

# Identification en ligne de systèmes commutants à structure variable

Laurent BAKO<sup>1,2</sup>, Guillaume MERCÈRE<sup>3</sup>, Stéphane LECOEUICHE<sup>1,2</sup>

1. Département Génie Informatique et Productique  
Ecole des Mines de Douai, 59508 Douai, France

2. Laboratoire d'Automatique, Génie Informatique et Signal  
Université des Sciences et Technologies de Lille, 59655, France

3. Laboratoire d'Automatique et d'Informatique Industrielle  
Bâtiment de Mécanique, 86022 Poitiers, France

bako@ensm-douai.fr; guillaume.mercere@univ-poitiers.fr; lecoeuiche@ensm-douai.fr  
<http://www.ensm-douai.fr>

**Résumé**—*La contribution de ce papier consiste dans la proposition d'une méthode d'identification structurée de systèmes multivariés (MIMO). Cette méthode ne fait pas intervenir de Décomposition en Valeurs Singulières (DVS) contrairement la majorité des méthodes des sous espaces disponibles. Couplée à une technique de détermination de l'ordre, cette méthode est utilisée récursivement pour l'identification de systèmes commutants avec ou sans changement d'ordre. Des résultats de simulation concluants confortent notre proposition.*

**Mots-clés**—*Identification, Estimation d'ordre, Méthodes des Sous-espaces, Systèmes Hybrides, Systèmes commutants.*

## I. INTRODUCTION

Il existe dans le monde industriel et même dans la nature un nombre foisonnant de systèmes hybrides. On entend par systèmes hybrides des systèmes qui sont le siège de phénomènes de nature à la fois continue et événementielle. Et pourtant, l'identification de tels systèmes est un domaine relativement récent et actuellement en pleine construction. Parmi les approches envisageables, on distingue le modèle de connaissances qui résulte des lois physiques qui gouvernent le système et le modèle boîte noire qui est basé uniquement sur les mesures entrées-sorties. La recherche d'un modèle boîte noire aux systèmes hybrides est pleine de défis et suscite par là même un vif intérêt de la part des chercheurs. En effet ce problème présente un couplage important entre l'estimation des états discrets et le calcul proprement dit des paramètres du modèle continu correspondant à chaque état discret.

La majorité des auteurs se sont intéressés à la sous classe de systèmes hybrides que représentent les systèmes linéaires par morceaux (Piecewise ARX). S'inspirant des techniques de reconnaissance des formes, [4] propose de partitionner d'abord l'espace des régresseurs en des régions où chaque modèle local est valide et d'estimer ensuite chaque modèle par les Moindres Carrés. La stratégie exposée en plus de nécessiter une connaissance de l'ordre du système n'est pas optimale car la convergence est fortement tributaire de l'initialisation. [14] transforme d'abord le problème en un problème de programmation en nombres entiers pour pouvoir mettre à profit la puissance des outils disponibles dans ce domaine. Cet algorithme est optimal mais souffre en revanche d'une complexité

onéreuse. [6] s'appuie sur l'inférence bayésienne pour élaborer une procédure de classification-identification. Son algorithme requiert une connaissance de l'ordre du système et du nombre de modes de fonctionnement. Le procédé d'identification inclut à chaque fois la détermination de régions de l'espace de régression, domaines de validité de chaque sous-modèle. Dans [16], il est proposé pour l'estimation de systèmes commutants, une méthode algébrique faisant intervenir des polynômes homogènes et leurs dérivées pour la segmentation des données. Elle est idéale seulement dans un contexte déterministe (voir aussi [5]).

Des caractéristiques communes à ces méthodes sont qu'elles sont itératives, difficilement applicables en ligne et fournissent toutes des modèles entrées sorties avec plus ou moins de garantie d'optimalité. Il y a pourtant bien des situations pratiques dans lesquelles un modèle d'état est plus approprié à la description des systèmes.

Une autre approche possible est celle basée sur la détection des instants de commutation. De récents travaux [13], [2] ont démontré l'intérêt de cette approche dans un contexte hors ligne. En effet, la détection semble être une alternative sérieuse pour l'identification des systèmes commutants par les méthodes des sous espaces, le partitionnement étant plus ardu du fait de la non disponibilité de l'état (vecteur de régression inconnu).

Dans cet article nous suivons cette idée de détection et nous l'appliquons dans un contexte en ligne grâce à un nouvel algorithme récursif assurant les tâches multiples d'estimation, de détection, de décision. La méthode présentée s'applique à des systèmes MIMO dont les paramètres et la structure sont susceptibles de varier dans le temps. La valeur de l'ordre du modèle est actualisée continuellement en ligne.

Dans la section II de l'article, nous expliquons la stratégie d'identification structurée qui consiste à estimer la matrice d'observabilité en même temps que l'ordre du système. Dans la section III notre méthode sera adaptée à l'estimation en ligne de modèles à commutation. La dernière partie présente quelques résultats de simulation.

## II. MÉTHODE D'IDENTIFICATION STRUCTURÉE

Considérons un système dynamique discret sous forme de représentation d'état

$$\begin{cases} x_{t+1} = Ax_t + Bu_t + w_t \\ y_t = Cx_t + Du_t + v_t \end{cases} \quad (1)$$

où  $u_t \in \mathbb{R}^{n_u}$ ,  $y_t \in \mathbb{R}^{n_y}$  et  $x_t \in \mathbb{R}^{n_x}$  désignent respectivement les vecteurs d'entrées, de sorties et d'états.  $v_t \in \mathbb{R}^{n_y}$  et  $w_t \in \mathbb{R}^{n_x}$  font référence à des bruits blancs gaussiens. Les séquences de signaux  $\{u_t\}$ ,  $\{v_t\}$ ,  $\{w_t\}$  sont supposées être indépendantes et non corrélées mutuellement. On suppose de plus que le système considéré est observable et asymptotiquement stable. Etant donné une collection finie de données entrées sorties générées par un tel système l'objectif est de recouvrir le quadruplet  $(A, B, C, D)$  de l'équation (1). Il est bien connu que ces paramètres ne sauraient être uniques ; pour toute matrice inversible  $T$ ,  $(TAT^{-1}, TB, CT^{-1}, D)$  explique aussi bien le comportement entrée sortie du modèle (1). Les méthodes d'identification par les sous espaces (4SID) reposent sur l'équation de données suivante :

$$Y_{t,f,N} = \Gamma_f X_{t,N} + H_f U_{t,f,N} + \mathcal{N}_{t,f,N} \quad (2)$$

avec  $\Gamma_f$  la matrice d'observabilité étendue,  $H_f$  une matrice de Toeplitz bloc triangulaire contenant les paramètres de Markov (réponses impulsionnelles) du système :

$$\Gamma_f = [ (C)^T \quad (CA)^T \quad \dots \quad (CA^{f-1})^T ]^T \quad (3)$$

$$H_f = \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{f-2}B & CA^{f-3}B & \dots & D \end{bmatrix} \quad (4)$$

et  $U_{t,f,N}$  la matrice de Hankel des mesures définie comme suit :

$$U_{t,f,N} = \begin{bmatrix} u_t & u_{t+1} & \dots & u_{t+N-1} \\ u_{t+1} & u_{t+2} & \dots & u_{t+N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{t+f-1} & u_{t+f} & \dots & u_{t+f+N-2} \end{bmatrix} \quad (5)$$

$\mathcal{N}_{t,f,N}$  est la résultante des matrices de bruit,  $Y_{t,f,N}$  est définie de façon similaire à  $U_{t,f,N}$  avec  $f \geq n_x$ .

Mercère propose dans [11] d'adapter à l'identification de systèmes une méthode utilisée en traitement d'antennes et connue sous le nom de méthode du propagateur [12], [9]. Cette méthode a pour objet d'identifier la matrice  $\Gamma_f$  dans l'équation (2) à partir d'une estimée de la matrice dite d'observations  $Z_{t,f,N} = Y_{t,f,N} - H_f U_{t,f,N}$  et cela, en s'appuyant sur l'hypothèse d'observabilité. En effet, l'observabilité du système (1) permet d'écrire  $\text{rang}(\Gamma_{n_x}) = \text{rang}(\Gamma_f) = n_x$ . Cela signifie concrètement qu'il existe  $n_x$  lignes linéairement indépendantes dans  $\Gamma_{n_x}$  qui génèrent l'espace des lignes de  $\Gamma_f$ . Il est donc possible de trouver une matrice de permutation  $S$  qui réordonne les lignes de  $\Gamma_f$  de façon à faire apparaître  $n_x$  lignes linéairement indépendantes dans les  $n_x$  premières positions. La permutation de lignes s'applique à toute l'équation

$Z_{t,f,N} = \Gamma_f X_{t,N} + V_{t,f,N}$ . La partition obtenue permet alors d'estimer (comme nous le verrons par la suite) l'espace des colonnes de la matrice d'observabilité via l'extraction d'une sous-matrice de celle-ci appelée propagateur.

Mais le problème majeur est qu'on ne connaît pas  $\Gamma_f$  (c'est ce qu'on veut estimer) pour en repérer les lignes linéairement indépendantes. Dans [11] il est admis implicitement que les  $n_x$  premières lignes de  $\Gamma_f$  sont libres et donc que  $S$  est égale à l'identité. Malheureusement, cela n'est pas toujours le cas pour un système multivariable quelconque.

La formulation faite ici prend en compte ce problème et révisé la méthode du propagateur en vue d'en faire non seulement un schéma d'identification de système MIMO quelconque sans aucune DVS, mais aussi un outil d'estimation de l'ordre. Le schéma proposé se décompose en trois phases :

- Sélection de la permutation adéquate  $S$ ,
- Estimation de la sous-matrice  $P$  de  $\Gamma_f$  (propagateur),
- Détermination de l'ordre  $n_x$  du système

Ces trois phases seront décrites dans les paragraphes suivants.

### A. Sélection de la permutation $S$

La matrice de permutation  $S^{-1}$  est telle que :

$$S\Gamma_f = \begin{bmatrix} \Gamma_0 \\ \Gamma_1 \end{bmatrix} \quad (6)$$

avec  $\Gamma_0$  carré inversible de dimension  $n_x$ . Comme les lignes de  $\Gamma_0$  engendrent l'espace des lignes de  $\Gamma_f$ , il existe une matrice unique  $P$  telle que

$$S\Gamma_f = \begin{bmatrix} \Gamma_0 \\ P\Gamma_0 \end{bmatrix} = \begin{bmatrix} I_{n_x} \\ P \end{bmatrix} \Gamma_0 \quad (7)$$

Une estimation de la matrice  $P$  est donc suffisante pour déterminer l'espace des colonnes de  $\Gamma_f$ . Pour trouver  $S$ , il suffit de caractériser  $n_x$  lignes linéairement indépendantes dans  $\Gamma_f$ .

Notons, dans le cas d'un MIMO général,

$$\gamma_j = \begin{bmatrix} C_j^T \\ C_j^T A \\ \vdots \\ C_j^T A^{f-1} \end{bmatrix} \quad (8)$$

où  $C_j^T$  désigne la  $j$ ème ligne de  $C$ ,  $j$  quelconque. Si tous les pôles du système sont observables de la seule sortie  $y_j$ , comme dans le cas d'un système MISO <sup>2</sup>, alors  $\gamma_j$  est nécessairement de rang  $n_x$ . Mais cela n'est pas toujours vérifié pour un système MIMO car les sorties autres que  $y_j$  peuvent introduire des dynamiques dans le vecteur d'état qui ne seraient pas visibles par  $y_j$ . Considérons une sortie factice sous la forme  $\tilde{z} = y_1 + \sum_{j=2}^{n_y} \alpha_j y_j$  avec  $\alpha_j$  des réels. L'objectif de cette manipulation est de remplacer une des sorties (par exemple  $y_1$ ) par une sortie virtuelle de laquelle tous les pôles du système pourraient être observables. Il est possible de choisir les nombres  $\alpha_j$  tels que  $\gamma = \gamma_1 + \sum_{j=2}^{n_y} \alpha_j \gamma_j$  soit une matrice de rang plein  $n_x$ . Une condition

<sup>1</sup>Rappelons qu'une matrice de permutation est orthogonale :  $S^T S = I$

<sup>2</sup>Pour un système MISO,  $\Gamma_{n_x}$  est une matrice carrée inversible et  $S$  peut être donc choisie égale à la matrice identité

importante est que  $\tilde{z}$  doit être sensible à toutes les dynamiques observables du système quels que soient leurs origines et leurs effets. Le choix des coefficients  $\alpha_j$  doit se faire de façon à garantir l'observabilité de tous les pôles du système dans  $\tilde{z}$ . Nous suggérons de générer les  $\alpha_j$  tous non nuls et de façon aléatoire. Toutefois il sera toujours possible d'intégrer une connaissance a priori dans ce choix. Dans l'équation de données du (2),  $\Gamma_f$  devient alors  $\tilde{\Gamma}_f$ .

$$\begin{aligned} [\tilde{z} \quad y_2 \quad \dots \quad y_{n_y}]^T &\leftarrow [y_1 \quad \dots \quad y_{n_y}]^T \\ \tilde{C}_1^T &\leftarrow C_1^T + \sum_{j=2}^{n_y} \alpha_j C_j^T \\ \tilde{\Gamma}_f &\leftarrow \Gamma_f \end{aligned} \quad (9)$$

En fait,  $\tilde{\Gamma}_f$  est la matrice  $\Gamma_f$  réécrite avec  $\tilde{C}$  et  $A$  et où  $\tilde{C}$  désigne la matrice  $C$  mais dans laquelle la première ligne a été remplacée par  $\tilde{C}_1^T$  à l'image de la première sortie qui est remplacée provisoirement par  $\tilde{z}$ .

L'équation (2) se change formellement en

$$\tilde{Y}_{t,f,N} = \tilde{\Gamma}_f X_{t,N} + \tilde{H}_f U_{t,f,N} + \tilde{N}_{t,f,N} \quad (10)$$

C'est à cette équation que s'applique la permutation de façon à ce que la sous-matrice  $\gamma$  occupe les  $f$  premières lignes de  $S\tilde{\Gamma}_f$ .  $S$  peut être choisie comme :

$$S = \begin{bmatrix} E(1, :) \\ E(n_y + 1, :) \\ \vdots \\ E((f-1)n_y + 1, :) \\ E(2 : n_y, :) \\ E(n_y + 2 : 2n_y, :) \\ \vdots \\ E((f-1)n_y + 2 : fn_y, :) \end{bmatrix} \quad (11)$$

$E$  désignant la matrice identité d'ordre  $fn_y$ . Dans la suite nous conserverons les notations du (2) pour désigner (10).

### B. Estimation du propogateur $P$

Habituellement, les méthodes MOESP procèdent par élimination du terme en  $U$  dans l'équation (2) en projetant l'ensemble de l'équation sur le sous-espace orthogonal à celui des lignes de  $U$ . Dans ce but, suivons la méthode d'implémentation QR de Verhaegen qui encapsule une variable instrumentale pour le traitement du bruit :

$$\begin{bmatrix} U_{t+f,f,N} \\ \Xi \\ Y_{t+f,f,N} \end{bmatrix} = \begin{bmatrix} R_{11}(\bar{t}) & 0 & 0 \\ R_{21}(\bar{t}) & R_{22}(\bar{t}) & 0 \\ R_{31}(\bar{t}) & R_{32}(\bar{t}) & R_{33}(\bar{t}) \end{bmatrix} \begin{bmatrix} Q_1(\bar{t}) \\ Q_2(\bar{t}) \\ Q_3(\bar{t}) \end{bmatrix} \quad (12)$$

avec  $\bar{t} = t + 2f + N - 2$  et  $\Xi$  une matrice d'instruments. Cette dernière est choisie en fonction des hypothèses fixées sur le bruit agissant sur le processus. Selon [8],  $\Xi = U_{t,f,N}$  en absence de bruit d'état ( $w_t = 0$ ) et  $\Xi = \begin{bmatrix} U_{t,f,N} \\ Y_{t,f,N} \end{bmatrix}$  dans le cas contraire, constitue un choix probant d'instruments. Dans le reste du papier nous adopterons chaque fois que la clarté de l'exposé le permet les notations simplifiées  $Y := Y_{t+f,f,N}$  et  $\Gamma := \Gamma_f$ . L'équation du (2) ainsi simplifiée est :

$$Y = \Gamma X + HU + \mathcal{N} \quad (13)$$

En prémultipliant par  $S$  et postmultipliant par  $Q_2^T$ , on obtient :

$$SR_{32} = SYQ_2^T \quad (14)$$

$$= \begin{bmatrix} I_{n_x} \\ P \end{bmatrix} \Gamma_0 X Q_2^T + \mathcal{N} Q_2^T \quad (15)$$

Chou et Verhaegen [3] démontrent qu'en théorie, si  $N$  tend vers l'infini la part de bruit s'atténue dans l'équation ci-dessus du fait de la présence de la variable instrumentale. Bien que  $N$  ne soit pas infinie dans la pratique, nous supposons que le bruit est pratiquement éliminé après application de la variable instrumentale. Comme  $\Gamma_0$  est inversible, on réalise une transformation similaire en posant  $Z = \Gamma_0 X$ . Il est important de remarquer que le fait d'attribuer une valeur à  $S$  revient à choisir une base de la représentation d'état (1) recherchée. Notons  $\Sigma_z = \frac{1}{N} Z Z^T$ ,<sup>3</sup> la matrice de covariance de la variable aléatoire centrée  $z = \Gamma_0 x$ .

En règle générale, toute matrice de covariance de variable aléatoire est au moins positive sémi-définie. En considérant de plus qu'en tant que variables aléatoires, toutes les composantes de  $z$  sont linéairement indépendantes, on peut supposer que  $\Sigma_z$  est définie positive. Cela induit trivialement que toute sous-matrice carrée de  $\Sigma_z$  est aussi définie positive.

$$\Sigma = \frac{1}{N} S R_{32} R_{32}^T S^T = \begin{bmatrix} \Sigma_z & \Sigma_z P^T \\ P \Sigma_z & P \Sigma_z P^T \end{bmatrix} = \begin{bmatrix} \Sigma_{11} & \star \\ \Sigma_{21} & \star \end{bmatrix} \quad (16)$$

Minimisant le critère

$$\|\Sigma_{21} - P \Sigma_{11}\|_F^2 \quad (17)$$

il vient

$$P = \Sigma_{21} \Sigma_{11}^{-1} \quad (18)$$

Il faudra effectuer des permutations à nouveau pour ramener les lignes de  $\Gamma$  dans leurs positions initiales en multipliant par  $S^T$ . La matrice d'observabilité étendue s'en déduit finalement comme  $\Gamma_f = S^T \begin{bmatrix} I_{n_x} \\ P \end{bmatrix}$ .

**Remarque 1.** Il n'est pas nécessaire de calculer entièrement la matrice  $\Sigma$ . Si  $n_x$  est connu, la connaissance de ses  $n_x$  premières colonnes sera suffisante ; sinon il faudra estimer ses  $f$  premières colonnes avec  $f \geq n_x$ .

### C. Estimation de l'ordre

Pour estimer convenablement la matrice  $P$  par la formule (18), il est impératif de connaître l'ordre  $n_x$  du système. Sous des hypothèses peu contraignantes, nous proposons justement dans cette sous section d'estimer l'ordre du système. Du point de vue des méthodes d'identification des sous-espaces, l'estimation de l'ordre résulte presque toujours d'une DVS. Dans le but d'isoler l'ordre, [1] définit les critères d'information NIC et SVC semblables à celui de Akaike et basés justement sur les valeurs singulières par exemple de la matrice  $R_{32}$  du (14). Yang pour sa part a proposé dans [17] un algorithme

<sup>3</sup>on fait l'approximation

$$E [z z^T] = \lim_{N \rightarrow \infty} \frac{1}{N} \left( \sum_{i=0}^N z_i z_i^T \right)$$

pour l'estimation simultanée d'un sous espace et de son rang (PASTd), extrêmement peu cher en calculs mais malheureusement basé sur une approximation. Auparavant Stewart [15], dans l'objectif d'approximer la DVS à moindre coût, avait envisagé une factorisation URV qui permet une lecture du rang. L'ambition affichée ici est de récupérer l'ordre par des moyens simples, applicables en ligne et moins coûteux que la traditionnelle DVS. Il s'agit d'exploiter la structure particulièrement intéressante de la matrice  $\Sigma$ .

Supposons que l'ordre du système (1) admette un majorant disponible, supposition qui est d'ailleurs le point de départ de toute méthode d'identification par les sous espaces (la valeur de  $f$  doit être choisie supérieure à  $n_x$ ). Supposons de plus que  $\Gamma_f$  est tel que le maximum de lignes libres se trouvent placées dans les premières places. Cette condition est toujours vérifiée par des systèmes MISO par exemple. Dans le cas d'un MIMO on se place facilement dans cette hypothèse moyennant une petite manipulation comme celle suggérée dans la sous partie précédente.

L'idée principale dans cette méthode est de découper séquentiellement dans la matrice  $\Sigma$  du (16) une sous-matrice de la forme  $\Delta_r = \Sigma(1 : r, 1 : r)$ ,  $r$  courant d'une valeur  $r_{min}$  à une valeur  $r_{max}$  avec  $r_{min} < n_x < r_{max}$  (illustration sur la figure 1). On associe à chaque valeur de  $r$  un critère de décision noté  $h_{r+1}$  dont la valeur permet la sélection de l'ordre. Cet algorithme est présenté en deux versions équivalentes.

*Version 1* : Une première possibilité consiste à calculer séquentiellement l'inverse de la sous-matrice  $\Delta_r$  extraite de  $\Sigma$  en partant de  $r = r_{min}$  vers  $r = r_{max}$  jusqu'à détection de l'ordre du système.

Si  $\Delta_r$  est inversible,  $\Delta_{r+1}^{-1}$  peut se calculer en faisant usage du lemme d'inversion de matrices blocs (Annexe A de [7]) de la façon suivante :

$$\begin{aligned} \Delta_{r+1}^{-1} &= \begin{bmatrix} \Delta_r & \underline{w}_{r+1} \\ \underline{w}_{r+1}^T & s_{r+1} \end{bmatrix}^{-1} \\ &= h_{r+1}^{-1} \begin{bmatrix} h_{r+1} \Delta_r^{-1} + \underline{\varphi}_{r+1} \underline{\varphi}_{r+1}^T & \underline{\varphi}_{r+1} \\ \underline{\varphi}_{r+1}^T & 1 \end{bmatrix} \end{aligned} \quad (19)$$

avec

$$\begin{aligned} \underline{\varphi}_{r+1} &= -\Delta_r^{-1} \underline{w}_{r+1} \in \mathbb{R}^r \\ h_{r+1} &= s_{r+1} + \underline{w}_{r+1}^T \underline{\varphi}_{r+1} \in \mathbb{R} \end{aligned} \quad (20)$$

Les éléments soulignés indiquent des vecteurs colonnes. Cette partition est possible grâce la structure symétrique de  $\Sigma$ .

Il apparaît grâce à la formule (19) que si  $\Delta_r$  n'est pas singulière,  $\Delta_{r+1}$  est inversible si et seulement si  $h_{r+1} \neq 0$ . Toute sous-matrice (par hypothèse) de  $\Sigma$  de la forme de  $\Delta_r$  est inversible tant que  $r \leq n_x$  et non inversible dès que  $r > n_x$ . Ces arguments justifient l'algorithme suivant pour la recherche de l'ordre.

On admet que  $\Delta_{r_{min}}^{-1}$  est connu et on commence par  $r = r_{min}$ . On procède d'abord à un calcul de  $h_{r+1}$  ; si  $h_{r+1} = 0$  alors  $\Delta_{r+1}$  est singulière et on en conclut  $n_x = r$  ; par contre, si  $h_{r+1} \neq 0$ , la récursion se poursuit en calculant  $\Delta_{r+1}$  puis  $h_{r+2}$  et ainsi de suite . La procédure est arrêtée dès que  $\Delta_{r+1}$  devient singulière. A la sortie de cette boucle

$$\Sigma_{11}^{-1} = \Delta_r^{-1} \quad (21)$$

est connue. La matrice  $P$  est alors calculée comme dans l'équation (18).

**Remarque 2.** La quantité  $h_{r+1}$  est en fait le complément de Schur de  $\Delta_r$  dans  $\Delta_{r+1}$  ;  $h_1 = \Sigma(1, 1)$  est strictement positif comme somme de nombres au carré non tous nuls. Puisque  $\Sigma_z$  est supposée définie-positive, on peut montrer facilement que  $h_r > 0 \forall r \leq n_x$ .

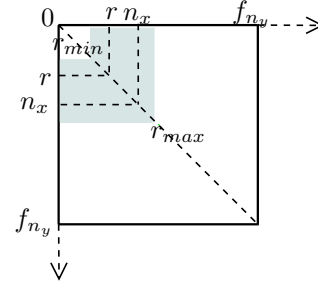


Fig. 1. Schéma de parcours de la matrice  $\Sigma$

*Version 2* : Une autre option de récursion équivalente à la précédente peut être conduite en se focalisant plutôt sur la valeur de la matrice  $P$  dans (18). Disposant de  $P_r$  on veut estimer  $P_{r+1}$ . Ces deux matrices prennent les formes

$$P_r = \begin{bmatrix} \underline{w}^T \\ Q \end{bmatrix} \Delta_r^{-1} = \begin{bmatrix} \underline{p}_r^T \\ \Pi_r \end{bmatrix} \quad (22)$$

$$P_{r+1} = \begin{bmatrix} Q & \underline{v} \end{bmatrix} \begin{bmatrix} \Delta_r & \underline{w} \\ \underline{w}^T & s \end{bmatrix}^{-1} = \begin{bmatrix} \Omega_{r+1} & \underline{\omega}_{r+1} \end{bmatrix} \quad (23)$$

où  $Q = \Sigma(r+2 : f n_y, 1 : r)$ ,  $\underline{w}^T = \Sigma(r+1, 1 : r)$ ,  $\underline{v} = \Sigma(r+2 : f n_y, r+1)$ ,  $s = \Sigma(r+1, r+1)$ .

Quand  $r$  s'incrémente de 1,  $P$  perd une ligne et gagne une colonne. En s'aidant de l'identité matricielle précédente (19), on retrouve facilement les lois d'adaptation suivantes :

$$\underline{\omega}_{r+1} = \frac{\underline{v} - \Pi_r \underline{w}}{s - \underline{p}_r^T \underline{w}}, \quad \Omega_{r+1} = \Pi_r - \underline{\omega}_{r+1} \underline{p}_r^T \quad (24)$$

Cette fois, le paramètre à surveiller est  $h_{r+1} = s - \underline{p}_r^T \underline{w}$ . La deuxième version de l'algorithme présente notamment l'avantage de fournir directement la valeur de la matrice  $P$  en fin de récursion. Mais compte tenu des dimensions des matrices impliquées on risque d'avoir plus de calculs que dans l'option précédente.

En pratique ou dans un contexte stochastique,  $h_{n_x+1}$  ne saurait être vraiment nul. Pour faire marcher cette méthode, il y a intérêt à utiliser un seuillage plutôt que de comparer les paramètres  $h_r$  directement à zéro. Le seuil utilisé peut être naturellement adapté en fonction du niveau du bruit affectant le système.

De façon générale les méthodes d'estimation d'ordre procèdent comme suit. Pour une famille de modèles donnée on essaie l'un après l'autre ou même simultanément différents ordres, celui qui traduit un meilleur compromis entre la complexité et la qualité du modèle est retenu pour être l'ordre du système. Les modèles sont classés du plus vraisemblable au moins probable relativement à un critère de compétitivité.

Ainsi, à défaut de sélectionner l'ordre réel, on sélectionne le plus susceptible d'être en accord avec les données. La stratégie utilisée ici est un peu différente. Elle recherche un ordre idéalisé unique en partant du modèle de représentation d'état sans opérer de mesure de qualité donc sans classement des modèles de cette famille.

Une fois l'ordre du système obtenu,  $\Sigma$  est estimé grâce à la relation (16) d'où on déduit  $P$  puis  $\tilde{\Gamma}_f$  et éventuellement  $\Gamma_f$  en passant par l'étape intermédiaire d'estimation de  $C$  grâce à (9).  $A$  et  $C$  du (1) se calculent alors immédiatement

$$A = \left( \Gamma_f^\dagger \right)^\dagger \Gamma_f^\perp$$

$$C = \Gamma_f(1 : n_y, :),$$

$\Gamma_f^\dagger = \Gamma_f(1 : (f-1)n_y, :)$ ,  $\Gamma_f^\perp = \Gamma_f(n_y + 1 : fn_y, :)$ , mais  $B$  et  $D$  sont souvent estimés par régression linéaire à partir de  $C$  et  $A$  (cf. [8]).

### III. APPLICATION EN LIGNE

#### A. Procédure de mise à jour

La procédure présentée ici s'appuie sur une mise à jour récursive de la matrice  $\Sigma$  du (16) au travers de la nouvelle information capturée. La technique décrite dans la partie précédente est alors répétée à chaque instant  $t$  pour déterminer l'ordre puis les paramètres du système.

En partant de  $r = 1$ ,<sup>4</sup> une récursion spatiale est engagée sur les éléments de  $\Sigma$  jusqu'à l'ordre  $n_x$ . La version récursive de la méthode précédente est fondée sur la mise à jour de la factorisation QR au travers de rotations de Givens. Supposons connue une factorisation QR comme en (12) à l'instant  $\bar{t}$ . A  $\bar{t} + 1$ , une nouvelle colonne de données est rajoutée.

$$\begin{bmatrix} \sqrt{\lambda} \begin{bmatrix} R_{11}(\bar{t}) & 0 & 0 \\ R_{21}(\bar{t}) & R_{22}(\bar{t}) & 0 \\ R_{31}(\bar{t}) & R_{32}(\bar{t}) & R_{33}(\bar{t}) \end{bmatrix} & \begin{bmatrix} u_f(\bar{t}+1) \\ \xi_f(\bar{t}+1) \\ y_f(\bar{t}+1) \end{bmatrix} \end{bmatrix}$$

avec  $y_f(\bar{t}+1) = [y^T(\bar{t}-f+2) \dots y^T(\bar{t}+1)]^T$ ;  $\lambda < 1$  désigne un facteur d'oubli;  $u_f$  et  $\xi_f$  sont définies d'une façon similaire à  $y_f$ ;  $\xi$  fait référence à une variable instrumentale. Dans le but de restituer à cette matrice sa forme triangulaire, on applique deux séquences de rotations de Givens [8] comme suit

$$\begin{aligned} & \begin{bmatrix} \sqrt{\lambda} \begin{bmatrix} R_{11}(\bar{t}) & 0 & 0 \\ R_{21}(\bar{t}) & R_{22}(\bar{t}) & 0 \\ R_{31}(\bar{t}) & R_{32}(\bar{t}) & R_{33}(\bar{t}) \end{bmatrix} & \begin{bmatrix} u_f(\bar{t}+1) \\ \xi_f(\bar{t}+1) \\ y_f(\bar{t}+1) \end{bmatrix} \end{bmatrix} \mathbf{G}_1(\bar{t}+1) \\ &= \begin{bmatrix} R_{11}(\bar{t}+1) & 0 & 0 & 0 \\ R_{21}(\bar{t}+1) & \sqrt{\lambda}R_{22}(\bar{t}) & 0 & \bar{\xi}_f(\bar{t}+1) \\ R_{31}(\bar{t}+1) & \sqrt{\lambda}R_{32}(\bar{t}) & \sqrt{\lambda}R_{33}(\bar{t}) & \bar{y}_f(\bar{t}+1) \end{bmatrix} \mathbf{G}_2(\bar{t}+1) \\ &= \begin{bmatrix} R_{11}(\bar{t}+1) & 0 & 0 & 0 \\ R_{21}(\bar{t}+1) & R_{22}(\bar{t}+1) & 0 & 0 \\ R_{31}(\bar{t}+1) & R_{32}(\bar{t}+1) & \sqrt{\lambda}R_{33}(\bar{t}) & \bar{y}_f(\bar{t}+1) \end{bmatrix} \end{aligned}$$

On peut ensuite montrer que la mise à jour de  $R_{32}R_{32}^T$  s'écrit

$$\begin{aligned} & R_{32}(\bar{t}+1)R_{32}(\bar{t}+1)^T \\ &= \lambda R_{32}(\bar{t})R_{32}(\bar{t})^T \\ &+ \underbrace{\bar{y}_f(\bar{t}+1)\bar{y}_f(\bar{t}+1)^T - \bar{y}_f(\bar{t}+1)\bar{y}_f(\bar{t}+1)^T}_{\Phi_R(\bar{t}+1)} \end{aligned} \quad (25)$$

<sup>4</sup>En absence de variable instrumentale il est possible de partir de  $r_{min} > 1$  où  $\Delta_{r_{min}}^{-1}$  est adapté récursivement dans le temps en utilisant le lemme d'inversion matricielle

On en déduit la mise à jour de la matrice  $\Sigma$  de l'équation (16)

$$\Sigma(\bar{t}+1) = \lambda \Sigma(\bar{t}) + S \Phi_R(\bar{t}+1) S^T \quad (26)$$

Comme nous l'avons déjà souligné plus haut, il n'est pas nécessaire d'adapter entièrement la matrice  $\Sigma$ . Si l'ordre est susceptible d'évoluer, il est essentiel d'adapter toutes les  $r_{max}$  premières colonnes, en admettant que  $S$  est convenablement choisi.

A chaque instant, on adapte  $\Sigma$  par (26) puis on estime l'ordre et le propagateur  $P$  comme dans la section II.  $\Gamma_f$  est ensuite extraite ainsi que les matrices de la représentation d'état.

#### B. Application aux systèmes linéaires commutants

Considérons maintenant un système à commutations, linéaire par morceaux. Un modèle local est décrit par la représentation d'état  $\mathcal{M}_i$

$$\mathcal{M}_i : \begin{cases} x_{t+1} = A_i x_t + B_i u_t + w_t \\ y_t = C_i x_t + D_i u_t + v_t \end{cases} \quad (27)$$

L'équation de données du (2) reste valable dans le cas des systèmes commutants à la condition que toutes les données impliquées soient issues du même modèle local  $i$ . Quand le système entre dans le voisinage d'une commutation, cette équation n'est plus acceptable sous la même forme puisque les matrices  $U$  et  $Y$  contiendront des données provenant de deux sous-modèles différents. Soit  $\tau$  l'instant de commutation d'un modèle  $i$  à modèle  $j$ . Pendant la transition, on a l'équation suivante :

$$\begin{aligned} & [ Y^i \quad y_f(\tau) \quad \dots \quad y_f(\tau+k) ] = \\ & [ \Gamma^i X^i \quad \Gamma(\tau)x(\tau) \quad \dots \quad \Gamma(\tau+k)x(\tau+k) ] \\ & + [ H^i U^i \quad H(\tau)u_f(\tau) \quad \dots \quad H(\tau+k)u_f(\tau+k) ] + \mathcal{N} \end{aligned}$$

$k = 1, \dots, f$ , avec  $y_f(\tau) = [y(\tau-f+1)^T \dots y(\tau)^T]^T$ ,  $Z^i$  désigne une matrice de dimensions appropriées relative aux données issues intégralement du modèle  $i$ . Le schéma ci-dessous donne une illustration du processus d'évolution de la matrice  $\Gamma_f$  du modèle  $i$  au modèle  $j$ .

$$\begin{array}{ccc} \tau-1 & \tau & \tau+1 \\ \begin{bmatrix} C_i \\ C_i A_i \\ \vdots \\ C_i A_i^{f-2} \\ C_i A_i^{f-1} \end{bmatrix} & \rightarrow & \begin{bmatrix} C_i \\ C_i A_i \\ \vdots \\ C_i A_i^{f-2} \\ C_j A_i^{f-1} \end{bmatrix} & \rightarrow & \begin{bmatrix} C_i \\ C_i A_i \\ \vdots \\ C_j A_i^{f-2} \\ C_j A_j A_i^{f-2} \end{bmatrix} & \rightarrow \dots \end{array}$$

La transition réelle de  $i$  à  $j$  dure environ  $f$  fois la période d'échantillonnage mais vue de l'algorithme, elle pourrait durer une période supplémentaire qui correspond au temps de convergence, lequel est fonction du facteur d'oubli. La gestion de la transition est plutôt une tâche délicate étant donnée la nature récursive de l'approche envisagée. L'approche de détection que nous utilisons ici se base sur l'algorithme d'estimation de l'ordre. En effet toute rupture de modèle (même non accompagnée d'un changement d'ordre) entraîne une sur-estimation de l'ordre (comme on peut le voir sur la figure 3). Une fois qu'un changement se produit (est détecté), l'apprentissage du régime en cours  $i$  est immédiatement stoppé et le

modèle final obtenu mémorisé.<sup>5</sup> Pendant ce temps, un nouveau modèle est créé pour capturer la nouvelle dynamique qui vient d'apparaître.

Durant l'intervalle de temps  $[\tau, \tau + f]$ , multiplier à gauche par  $Q_2^T$  de (12) ne permet plus d'éliminer le terme  $HU$  qui n'existe même plus sous cette forme. Aussi longtemps que des données mixtes seront présentes dans  $U$  et  $Y$  et que leur contribution sera non négligeable, la procédure d'identification ne sera pas à même de fournir les estimées espérées. Mais l'identification recouvre de nouveau tout son sens aussitôt que ces données seront rejetées. La stratégie d'estimation et de détection adoptée ici s'appuie sur l'hypothèse que les instants de commutation sont distants d'au moins une durée  $T_s$  appelée temps de séjour qui tient compte du temps de convergence de l'algorithme d'estimation du propagateur  $P$  et de la largeur de la fenêtre de données.

### C. Reconnaissance de modes

Nous abordons dans cette partie la phase de classification inhérente à la nature de notre problème. Si on se retrouve dans un mode de fonctionnement déjà connu, il serait convenable de le reconnaître et si besoin est, de le fusionner avec celui déjà mémorisé. Dans cette optique considérons une mesure de similarité entre les paramètres de Markov en nous basant sur une métrique appropriée. La distance de Mahalanobis est très prisée pour ces types de tâches. Mais pour des raisons de simplicité et de coût de calculs nous utiliserons plutôt la distance euclidienne. Les paramètres de Markov du modèle actif (en cours d'identification) sont comparés à ceux de tous les modèles connus et enregistrés. A partir de ce moment, si la mesure de similarité par rapport à un modèle donné est inférieure à un certain seuil, le modèle actif n'est pas nouveau. Par conséquent il pourrait être fusionné avec celui déjà connu. Si la distance à tous les modèles connus est supérieure à ce seuil, le modèle est nouveau. Soit  $H = \begin{bmatrix} D \\ \Gamma_f B \end{bmatrix}$  et  $\underline{\theta} = \text{vec}(H)$ , un vecteur de dimension  $(f+1)n_y n_u$ .  $\text{vec}(\cdot)$  est l'opérateur de vectorisation. La mesure de similarité entre le sous-modèle en cours d'identification et un sous-modèle  $i$  connu et mémorisé est définie comme

$$d_i(t)^2 = \underline{e}_i(t)^T \underline{e}_i(t) \quad (28)$$

avec  $\underline{e}_i(t) = \underline{\theta}(t) - \underline{\theta}_i$ .

Un seuillage est utilisé pour décider si le nouveau modèle est connu ou non. On fusionne alors si nécessaire les sous-modèles semblables. Le principe suivi est à peu près le même que dans [10].

## IV. SIMULATION

Pour illustrer la méthode présentée dans ce document considérons un exemple numérique de simulation. Quatre sous-modèles linéaires  $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$  respectivement d'ordre 2, 2, 4, 3 commutent comme sur la figure 2.

$$\mathcal{M}_1 : \begin{cases} A_1 = \begin{bmatrix} 1.23 & -0.49 \\ 1 & 0 \end{bmatrix} & B_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \\ C_1 = [ 0.864 \quad 0.755 ] & D_1 = 1 \end{cases}$$

$$\mathcal{M}_2 : \begin{cases} A_2 = \begin{bmatrix} 0.36 & -0.5 \\ 0.5 & 0 \end{bmatrix} & B_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \\ C_2 = [ 1.62 \quad 0.1 ] & D_2 = 2 \end{cases}$$

$$\mathcal{M}_3 : \begin{cases} A_3 = \begin{bmatrix} 0.563 & -0.32 & 0.46 & -0.314 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & B_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ C_3 = [ 1.68 \quad .75 \quad 0.28 \quad 1.58 ] & D_3 = 0 \end{cases}$$

$$\mathcal{M}_4 : \begin{cases} A_4 = \begin{bmatrix} 0.1725 & -0.032 & -0.65 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & B_4 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \\ C_4 = [ 1.25 \quad .5 \quad 0 ] & D_4 = 0 \end{cases}$$

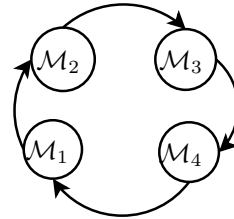


Fig. 2. Commutation entre les modèles locaux

On choisit  $f = 7, \lambda = 0.9$ . Le choix de  $f$  obéit à la contrainte  $f \geq n_x$ ; le facteur d'oubli, lui, résulte d'un compromis entre la vitesse de convergence de l'algorithme d'identification et l'influence du bruit sur les estimées. Le signal d'entrée est un bruit blanc gaussien, centré de variance 1.5. Le signal et le bruit blanc de sortie sont dans un rapport de 35 dB. La matrice des entrées passées est utilisée comme variable instrumentale (cf. eq. 12). Les instants de commutation sont 300, 600, 800, 1100 etc. Ces instants sont estimés avec un retard assez faible comme on peut le voir sur la figure 3. En revanche, puisque toute transition dure  $f$  échantillons, le temps mis pour retrouver le bon ordre après détection d'une rupture de modèle est nécessairement supérieur à  $f$  (cf. fig. 3).

La figure 3 présente l'estimée de l'ordre en même temps qu'une représentation des modules des pôles du système. On notera que les pôles de module faible sont plus sensibles à l'effet du bruit. Sur la figure 4, c'est la mesure de vraisemblance ( $\exp(-d_j^2)$ ) entre le sous-modèle en cours d'identification et les sous-modèles enregistrés qui est représentée. On voit bien qu'au delà de 1100 une de ces mesures devient pratiquement égale à 1, ce qui traduit le retour au mode  $\mathcal{M}_1$ .

Sur la figure 5 est représentée la valeur résiduelle du paramètre  $h$  définie dans la partie II-C. Rappelons que cette quantité est normalement nulle en théorie et en absence de toute influence de bruit sur les données. Cette courbe montre que la présence de bruit tend à accroître les valeurs des paramètres  $h$  et pourrait donner l'impression d'un ordre plus élevé qu'il n'est en réalité.

## V. CONCLUSION

Ce travail démontre la possibilité d'identifier en ligne de façon récursive le modèle d'un système MIMO commutant, linéaire par morceaux. La méthode proposée peut s'appliquer

<sup>5</sup>La détection est probablement faite avec un retard  $\delta$ . Donc, au lieu de mémoriser les valeurs disponibles à l'instant  $\tau$ , on enregistre celles estimées à l'instant  $\tau - \delta$

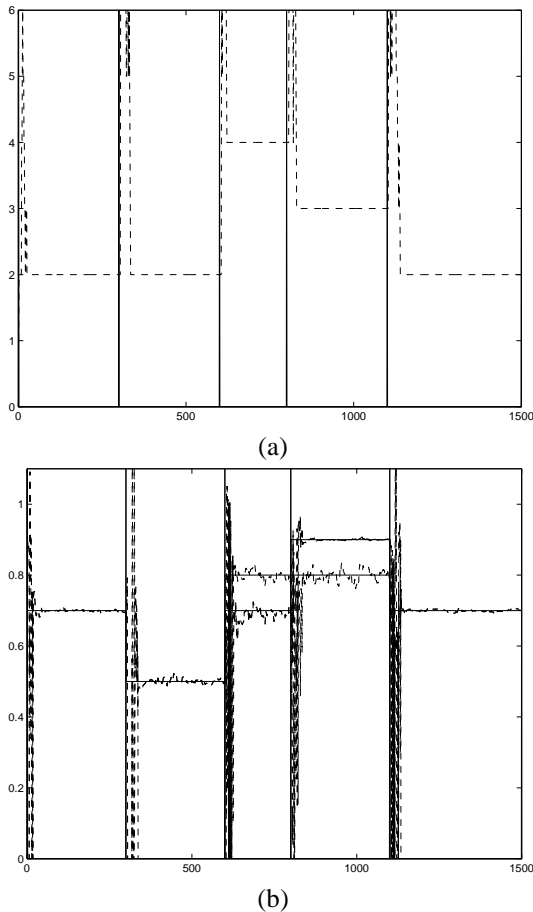


Fig. 3. (a) : Estimation en ligne de l'ordre du système. Les lignes verticales en trait plein indiquent les instants de commutation réels - (b) : Module des pôles des modèles locaux estimés en ligne

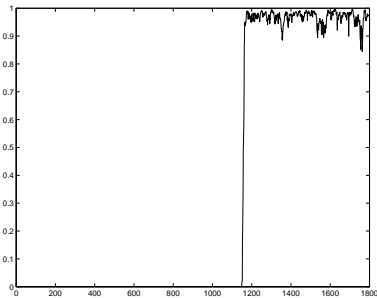


Fig. 4. Mesure de vraisemblance avec la distance euclidienne

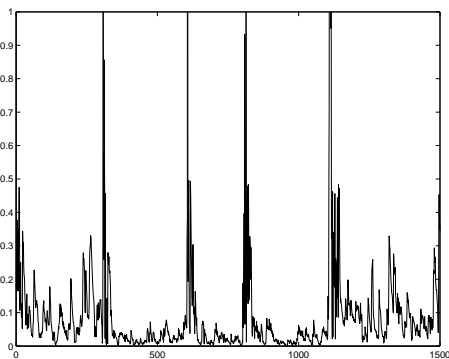


Fig. 5. Valeur du paramètre  $h$ . Cette valeur est constituée d'incertitude et de bruit puisqu'elle est normalement nulle.

aussi bien à des systèmes LTI qu'à des systèmes variant lentement dans le temps. Généralement la base de données servant à l'identification d'un modèle n'est pas complète en ce sens qu'elle ne recouvre pas tous les modes de fonctionnement du système. Dans ce cas, la plupart des méthodes proposées dans la littérature qui sont plutôt des méthodes hors ligne pourraient présenter une grande fragilité. La démarche suivie dans cet article ne souffre pas de ce type de problème. Elle permet de caractériser en continu les modes de fonctionnement au fur et à mesure qu'ils apparaissent. A un instant donné un certain nombre de modèles locaux ont été déjà identifiés et mémorisés. Un point intéressant à développer serait de gagner du temps dans la reconnaissance des prochaines occurrences de ces mêmes modes. Ainsi, il n'y aurait plus besoin de réidentifier à chaque fois un même sous-modèle.

## RÉFÉRENCES

- [1] Dietmar Bauer. Order estimation for subspace methods. *Automatica*, 37 :1561–1573, 2001.
- [2] José Borges, Vincent Verdult, Michel Verhaegen, and Miguel Ayala Botto. A switching detection method based on projected subspace classification. In *CDC-ECC05*, 2005.
- [3] C. T. Chou and Michel Verhaegen. Subspace algorithms for the identification multivariables dynamic error-in-variables models. *Automatica*, 33, No.10 :1857–1869, 1997.
- [4] Giancarlo Ferrari-Trecate, Marco Muselli, Diego Liberati, and Manfred Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39 :205–217, 2003.
- [5] A. Lj. Juloski, W.P.M.H Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J.H.G Niessen. Comparison of four procedures for the identification of hybrid systems. In *HSCC*. HSCC, 354-369, 2005.
- [6] A. Lj. Juloski, S. Weiland, and W.P.M.H. Heemels. A bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50, No 10 :1520–1533, 2005.
- [7] T. Kailath and A. H. Sayed. *Fast Reliable Algorithms for Matrices with Structure*. SIAM, PA, 1999.
- [8] Marco Lovera, Tony Gustafsson, and Michel Verhaegen. Recursive subspace identification of linear and non-linear wiener state-space models. *Automatica*, 36 :1639–1650, 2000.
- [9] Sylvie Marcos, Alain Marsal, and Messaoud Benidir. The propagator method for source bearing estimation. *Signal Processing*, 42 :121–138, 1995.
- [10] Guillaume Mercère, H. Amadou-Boubacar, Laurent Bako, and Stéphane Lecoche. Suivi de systèmes non stationnaires basé sur une approche de reconnaissance des formes. In *CIFA, Bordeaux, France*, 2006.
- [11] Guillaume Mercère, Stéphane Lecoche, and Christian Vasseur. A new recursive method for subspace identification of noisy systems : Eivpm. In *13th IFAC Symposium on System Identification*, 2003.
- [12] J. Munier and G. Y. Delisle. Spatial analysis using new properties of the cross spectral matrix. *IEEE Transactions on Signal Processing*, 39 :746–749, 1991.
- [13] Komi Midzodzi Pekpe, Gilles Mourot, Komi Gasso, and José Ragot. Identification of switching systems using change detection technique in the subspace framework. In *43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.
- [14] Jacob Roll, Alberto Bemporad, and Lennart Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40 :37–50, 2004.
- [15] G.W. Stewart. An updating algorithm for subspace tracking. *IEEE Transactions on Signal Processing*, 40 :1535–1541, 1992.
- [16] René Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (gpca). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, No 12 :1–15, 2005.
- [17] Bin Yang. An extension of the pastd algorithm to both rank and subspace tracking. *IEEE Signal Processing Letters*, 2, No. 9 :179–182, 1995.