans la modélisation d'un tel système à l'aide d'un ensemble de modèles linéaires représentatifs des différents modes de fonctionnement. Dans cette représentation multi-modèle ou hybride, le changement de mode peut alors s'apparenter à une commutation entre modèles, la dérive de mode

s'interprète comme une évolution des paramètres du modèle. Le comportement du système peut être ainsi décrit par une succession de modes présentant ou non des glissements. Malheureusement, pour obtenir un suivi fiable, ce type de modélisation nécessite une détermination quasi-exhaustive de l'ensemble des modèles, de leurs conditions de commutation ainsi de leur déformations. Afin d'éviter le recours à une telle connaissance *a priori* tel que le nécessite la plupart des méthodes basées sur l'utilisation des résidus, l'espace de parité ou l'identification multimodèle [2][15][17][14], nous présentons une approche basée sur la reconnaissance de formes où la connaissance des modes se fait au travers d'un apprentissage continu ne nécessitant pas (ou peu) de connaissances *a priori*. Cette approche permet de connaître à chaque instant le mode courant dans lequel se trouve le système à l'aide d'un espace de décision continuellement mis à jour.

II. PRINCIPE DE L'APPROCHE

Basée sur des algorithmes récursifs, l'approche comporte 2 niveaux illustrés figure 1 :

- le premier concerne l'extraction d'un vecteur représentatif de l'état de fonctionnement à partir d'un modèle linéaire identifié en ligne. Il nécessite la mise en œuvre d'un algorithme d'identification récursive ;
- le second détermine le mode courant de fonctionnement en se basant sur un espace de décision adapté en continu par apprentissage. Pour cela, un algorithme de classification dynamique est utilisé pour modéliser les modes de fonctionnement.

A. Identification en ligne

A partir de l'ensemble des données d'entrée-sortie du procédé, un algorithme récursif d'identification est utilisé pour estimer à chaque instant le modèle linéaire du système suivi. En fixant comme hypothèse que le système est observable et stable, il est possible de déterminer un modèle sous forme d'état caractéristique de l'état de fonctionnement du procédé. Le choix de ce type de représentation se justifie par le fait que la forme d'état est particulièrement bien adaptée aux systèmes multivariés et qu'elle ne nécessite aucune forme

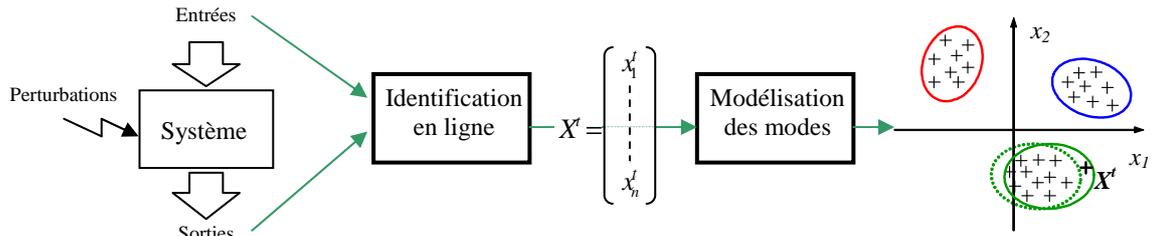


Figure 1: Principe général de l'approche considérée

canonique. Dans ce contexte, l'algorithme EIVPM [11] est utilisé pour sa simplicité de mise en œuvre et son faible coût numérique. À partir du modèle obtenu, un vecteur caractéristique de l'état de fonctionnement courant du procédé est extrait. Cette extraction est basée sur l'exploitation de la matrice d'état du modèle.

B. Classification dynamique

Il s'agit ici de déterminer le mode courant de fonctionnement en se basant sur un espace de décision obtenu par apprentissage. En effet, chaque mode est représenté par un modèle de classe. La particularité de notre approche est qu'elle exploite un algorithme de classification spécifique permettant de réaliser une modélisation en continu des modes de fonctionnement. Le réseau AUDyC [6] dispose en effet de capacités à créer les nouveaux modes (modes inconnus) et à adapter leurs modèles lorsque les distributions des données qui les caractérisent évoluent. Aucune connaissance *a priori* des modes à venir n'est ainsi nécessaire et les variations résultant de non-stationnarités font évoluer dans le temps les modèles des classes associées aux modes.

Après avoir présenté brièvement le principe, les deux paragraphes suivants sont consacrés à la présentation des deux algorithmes récursifs utilisés :

- estimation récursive de l'état de fonctionnement : algorithme récursif des sous-espaces EIVPM ;
- détermination et modélisation en continu des modes de fonctionnement : classifieur dynamique AUDyC.

III. IDENTIFICATION EN LIGNE : EIVPM

La démarche choisie au sein de ce papier repose, dans un premier temps, sur l'estimation récursive d'un modèle linéaire représenté sous forme d'état à structure fixe (ordre connu) par l'approche des sous-espaces. Cette approche permet en effet d'estimer en ligne les matrices d'état $[A, B, C, D]$ du modèle dans un contexte bruité :

$$\begin{aligned} x(t+1) &= Ax(t) + B\tilde{u}(t) + w(t) \\ \tilde{y}(t) &= Cx(t) + D\tilde{u}(t) \\ y(t) &= \tilde{y}(t) + v(t) \end{aligned} \quad (1)$$

où $\tilde{u} \in \mathfrak{R}^{m \times 1}$ et $\tilde{y} \in \mathfrak{R}^{l \times 1}$ sont respectivement les vecteurs des entrées et des sorties, $x \in \mathfrak{R}^{n \times 1}$ le vecteur d'état, $w \in \mathfrak{R}^{n \times 1}$ le bruit d'état et $v \in \mathfrak{R}^{l \times 1}$ un bruit de mesure en sortie. Le principe repose sur l'adaptation des

méthodes hors ligne d'estimation MOESP [16] au problème d'identification récursive. Cette classe d'algorithmes d'identification hors ligne, composée de plusieurs variantes selon les hypothèses fixées sur les perturbations [16], s'appuie sur deux étapes successives :

- une compression de matrices de données d'entrée-sortie à l'aide d'une factorisation QR [5];

- l'application d'une décomposition en valeurs singulières (DVS) [5] à certaines sous-matrices issues de la factorisation précédente conduisant à une estimation consistante de la matrice d'observabilité du système.

Cette dernière étape étant difficile à appliquer en ligne de par sa complexité calculatoire, plusieurs alternatives ont été développées pour le contexte récursif. Deux approches, inspirées du traitement d'antennes, ont plus précisément été proposées :

- la première consiste à adapter le critère de Yang à la mise à jour récursive de la matrice d'observabilité [8] ;
- la seconde repose sur l'ajustement d'une autre technique de traitement d'antennes : la méthode du propagateur [13]. L'avantage de cette démarche relativement aux conceptions précédentes réside dans l'utilisation d'un opérateur linéaire baptisé propagateur conduisant à des critères quadratiques sans contrainte et sans approximation plus faciles à minimiser [10].

Plusieurs études [10] ont permis de mettre en évidence divers avantages théoriques et pratiques de cette seconde approche. Parmi les différentes techniques d'identification récursives utilisant le propagateur, le choix s'est porté sur l'algorithme EIVPM pour sa simplicité de mise en œuvre et son faible coût numérique. Ce dernier est fondé sur la technique hors ligne Ordinary MOESP [16]. La première phase de la méthode EIVPM consiste à mettre à jour, à chaque nouvelle acquisition de données d'entrée-sortie $\{u(t+1), y(t+1)\}$ la décomposition QR :

$$\begin{bmatrix} \sqrt{\lambda} \begin{bmatrix} R_{11}(t) & 0 \\ R_{21}(t) & R_{22}(\bar{t}) \end{bmatrix} & \begin{bmatrix} u_\alpha(t+1) \\ y_\alpha(t+1) \end{bmatrix} \end{bmatrix} \begin{bmatrix} Q_1(t) & 0 \\ Q_2(t) & 0 \\ 0 & 1 \end{bmatrix} \quad (2)$$

où u_α et y_α sont définies comme suit :

$$u_\alpha(t) = [u^T(t) \dots u^T(t+\alpha-1)]^T \quad (3)$$

avec $N \gg \alpha > n$ et λ facteur d'oubli.

Cette mise à jour est réalisée en appliquant une séquence de rotations de Givens [5] choisie pour remodeler le bloc matriciel R . Cette opération permet d'annuler les termes relatifs à u_α et d'extraire un vecteur $z_\alpha \in \mathfrak{R}^{l\alpha \times 1}$ obtenu en intégrant dans y_α les informations contenues dans u_α et $\begin{bmatrix} R_{11}^T & R_{22}^T \end{bmatrix}^T$. Il a été démontré dans l'article [11] que ce vecteur vérifie la propriété suivante :

$$z_\alpha(t+1) = \Gamma_\alpha x(t+1) + b_\alpha(t+1) \quad (4)$$

où Γ_α est la matrice d'observabilité du système

$$\Gamma_\alpha = \begin{bmatrix} C^T & (CA)^T & \dots & (CA^{\alpha-1})^T \end{bmatrix}^T \quad (5)$$

et b_α le vecteur regroupant l'ensemble des perturbations. Connaissant ce vecteur d'observation z_α à l'instant $\bar{t} + 1$, il est possible d'estimer récursivement une base de l'espace d'observabilité sous l'hypothèse que le système étudié soit observable. Cette hypothèse permet en effet d'extraire de la matrice d'observabilité un ensemble de n lignes linéairement indépendantes, regroupées dans une sous-matrice Γ_{α_1} et de récrire Γ_α comme suit :

$$\Gamma_\alpha = \begin{bmatrix} \Gamma_{\alpha_1} \\ \Gamma_{\alpha_2} \end{bmatrix} = \begin{bmatrix} \Gamma_{\alpha_1} \\ P^T \Gamma_{\alpha_1} \end{bmatrix} = \begin{bmatrix} I_n \\ P^T \end{bmatrix} \Gamma_{\alpha_1} \quad (6)$$

où Γ_{α_2} est la matrice complémentaire de Γ_{α_1} et $P \in \mathfrak{R}^{n \times l\alpha - n}$ le propagateur [13] défini comme l'unique opérateur linéaire tel que :

$$\Gamma_{\alpha_2} = P^T \Gamma_{\alpha_1} \quad (7)$$

Puisque la représentation d'état est plurielle, l'équation (6) montre qu'il est possible de déterminer une base de l'espace d'observabilité en estimant le propagateur P . Son estimation récursive est alors réalisée en appliquant au vecteur d'observation z_α une décomposition similaire à celle pratiquée sur Γ_α :

$$z_\alpha(t) = \begin{bmatrix} z_{\alpha_1}(t) \\ z_{\alpha_2}(t) \end{bmatrix} = \begin{bmatrix} I_n \\ P^T \end{bmatrix} \Gamma_{\alpha_1} x(t) + b_\alpha(t) \quad (8)$$

puis en considérant le critère quadratique suivant¹:

$$J(P) = \left\| C_{z_{\alpha_2} \xi} + P^T C_{z_{\alpha_1} \xi} \right\|_F^2 \quad (9)$$

où $\xi \in \mathfrak{R}^{\beta \times 1}$, $\beta > n$ une variable instrumentale non corrélée avec le bruit b_α mais suffisamment avec l'état. La minimisation de cette fonction coût est ensuite accomplie en adaptant la méthode de la variable instrumentale étendue. L'algorithme obtenu s'appelle EIVPM. Pour plus de détails concernant la mise en œuvre de cet algorithme (choix des instruments, réglage des paramètres...), la référence [11] peut-être consultée.

¹ C_{ab} est la matrice de corrélation entre a et b

L'étape d'extraction des matrices d'état du modèle peut ensuite être considérée. Les matrices A et C sont obtenues grâce à la propriété d'A-invariance de Γ_α :

$$\hat{A} = \hat{\Gamma}_\alpha(1:l(\alpha-1),:)^\dagger \hat{\Gamma}_\alpha(\alpha+1:l\alpha,:) \quad (10)$$

$$\hat{C} = \hat{\Gamma}_\alpha(1:\alpha,:) \quad (11)$$

B et D sont estimées quant à elles à l'aide d'une simple régression linéaire [10].

Dans cet article, nous nous intéressons uniquement à l'évolution dynamique du procédé. La poursuite des pôles de ce dernier est un indice très intéressant pour suivre ces variations. Le vecteur caractéristique sera donc constitué des valeurs propres de la matrice A .

IV. MODELISATION EN LIGNE DE MODES : AUDYC

Pour un suivi en ligne, l'outil de modélisation des modes doit traiter les données au fur et à mesure qu'elles arrivent. Il doit notamment incorporer en continu les nouvelles informations, redéfinir la structure des classes et ainsi modéliser leurs évolutions. La prise en compte des différentes situations liées aux non-stationnarités nécessite la mise en place de règles d'adaptation au travers d'un processus d'apprentissage continu.

Au sein des méthodes neuronales, quelques techniques existent avec des architectures autorisant un mode d'apprentissage incrémental [3][4][12]. La plupart de ces algorithmes présentent des inconvénients liés à une modélisation grossière des classes ou à des capacités d'adaptation limitées en environnement non-stationnaire. Afin de combler ces lacunes, un algorithme spécifique a été développé pour la classification dynamique des données évolutives [6]. Il utilise une approche multi-prototype à base de modèles gaussiens permettant de modéliser correctement la structure de classes complexes. L'architecture du réseau proposé (AUDyC) est du type feed-forward à trois couches où les prototypes et les classes sont respectivement représentés par les neurones de la couche cachée et de sortie. Contrairement à ces deux couches, la taille de la couche d'entrée est fixe et imposée par la dimension des données d'entrée. La fonction d'activation de chaque neurone de la couche cachée détermine le degré d'appartenance μ_j^t de l'observation X_t au prototype ϕ_j . Celui-ci est basé sur la distance de Mahalanobis afin d'obtenir une représentation plus fine des classes :

$$\mu_j^t = \exp\left(-\frac{1}{2}(X_t - \bar{X}_j)^T \Sigma_j^{-1} (X_t - \bar{X}_j)\right) \quad (12)$$

où \bar{X}_j et Σ_j sont respectivement le centre et la matrice de covariance du prototype ϕ_j .

Par ailleurs, la fonction d'appartenance permet la mise en œuvre des règles d'évolution du réseau AUDyC. A la première acquisition, le réseau est initialisé avec la création du premier prototype ϕ_1 constituant la première classe Ω_1 . Le prototype ϕ_1 est paramétré par son centre $\bar{X}_1 = X_1$ et par une matrice de covariance initiale Σ_{ini}

préalablement choisie [9]. Ensuite, en fonction des nouvelles acquisitions, différentes situations peuvent se présenter selon le degré d'appartenance μ_j^t comparé à deux seuils μ_{\min} et μ_{\max} (resp. seuil d'appartenance aux prototypes et aux classes). Chaque cas conduit à une procédure spécifique du tableau 1.

	Situations	Règles d'évolution
1	$\forall \phi_j, \mu_j^t < \mu_{\min}$	Création : $\phi_{\text{new}} \subseteq \Omega_{\text{new}}$
2	$\exists \phi_j \subseteq !\Omega_i, \mu_{\min} < \mu_j^t < \mu_{\max}$	Création : $\phi_{\text{new}} \subseteq \Omega_i$
3	$\exists !\phi_j, \mu_j^t > \mu_{\max}$	Adaptation : $\hat{\phi}_j = \phi_j(X_t)$
4	$\exists \phi_{j1} \subseteq \Omega_{i1} \& \phi_{j2} \subseteq \Omega_{i2}, \mu_j^t > \mu_{\min}$	Ambiguïté : $X_t \in \mathcal{X}_{\text{amb}}$

Tableau 1: Règles d'évolution du réseau AUDyC

Le processus d'apprentissage de l'AUDyC est établi en trois phases principales :

1^{ère} Phase : Classification

La phase de classification correspond à la création et à l'adaptation des prototypes et des classes. Dans les situations 1 et 2 du tableau 1, l'observation n'est proche d'aucun prototype existant. C'est une situation de rejet en distance qui est utilisée pour détecter l'apparition de nouveautés dans l'environnement multi-classe et conduit donc à l'insertion d'un nouveau prototype. Si l'observation n'est suffisamment proche d'aucune classe (cas 1), elle conduit également à la création d'une nouvelle classe contenant le prototype; sinon (cas 2) le prototype est affecté à la classe la plus proche. Par contre, dans la situation 3 l'observation est assez proche d'un prototype pour participer à sa définition. Son adaptation est effectuée en utilisant les équations récursives permettant à la fois l'ajout et l'oubli d'information sur une fenêtre de taille N_p :

$$\bar{X}_j^t = \bar{X}_j^{t-1} + \frac{1}{N_p} (X_t - X_{t-N_p+1}) \quad (13)$$

$$\Sigma_j^t = \Sigma_j^{t-1} + \Delta X \begin{bmatrix} \frac{1}{N_p} & \frac{1}{N_p(N_p-1)} \\ \frac{1}{N_p(N_p-1)} & \frac{-(N_p-1)}{N_p(N_p-1)} \end{bmatrix} \Delta X^T \quad (14)$$

$$\Delta X = [X_t - \bar{X}_j^{t-1}, X_{t-N_p+1} - \bar{X}_j^{t-1}], N_p : \text{taille prototype.}$$

2^{ème} Phase : Fusion

La situation 4 du tableau 1 présente le cas du rejet en ambiguïté lorsqu'une observation est suffisamment proche de deux ou plusieurs prototypes pour contribuer à leur structure (en réalité N_{amb} observations). Une nouvelle procédure de fusion a été proposée pour améliorer le processus d'apprentissage de l'AUDyC [1]. Cette procédure consiste à introduire un nouveau critère d'acceptation basé sur la distance de Kullback-Leibler [18] évaluant la similarité de deux densités. Lorsque ce critère est supérieur à un seuil, deux cas sont à

considérer. Si les prototypes appartiennent à la même classe, ils sont fusionnés. Dans le cas contraire, ce sont les classes qui fusionnent.

3^{ème} Phase : Evaluation

La phase d'évaluation est importante pour éliminer les prototypes et classes parasites éventuellement créés par l'influence du bruit. Cette phase se base sur la cardinalité des classes pour détecter les modes non-représentatifs, à savoir celles de cardinalité inférieure à un seuil NC_{\min} .

Pour plus de détails sur le réseau AUDyC, nous invitons le lecteur à consulter [1][6][9]. Une référence concernant l'application de ce réseau à la surveillance d'un procédé est également disponible [7].

A titre d'exemples, les figures 2.a et 2.b illustrent les capacités de l'AUDyC pour la classification dynamique en environnement non stationnaire.

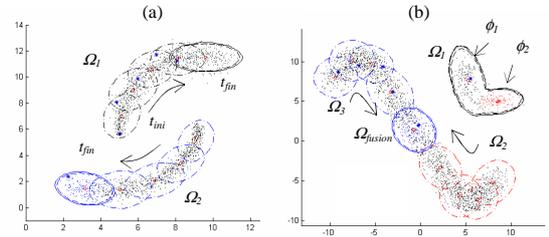


Figure 2: Modélisation en ligne avec le réseau AUDyC.

Simulation (a): Evolution de deux modes gaussiens (2000 données). Simulation (b): Modélisation et adaptation d'une classe multi-prototypes (200 données); Evolution et Fusion de classes (1200 données) dans l'espace non-stationnaire

$$\Sigma_{\text{ini}} = 0.8I_2; \mu_{\min} = 0.018; \mu_{\max} = 0.03; NC_{\min} = 10; N_p = 150; N_{\text{amb}} = 5.$$

V. RESULTATS DE SIMULATION

Cette dernière partie concerne l'application de la méthode proposée à la surveillance d'un procédé non-stationnaire. Dans ce papier, ne sont donnés que des résultats issus d'un système simulé. Le système est décrit par la représentation d'état de l'équation (1) avec:

$$A(t) = \begin{bmatrix} v_1(t) & -0.4 & 0.2 \\ 0 & v_2(t) & -0.5 \\ 0 & 0 & v_3(t) \end{bmatrix}, \quad vP_h(t) = [v_1(t), v_2(t), v_3(t)] \text{ tel que :}$$

$$vP_h(t) = \begin{cases} [0.8, -0.6, 0.1], & 0 < t \leq 1000 \\ [0.65, -0.45, 0.1], & 1000 < t \leq 2000 \\ [0.65, -0.65, 0.05], & 2000 < t \leq 3000 \\ [0.65, -0.65, 0.05] + [-0.05, -0.05, -0.2] \times t / 2780, & 2780 < t \leq 4700 \end{cases}$$

$$\text{et } B = \begin{bmatrix} 0 & 0 & 0.5 \\ 0 & -0.6 & 0 \end{bmatrix}^T, \quad C = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Les entrées du système sont des bruits blancs gaussiens (variance 1). Il est perturbé par un bruit de mesure en sortie coloré (variance 0.03) ainsi que par un bruit d'état coloré (variance 0.05).

Au départ de la simulation, les deux algorithmes sont initialisés : aucune connaissance *a priori* excepté l'ordre du système n'est nécessaire et aucun mode n'est connu. Ensuite de manière récursive, l'algorithme EIVPM identifie le processus et détermine à chaque instant les matrices correspondant à la forme d'état du modèle linéaire approximant le mieux le procédé. Comme cela a été précisé, la caractérisation de l'état de fonctionnement du système est faite à partir de l'extraction de ses valeurs propres instantanées vp_i^t de la matrice \hat{A} (figure 3). qui constitueront le vecteur d'observation X_t utilisé pour déterminer le mode de fonctionnement du système à chaque instant t de telle sorte que : $X_t = [vp_1^t \quad vp_2^t \quad vp_3^t]^T$

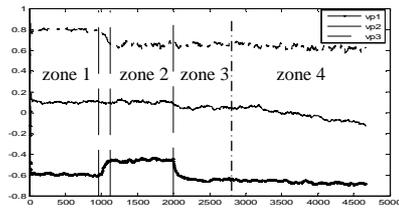


Figure 3: Les valeurs propres de la matrice \hat{A} estimées en ligne avec l'algorithme EIVPM avec $\alpha = \beta = 6, \lambda = 0.995$

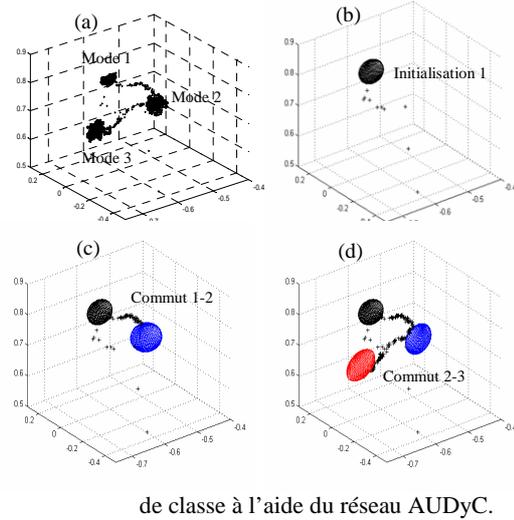
Ce vecteur observation X_t est directement envoyé au réseau AUDyC. A partir de cette observation, l'algorithme AUDyC va mettre à jour les modèles des modes de fonctionnement du processus si ceci s'avère nécessaire puis détermine le mode courant du processus. Pour l'ensemble de cette étude, les paramètres du réseau AUDyC ont été fixés avec les valeurs suivantes : $\Sigma_{ini} = 0,05I_2; \mu_{min} = 0,015; \mu_{max} = 0,02; NC_{min} = 30; N_p = 500; N_{amb} = 5$. Pour plus d'informations sur le choix des paramètres, le lecteur pourra se référer à [9].

- Commutation de modes de fonctionnement

A la première acquisition, le réseau AUDyC initialise la première classe (figure 4.b) correspondant au mode de fonctionnement du système à l'état de départ. Cette classe est adaptée par le réseau AUDyC au fur et à mesure que les observations arrivent. Il se produit alors une dérive rapide d'observations jusqu'à stabilisation. Celle-ci est détectée lorsque le nombre des observations dépasse le seuil NC_{min} . Ce seuil peut être interprété comme la durée minimum permettant de caractériser un mode stable. Une nouvelle classe est alors créée (figure 4.c). De la même façon, un saut de classe conduit à la création de la classe 3 (figure 4-d).

Grâce aux règles de création et d'adaptation du réseau AUDyC, l'apprentissage de tous les modes de la figure 4.a est ainsi élaboré et le mode fonctionnement du système est également déterminé à chaque instant t .

Figure 4: Apprentissage des modes et détection de saut



de classe à l'aide du réseau AUDyC.

- Evolution d'un mode de fonctionnement
Ce cas caractérise une variation dans le temps du mode courant. La figure 5.a présente une évolution du mode 3 de notre processus simulé. L'algorithme de classification dynamique prend en compte la variation de ce mode en adaptant le modèle de classe par l'intégration des nouvelles connaissances. La classe représentant le mode 3 est ainsi correctement définie à chaque instant t (figure 5.b à 5.d).

A travers ces premiers résultats, la figure 6 montre que le mode fonctionnement du système est déterminé à chaque instant en prenant en compte toutes les évolutions, notamment les commutations de modes caractérisées par ses sauts de classe ainsi que les dérives de classe occasionnées par les évolutions de mode de fonctionnement du système au cours du temps.

A partir de ces résultats de modélisation, il est nécessaire d'introduire une connaissance experte permettant de caractériser par une étiquette symbolique les différents modes connus. Cette connaissance permettra également de déterminer si les dérives sont dues à une évolution normale du procédé (mise en température, transition chimique...) ou à l'apparition d'une défaillance.

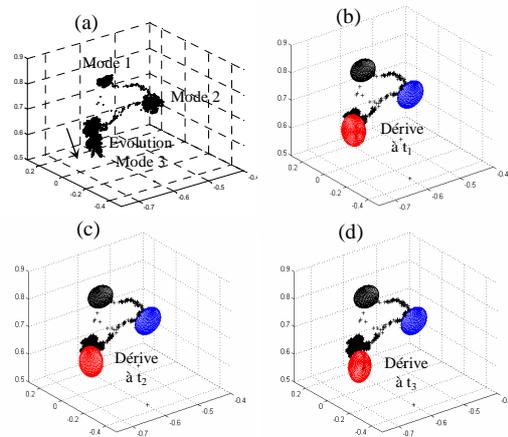


Figure 5: Adaptation de classes par le réseau AUDyC en fonction de l'évolution d'un mode de fonctionnement.

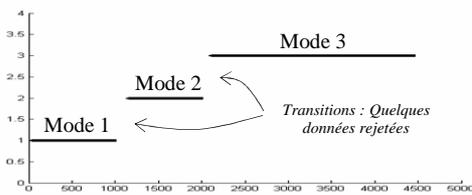


Figure 6: Détermination du mode de fonctionnement courant en fonction du temps.

VI. CONCLUSION

Dans ce papier, une approche originale a été présentée pour la détermination en ligne du mode de fonctionnement d'un système non-stationnaire et son suivi. Cette approche se base sur un outil d'identification récursive couplé avec un algorithme de classification dynamique doté de règles d'adaptation récursives. L'intérêt de cette approche est de prendre en compte les commutations et évolutions de modes du système au cours du temps sans connaissance *a priori*. Les résultats de simulation montrent l'efficacité de cette approche. Des expériences sont en cours sur des données réelles. Les travaux à venir concernent l'amélioration de la partie identification afin de prendre en compte une variation de la structure du modèle. Par ailleurs, des nouveaux outils de détection de sauts et dérives de modes sont à l'étude afin de constituer un module de surveillance en ligne.

VII. REFERENCES

- [1] Amadou B., H., Lecoecue, S., Maouche, S. « AUDyC Neural Network using a new Gaussian Densities Merge Mechanism », *Conf. ICANNGA'05 proceedings*, Coimbra, Portugal, pp. 155-158, 2005.
- [2] Coquempot, V., « Contribution à la surveillance des systèmes industriels complexes », *HDR de l'université des sciences et technologies de Lille*, Lille, France, 2004.
- [3] Deng, D., Kasabov, N., « On-line pattern analysis by evolving self organizing maps », *Neurocomputing*, Vol. 51, pp. 87-103, 2003.
- [4] Eltoft, T., « A new Neural Network for Cluster-Detection-and-Labeling », *IEEE Transactions on Neural Networks*, Vol. 9, pp. 1021-1035, 1998.
- [5] Golub, G., H., Van Loan C., F., « Matrix computations », *John Hopkins University Press*, Baltimore MD, 3rd edition, 1996.
- [6] Lecoecue, S., Lurette C., « Auto-adaptive and Dynamical Clustering Neural Network », *Conf. ICANN'03 proceedings*, Vienna, Austria, pp. 350-358, 2003.
- [7] Lecoecue, S., Lurette, C., « New supervision architecture based on online modelization of non stationary data », *Neural Computing and Applications*, Vol.13, pp. 1433-3058, 2004.
- [8] Lovera, M., Gustafsson, T., Verhaegen, M., « Recursive subspace identification of linear and non linear Wiener state space models », *Automatica*, Vol. 36, pp. 1639-1650, 2000.
- [9] Lurette, C., « Classification dynamique des données évolutives, Application au diagnostic d'une presse hydraulique », *Thèse de doctorat*, Université des Sciences et Technologies de Lille, France, 2003.
- [10] Mercère, G., « Contribution à l'identification récursive des systèmes par l'approche des sous-espaces », *Thèse de doctorat*, Université des Sciences et Technologies de Lille, France, 2004.
- [11] Mercère, G., Lecoecue, S., Lovera, M., « Recursive subspace identification based on instrumental variable unconstrained quadratic optimization », *International Journal of Adaptive Control and Signal Processing*, Vol. 18, pp. 771-797, 2004.
- [12] Mouchaweh S., M., « Diagnosis in real time for evolutionary processes in using pattern recognition and possibility theory », *International Journal of Computational Cognition*, Vol. 2, pp. 79-112, 2004.
- [13] Munier J., Delisle, G., Y., « Spatial analysis using new properties of the cross spectral matrix », *IEEE Trans. on Signal Processing*, Vol. 39, pp. 746-749, 1991.
- [14] Oku, H., « Application of a recursive subspace identification algorithm to change detection », *In The 13th IFAC SYSID*, Rotterdam, The Netherlands, 2003.
- [15] Pekpe, K., Gasso, Kg., Mourot, G., Ragot, J., « Subspace identification of switching model », *In the 13th IFAC SYSID*, Rotterdam, Netherlands, 2003.
- [16] Verhaegen, M., « Identification of the deterministic part of MIMO state space models given in innovations form from input output data », *Automatica*, 1994, Vol. 30, pp. 61-74.
- [17] Domlan, E.A., Maquin D., Ragot J., « Diagnostic des systèmes à commutation, approche par la méthode de l'espace de parité », *Conférence Internationale Francophone d'Automatique*, Douz, Tunisie, 2004.
- [18] Zhou, S. K., Chellappa, R., « Kullback-leibler distance between two gaussian densities in reproducing kernel hilbert space », *ISIT2004*, Chicago IL, USA, 2004.