UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE **Numéro d'ordre : 4260**

Contribution à l'identification de systèmes dynamiques hybrides

THÈSE

présentée et soutenue publiquement le 21 Novembre 2008

pour l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille

(Spécialité Automatique et Informatique Industrielle)

par

Laurent Bako

Composition du jury

Président :	C. Vasseur	Professeur des Universités, USTL, Lille
Rapporteurs :	H. Guéguen	Professeur Supélec, Rennes
	J. Ragot	Professeur des Universités, INPL, Nancy
Examinateurs :	J. Bordeneuve-Guibé	Professeur ISAE/ENSICA, Toulouse
	H. Garnier	Professeur des Universités, UHP, Nancy 1
	W. Perruquetti	Professeur des Universités, Ecole Centrale de Lille
Directeur de thèse :	S. Lecœuche	Professeur des Ecoles des Mines, ENSM/Douai
Co-encadrant :	G. Mercère	Maître de Conférences, Université de Poitiers

Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS) — UMR CNRS 8146.
Ecole des Mines de Douai, Département Informatique et Automatique.

Mis en page avec la classe thloria.

Remerciements

Il y a tellement de gens à qui je dois des remerciements pour être arrivé à ce stade qu'il m'est vraiment difficile de pouvoir citer intégralement tout le monde. Ainsi, je remercie d'une manière générale tous ceux qui, d'une manière ou d'une autre, m'ont aidé durant ces trois années de thèse.

Je voudrais remercier plus particulièrement Prof. Stéphane Lecœuche, mon directeur de thèse, de m'avoir accepté en thèse. Je le remercie pour sa confiance, sa sympathie et son assistance constante durant toutes ces trois années. Grâce à son aimable soutien, j'ai pu visiter durant ma thèse, le laboratoire de Dr René Vidal, à l'Université Johns Hopkins à Baltimore. Je lui en suis très sincèrement reconnaissant.

J'exprime ma gratitude à Dr Guillaume Mercère, Maître de Conférences à l'Université de Poitiers pour avoir co-encadré cette thèse et pour les nombreux échanges scientifiques qu'on a pu avoir ensemble. Sa sympathie, sa disponibilité et ses encouragements m'ont aidé à prendre progressivement goût à ce sujet de recherche.

J'aimerais exprimer aussi ma gratitude à Dr René Vidal pour m'avoir accueilli durant six mois dans son équipe à l'Université Johns Hopkins. J'ai été vraiment ravi de travailler sous sa direction. Le chapitre 5 de cette thèse a été rédigé durant ce séjour. Ce fut une aventure riche tant sur les plans humain, culturel que scientifique.

Mes vifs remerciements s'adressent maintenant à

- Prof. Hervé Gueguen, Professeur à Supelec Rennes,
- Prof. José Ragot, Professeur à l'Institut National Polytechnique de Lorraine,

qui m'ont fait l'honneur d'être les rapporteurs de mon travail.

Je suis très reconnaissant à

- Prof. Joël Bordeneuve-Guibé, enseignant-chercheur à l'Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse,
- Prof. Hugues Garnier, professeur à l'Université Henry Poincaré de Nancy,
- Prof. Wilfrid Perruquetti, professeur à l'Ecole centrale de Lille,
- Prof. Christian Vasseur, professeur à l'Université des Sciences et Technologies de Lille

qui ont accepté de participer à mon jury en qualité d'examinateurs.

Je n'oublie pas Dr Eric Duviella et Dr Abdelkader Akhenak pour les collaborations enrichissantes qu'on a pu avoir concernant la modélisation de systèmes hydrographiques, le diagnostic et l'observation de processus. Je tiens à dire un grand merci à l'équipe du département Informatique et Automatique des Mines de Douai. Merci plus particulièrement à mes collègues H. Lyes, B. Khaled, T. Moussa et A. Ouafae pour le climat de bonne humeur qu'ils contribuent à créer et à maintenir.

Enfin, j'ai une pensée très particulière pour Lilly ainsi que pour toute ma famille. Je ne serais certainement pas arrivé au bout de ces trois années sans votre soutien, vos encouragements et vos prières. Merci à vous.

A la mémoire de mon père

Table des matières

Notations

$\mathbb{E}[\cdot]$	:	espérance mathématique
$\mathbb{R}$	:	ensemble des nombres réels
$\mathbb{Z}$	:	ensemble des nombres entiers relatifs
$\mathbb{R}^n$	:	espace euclidien de dimension n
Γ_f	:	matrice d'observabilité étendue
H_f	:	matrice de Toeplitz des paramètres de Markov
I_n	:	matrice identité d'ordre n
$\text{im}(A)$	:	espace colonne de A
$\text{null}(A)$	:	espace noyau de A
$\hat{A}$	:	estimée de A
$A(i,:)$	:	$i$ème vecteur ligne de A
$A(:,i)$	:	$i$ème vecteur colonne de A

Acronymes

SISO	:	Single Input-Single Output
MISO	:	Multiple Input-Single Output
MIMO	:	Multiple Input-Multiple Output
ARX	:	AutoRegressive eXogenous
PWA	:	PieceWise Affine
PWARX	:	PieceWise AutoRegressive eXogenous
SARX	:	Switched AutoRegressive eXogenous
MLD	:	Mixed Logical Dynamical
LC	:	Linear Complementarity
ELC	:	Extended Linear Complementarity
MMPS	:	Max-Min-Plus-Scaling
SLC	:	Systèmes Linéaires Commutants
SVM	:	Support Vectors Machines
RBF	:	Radial Basis Function
DVS	:	Decomposition en Valeurs Singulières
MOESP	:	Multivariable Output Error State Space
N4SID	:	Numerical Subspace State Space System Identification
ERA	:	Eigenvalue Realization Algorithm
DWO	:	Direct Weight Optimization
MCR	:	Moindres Carrés Récursifs
EM	:	Expectation Maximization
GPCA	:	Generalized Principal Component Analysis
MIN PFS	:	Minimum Partition into Feasible Subsystems

Introduction générale

1 Contexte de la thèse

Dans de nombreuses applications modernes, l'interaction de plus en plus importante entre les systèmes numériques (ordinateurs, logiciels, composants logiques, etc.) et les processus physiques (équations différentielles impliquant des signaux continus) a conduit, en Automatique, à l'émergence et à la formalisation des systèmes dits hybrides [43], [55]. Formellement, les systèmes hybrides peuvent être définis comme des systèmes où interagissent des phénomènes de nature à la fois continue et événementielle. Le comportement continu est le fait de l'évolution naturelle du procédé physique tandis que le comportement discret ou événementiel peut être dû à la présence de commutateurs, de phases de fonctionnement, de transitions, de codes de programme informatique, etc. Par ailleurs il existe des procédés physiques continus de par leur fonctionnement naturel, mais qui peuvent être regardés d'un point de vue de modélisation comme des systèmes hybrides. Par exemple, dans un système comme le trafic routier en milieu urbain, le flux de véhicules est un phénomène quasi-continu mais qui peut, en pratique, être modélisé comme un système hybride dont les sous-modèles correspondent à différentes périodes de la journée. De même, un système continu peut devenir hybride par adjonction d'une stratégie de commande discrète (par exemple, une batterie de correcteurs avec un critère de sélection instantanée du correcteur approprié).

L'analyse et la conduite de systèmes dynamiques hybrides, comme de tout autre type de système dynamique, nécessitent bien souvent que l'on dispose d'un modèle mathématique de ces systèmes. Ce modèle peut être inféré directement à partir des lois physiques qui régissent le comportement du système. Mais cette option a une portée assez limitée quand le processus considéré devient complexe ou quand les principes physiques qui gouvernent son fonctionnement ne sont pas assez bien connus ou impliquent des paramètres difficilement (ou non) mesurables. Dans ce cas, une alternative peut être de recourir à un modèle comportemental estimé directement à partir des mesures entrée-sortie du système. C'est dans ce contexte que s'inscrivent les travaux présentés dans ce manuscrit. La modélisation fait ici référence à la procédure qui consiste dans le choix ou la construction d'une structure de modèle mathématique appropriée pour la description d'un système physique donné. Quant à l'identification, c'est la démarche qui se rapporte à l'estimation des paramètres du modèle que l'on s'est fixé pendant la phase de modélisation, à partir des données de mesure. Nous nous intéresserons à la modélisation et à l'identification de systèmes dynamiques et plus particulièrement de systèmes linéaires hybrides, c'est-à-dire des modèles

mathématiques constitués d'une collection d'un certain nombre de sous-modèles linéaires ou affines en interaction.

2 Travaux antérieurs

Le problème de l'identification de systèmes hybrides peut s'énoncer génériquement de la façon suivante : à partir de données entrée-sortie générées par l'ensemble des sous-systèmes en interaction et collectées de façon aléatoire, estimer aussi bien les ordres des sous-modèles que leur nombre et leurs paramètres sachant que les ordres peuvent différer d'un sous-modèle à un autre et que les commutations (qui déclenchent le passage d'un sous-modèle à un autre) peuvent survenir à tout instant. Dans une formulation plus générale de ce problème, il est même possible de considérer que les sous-modèles constitutifs du système hybride à estimer, sont de structures différentes (des sous-modèles linéaires mélangés à des sous-modèles non-linéaires par exemple).

L'identification d'un système hybride à partir de ses mesures entrée-sortie est un sujet délicat qui suscite, depuis le début du siècle, de plus en plus d'intérêt de la part des chercheurs [46], [99], [131], [67], [101], [87], [18], [11]. En plus d'être un problème d'identification standard, c'est également un problème de partitionnement des données de régression. Ces deux étapes sont si couplées qu'elles sont difficilement dissociables. En effet, la connaissance des paramètres caractéristiques de chaque sous-modèle du système permettrait un partitionnement presque immédiat (estimation de l'état discret) de l'ensemble des données. De même, si les séquences de données disponibles étaient déjà séparées par sous-modèle générateur, il ne resterait plus qu'à estimer les paramètres de chacun d'eux en recourant aux méthodes classiques de régression.

Le grand défi du problème d'identification de systèmes hybrides est que les données de mesure collectées sont seulement disponibles comme un mélange de données générées par des sous-modèles différents de sorte qu'on ne sait pas *a priori* quel sous-modèle a généré quelle donnée. Ainsi, une étape cruciale dans la tâche d'identification consiste à séparer les données de régression selon leur sous-modèle générateur respectif. Une fois que cette classification est réalisée, il ne reste plus qu'à recourir aux méthodes classiques d'identification linéaire [74] pour estimer l'ensemble des sous-modèles constitutifs du système hybride considéré. Cette approche a été suivie dans [46] et [87] pour l'identification des modèles affines par morceaux PWA (Piecewise Affine models), c'est-à-dire, des modèles définis sur une partition polyhédrale de l'espace de régression.

Une autre catégorie assez variée de méthodes alterne entre l'assignation des données de régression à des sous-modèles et l'estimation simultanée des paramètres en s'appuyant sur une technique d'apprentissage de poids [99], en résolvant un problème de partition minimum en sous-systèmes réalisables [18], ou encore en ayant recours à l'apprentissage bayésien [67]. Une faiblesse commune à toutes ces méthodes est qu'elles sont sous-optimales dans le sens où elles ne fournissent pas en général une solution optimale au problème de minimisation du critère mixte de classification-identification.

En considérant toujours des systèmes PWA, on distingue aussi l'approche basée sur l'optimisation mixte en nombres entiers. Dans [101], le problème d'identification est converti en un problème de pro-

Ordres connus	Nombre de sous-modèles connu	Références
oui	oui	[46], [101], [99], [67]
oui	non	[18], [87], [25], [131]
non	oui	[125]
non	non	[126], [77]

TABLE 1 – Classification des techniques d'estimation de modèles hybrides entrée-sortie selon les hypothèses que l'ordre et/ou le nombre de sous-modèles sont connus *a priori*.

grammation linéaire ou quadratique mixte en nombres entiers pour lequel des algorithmes de résolution efficaces existent [104], [33]. Ainsi, cette méthode garantit une solution optimale. En contrepartie de cette optimalité, elle souffre d'une complexité onéreuse qui en limite l'application à des systèmes de faibles dimensions.

Dans [131], une méthode algébrique est développée par R. Vidal *et al.* pour l'estimation de modèles commutants. Sous l'hypothèse que les données ne sont pas entachées de bruit, les auteurs reformulent le problème d'identification en un problème d'estimation puis de différentiation d'un polynôme homogène dont on peut déduire de manière élégante les sous-modèles du système sans itération (cf. aussi [77]). Un problème de cette méthode est que la dimension du vecteur de coefficients qui définit le polynôme homogène augmente exponentiellement en fonction des dimensions du système à estimer.

D'autres techniques [119], [58], [93], [22] ont été développées pour l'estimation de systèmes multivariable (MIMO) commutants, représentés par des modèles d'état. Ces méthodes consistent essentiellement dans l'application des méthodes des sous-espaces [72], [115] entre deux instants de commutation consécutifs. Notons toutefois que ces algorithmes s'appliquent sous l'hypothèse souvent restrictive que les changements de mode se produisent avec une certaine fréquence maximum.

De toutes les méthodes mentionnées ci-dessus, il émerge un constat essentiel : l'estimation des systèmes hybrides linéaires et multivariables n'ont pas encore reçu suffisamment d'attention. De plus, l'application de ces techniques requiert généralement que l'on dispose *a priori* du nombre de sous-modèles et/ou des ordres (cf. Tableau 1) dans le cas des modèles entrée-sortie. Lorsque les modèles estimés sont des modèles d'état, il est généralement supposé que les instants de commutation sont temporellement assez espacés et que les instants de commutations sont précisément connus [119]. Les articles [93] et [22] ne supposent pas la disponibilité des instants de commutations mais ne traitent pas en revanche du problème important de la *cohérence* des bases de représentation des différents sous-modèles.

3 Contributions

Dans ce travail de thèse, nous considérons principalement le problème de la modélisation et de l'identification de systèmes hybrides sous différentes hypothèses. Puisqu'un système dynamique hybride résulte de commutations entre un certain nombre de sous-systèmes, ce problème revient à inférer le nombre de ces sous-systèmes et leurs paramètres à partir uniquement d'une collection finie d'observations entrée-sortie du système, éventuellement bruitées.

La contribution de la thèse consiste essentiellement dans :

- la proposition d'algorithmes d'identification des ordres et des paramètres d'un système commutant décrit par un modèle d'état [9], [11]. En amont de ces méthodes d'estimation de modèles d'état commutants, nous développons un ensemble de nouveaux algorithmes d'identification de modèles linéaires d'état ayant la particularité attractive de permettre une fixation de la base de coordonnées de l'état du système tout en évitant l'usage de la traditionnelle Décomposition en Valeurs Singulières (DVS) [10].
- la généralisation de la méthode algébro-géométrique de R. Vidal à des modèles MIMO commutants du type ARX (SARX) sous les hypothèses très générales que le nombre de sous-modèles, les ordres et les paramètres de ces sous-modèles sont simultanément inconnus [12].
- la proposition d'une méthode récursive d'identification de modèles MIMO SARX alternant entre l'assignation des données aux sous-modèles et l'estimation de leurs paramètres [8].

Le manuscrit est structuré de la façon suivante.

Dans le Chapitre 1, nous faisons quelques brefs rappels sur la modélisation et l'identification de systèmes non linéaires. Nous présentons les systèmes hybrides linéaires ou affines par morceaux comme un moyen possible et par ailleurs intéressant de représenter des systèmes non linéaires complexes. Cela nous amène au problème général d'identification de systèmes hybrides et plus particulièrement de systèmes affines par morceaux (PWA).

Dans le Chapitre 2, nous présentons un état de l'art des méthodes d'identification de systèmes linéaires hybrides. Nous décrivons certaines méthodes bien connues dans la littérature tout en indiquant leurs avantages et leurs faiblesses respectives. Ces méthodes s'intéressent, pour la plupart, à l'estimation de modèles ARX par morceaux (PWARX), généralement mono-entrée mono-sortie. Cependant, plusieurs méthodes d'analyse de systèmes dynamiques hybrides MIMO étant basées sur des modèles d'état, il peut être parfois nécessaire d'estimer directement des modèles d'état à partir des données entrée-sortie.

Ainsi, le Chapitre 3 traite de l'identification par les méthodes des sous-espaces d'un seul modèle d'état linéaire. En effet, afin de mieux appréhender le problème de l'identification d'un système hybride linéaire qui est un ensemble de sous-systèmes linéaires, il paraît crucial de commencer par considérer le problème qui consiste à inférer un seul modèle linéaire à partir des données entrée-sortie. Après avoir mis en évidence l'inadéquation des méthodes des sous-espaces existantes pour traiter des systèmes hybrides linéaires, nous développons quelques nouveaux algorithmes pour l'identification *structurée* de systèmes linéaires décrits par des modèles d'état. Ces algorithmes offrent entre autres la possibilité de fixer la base de coordonnées de l'état et ne nécessitent pas l'utilisation de l'algorithme de DVS, connue pour être numériquement robuste mais onéreuse.

Dans le Chapitre 4, nous généralisons les résultats du Chapitre 3 à l'identification de systèmes à commutations. En amont de cette généralisation, nous posons le problème de la réalisation de systèmes à commutations sous forme de représentation d'état. Ensuite, contrairement aux algorithmes existants pour l'estimation de modèles d'état commutants, nous étudions le cas très ardu où aucun temps de séjour minimum n'est supposé entre les instants de commutation. Nous présentons alors, sous une hypothèse d'observabilité appropriée, un principe très générique de méthodes pour l'estimation de ces types de

modèles. Cependant, sans faire l'hypothèse que les instants de commutation sont séparés par un certain délai minimum (temps de séjour), la méthode développée est très difficilement applicable en pratique à cause de son coût d'implémentation. Ainsi, dans un objectif de simplification, nous faisons l'hypothèse d'un certain temps de séjour minimum du système hybride global dans chaque sous-système. Partant de cette hypothèse, nous proposons une méthode d'identification qui effectue en ligne les multiples tâches d'estimation des paramètres, de détection des instants de commutation et de classification des sous-modèles.

Le Chapitre 5 considère plutôt l'identification de systèmes commutants représentés par des modèles entrée-sortie. Nous présentons une extension de la méthode algébrique de R. Vidal [131] à l'identification de modèles MIMO commutants du type ARX, sans connaissance *a priori* ni des ordres (qui peuvent être différents d'un sous-modèle à un autre), ni du nombre de sous-modèles, ni des paramètres. Une telle situation aussi délicate n'a, à notre connaissance, pas encore été considérée dans la littérature excepté dans [77] et [126] où le modèle étudié est monovariable. Le couplage entre l'état discret et les paramètres est éliminé en formant un ensemble de contraintes polynomiales vérifiées par toutes les données de régression indépendamment de l'état discret auquel elles sont associées. Nous déterminons ensuite les ordres et le nombre de sous-modèles en exploitant certaines propriétés de rang des matrices de données. Un problème cependant est que le nombre de polynômes à estimer et le nombre de coefficients de ces polynômes augmentent exponentiellement avec le nombre d'états discrets et la dimension de l'espace de régression. Ainsi, nous discutons une alternative à cette méthode basée sur une alternance entre classification des données et estimation des paramètres.

Le Chapitre 6 présente, à titre illustratif, quelques résultats de simulation numérique. Nous procédons aussi à quelques tests de validation d'une partie des méthodes développées sur un procédé physique réel. Le système étudié est un système manufacturier de montage de composants électroniques fréquemment utilisé dans la littérature [66] pour la validation de méthodes d'estimation de modèles hybrides. Finalement, l'approche hybride est appliquée à la modélisation puis à l'identification d'un système non-linéaire défini par l'écoulement d'eau dans des tronçons de rivières (réseau hydrographique) [41]. Ce système, qui est en réalité continu et non-linéaire, est modélisé comme un système commutant entre un certain nombre de sous-systèmes linéaires. Nous montrons sur l'ensemble des résultats présentés que nos méthodes permettent d'obtenir des modèles d'une précision satisfaisante. Ces modèles sont donc exploitables par exemple à des fins de commande, de surveillance, de diagnostic, etc.

1

De la non-linéarité régulière à la linéarité par morceaux

Sommaire

Avant d'être formalisés plus récemment sous différentes formes [55], les modèles dynamiques hybrides apparaissent initialement dans les travaux de Chua et Kang [32, 71], Sontag [108], Breiman [25] sous forme essentiellement de modèles affines par morceaux (PWA) pour approximer des fonctions non-linéaires sur une partition polyhédrale (c'est-à-dire une partition constituée de régions délimitées par des hyperplans) de leur domaine. En comparaison avec les autres méthodes de modélisation de fonctions non-linéaires [106], les modèles affines par morceaux présentent l'avantage d'être moins complexes et donc plus faciles à exploiter. De plus, il est théoriquement démontré que ces types de modèles possèdent la propriété d'*approximateurs universels* [25] dans le sens où, en utilisant des fonctions PWA, il est possible d'approcher une fonction non-linéaire continue sur un certain compact avec une précision arbitraire.

Dans ce chapitre introductif, nous revisitons la modélisation et l'identification de systèmes dynamiques non-linéaires. Cela nous permettra de présenter brièvement la philosophie générale qui sous-tend toute technique de modélisation non-linéaire puis d'introduire l'approche dite multi-modèles. Plus précisément, nous présentons des multi-modèles affines c'est-à-dire, constitués d'une agrégation de sous-modèles affines. Ce sont des modèles, qui, de par leur construction mathématique et leur interprétation, sont très proches des modèles PWA qui seront définis plus formellement au Chapitre 2.

1.1 Modélisation et identification de systèmes non-linéaires

Etant donné un système physique, la modélisation de ce système désigne la procédure qui consiste à décider d'une structure paramétrée de modèle mathématique pour représenter le comportement de ce système. Dans cette démarche, aucune connaissance *a priori* des mécanismes physiques qui régissent le système considéré n'est supposée disponible. C'est l'approche dite « boîte noire » [106]. Il s'agit, à partir d'observations entrée-sortie collectées sur un système donné, potentiellement non-linéaire, de construire puis d'estimer une structure de modèle qui pourrait reproduire au mieux le comportement de ce système. On distingue principalement deux catégories de modèles utilisées pour représenter les systèmes dynamiques en temps discret :

- le modèle entrée-sortie qui est une relation directe entre la sortie $y(t) \in \mathbb{R}^{n_y}$ et les entrées et sorties passées du système

$$\begin{cases} y(t) = f(\varphi(t), \mu) + e(t), \\ \varphi(t) = \begin{bmatrix} y(t-1)^\top & \cdots & y(t-n_a)^\top & u(t)^\top & \cdots & u(t-n_b)^\top \end{bmatrix} \in \mathbb{R}^d, \end{cases} \tag{1.1}$$

  où $u(t) \in \mathbb{R}^{n_u}$ est l'entrée à l'instant t du système considéré, $d = n_a n_y + (n_b + 1)n_u$, f est une fonction à valeurs dans $\mathbb{R}^{n_y}$, μ est un un vecteur de paramètres et $e(t)$ symbolise une perturbation.
- le modèle d'état qui est une relation récurrente du type entrée-état-sortie

$$\begin{cases} x(t+1) = g(x(t), u(t), w(t), \alpha), & x(0) = x_0 \\ y(t) = h(x(t), u(t), v(t), \beta). \end{cases} \tag{1.2}$$

Ici, g et h sont des fonctions potentiellement non-linéaires, parametrées par les vecteurs α et β ; $v(t)$, $w(t)$ font référence à d'éventuels bruits. Le vecteur $x(t) \in \mathbb{R}^n$, appelé état du système, encode l'information relative au passé du système tandis que le vecteur $\varphi(t)$ qui est défini dans (1.1) et référencé comme le vecteur de régression, consiste en une collection des entrées et sorties passées jusqu'aux ordres n_a et n_b. Les entiers n_a et n_b pour le modèle (1.1) et n pour le modèle (1.2) sont des caractéristiques intrinsèques du système à modéliser et peuvent être regardés comme des mesures de la mémoire du système. Ils seront supposés finis.

La suite du chapitre est basée sur un modèle du type (1.1), dans lequel nous supposons, pour des raisons de simplicité, que $n_y = 1$. Toutefois, dans le cas où $n_y > 1$, en notant $f(\cdot) = \begin{bmatrix} f^1(\cdot) & \cdots & f^{n_y}(\cdot) \end{bmatrix}^\top$, où $f^j(\cdot)$ est la fonction non-linéaire qui définit la $j$ème sortie, les méthodes présentées peuvent s'appliquer à chacune des fonctions f^j individuellement.

A partir d'un nombre fini N d'observations entrée-sortie $\{\varphi(t), y(t)\}_{t=\bar{n}+1}^{N} \subset \mathbb{R}^d \times \mathbb{R}$, où $\bar{n} = \max(n_a, n_b)$, le but de l'identification est par exemple d'inférer la fonction f de l'équation (1.1). Dès lors, une question essentielle est : de quelle façon ces fonctions dépendent-elles de leurs entrées ? Autrement dit, quelle est la forme structurelle de ces fonctions ? L'idée centrale dans la plupart des méthodes d'approximation de fonctions non-linéaires [106] consiste à développer la fonction non-linéaire f du modèle (1.1) sur une base connue de fonctions élémentaires $\{f_j\}_{j=1}^{\infty}$.

$$f(\varphi, \mu) = \sum_{j=1}^{\infty} \mu_j f_j(\varphi). \tag{1.3}$$

La principale différence entre les approches rapportées dans la littérature réside dans le choix de la famille $\{f_j\}$ de fonctions élémentaires. Celles-ci peuvent être par exemple des polynômes, des fonctions-noyaux, de fonctions sigmoïdales, des fonctions à base radiale (RBF), des ondelettes etc. Le lecteur intéressé pourra se reporter par exemple à [106], [64] pour plus de commentaires sur les propriétés de ces différentes bases de fonctions de support.

En résumé, une philosophie dans la modélisation de systèmes non-linéaires consiste à décomposer le domaine de fonctionnement du système global en des sous-régions de fonctionnement dans lesquelles le comportement du système global peut être approché par des sous-modèles locaux simples. Le modèle global est alors envisagé comme une certaine combinaison linéaire de ces sous-modèles [61]. Cela s'apparente au principe suivant [61] : subdiviser un problème complexe en plusieurs sous-problèmes simples solvables séparément et dont les solutions remises bout à bout ou habilement combinées entre elles permettent de retrouver la solution du problème global. La plupart (sinon la totalité) des techniques de modélisation non-linéaire s'appuient sur cette idée fondamentale de décomposition de la non-linéarité en morceaux simples.

En pratique, on va substituer au modèle (1.3), un modèle tronqué c'est-à-dire, un développement sur un nombre fini de fonctions de base supposées connues *a priori*. On a alors

$$f(\varphi, \mu) = \sum_{j=1}^{s} \mu_j f_j(\varphi) = \mu^\top F(\varphi). \tag{1.4}$$

avec s un entier donné [1] *a priori*, $\mu = \begin{bmatrix} \mu_1 & \cdots & \mu_s \end{bmatrix}^\top \in \mathbb{R}^s$ le vecteur de poids et $F(\varphi) = [f_1(\varphi) \ \cdots \ f_s(\varphi)]^\top$. Ainsi, l'estimation de μ se réduit à un problème de moindres carrés standard, c'est-à-dire, la minimisation du critère

$$\mathcal{J}(\mu) = \sum_{t=\bar{n}+1}^{N} \big(y(t) - \mu^\top F(\varphi(t))\big)^2.$$

La solution d'un tel problème est donnée par

$$\hat{\mu}_N = \left(\sum_{t=\bar{n}+1}^{N} F(\varphi(t)) F(\varphi(t))^\top \right)^{-1} \sum_{t=\bar{n}+1}^{N} F(\varphi(t)) y(t). \tag{1.5}$$

Bien évidemment, dans le calcul de $\hat{\mu}_N$ par l'équation (1.5), nous supposons que la matrice $\sum_{t=\bar{n}+1}^{N} F(\varphi(t))F(\varphi(t))^\top$ est inversible, c'est-à-dire que $\text{rang}\left(\begin{bmatrix} F(\varphi(\bar{n}+1)) & \cdots & F(\varphi(N)) \end{bmatrix}\right) = s$. Du point de vue de l'identification, le fait de supposer la base $\{f_j\}_{j=1}^{s}$ entièrement connue est évidemment très commode. Cependant, cela comporte l'inconvénient de ne pas permettre l'ajustement des fonctions de la base paramétrée aux données. En général, cette base est construite à partir d'une fonction mère [106] munie de paramètres de dilatation et de translation. Ainsi, il serait judicieux de sélectionner une base qui soit non seulement en adéquation avec les caractéristiques topologiques de la non-linéarité à approximer mais aussi ajustable en fonction des données de mesure disponibles. Malheureusement dans ce cas, le problème d'identification nécessite une optimisation non-linéaire (généralement réalisée avec la méthode de descente du gradient, la méthode de Newton, la méthode de Marquardt, etc.) [110] difficile à résoudre efficacement.

Comme alternative à ces modèles, il existe des formes de modèles non paramétrées pour la régression non-linéaire. Dans la suite de ce chapitre, nous discutons deux techniques récentes portant sur cette classe de modèles.

1.1.1 Régression par les LS-SVM

La méthode de régression Least Squares-Support Vector Machines (LS-SVM) [111] est une variante des outils statistiques SVM [116]. Ces derniers sont très populaires dans la communauté de l'apprentissage automatique. Afin de présenter le principe de la régression par LS-SVM, considérons un ensemble de

1. En pratique, s peut être déterminé incrémentalement en partant de $s = 1$ jusqu'à ce que la précision du modèle (1.4) n'augmente plus de manière significative lorsqu'on continue à incrémenter s.

données $\{\varphi(t), y(t)\}_{t=\bar{n}+1}^{N} \subset \mathbb{R}^d \times \mathbb{R}$ générées par un modèle du type

$$y(t) = f(\varphi(t)) + e(t), \tag{1.6}$$

où $e(t)$ est un bruit de mesure, $e(t) \sim N(0, \sigma_e^2)$ et $f : \mathbb{R}^d \to \mathbb{R}$ est une fonction supposée être d'allure suffisamment lisse et régulière. On considère qu'il existe un espace $\mathbb{R}^{n_h}$ de caractéristiques, de dimension n_h élevée voire infinie dans lequel f peut être considéré comme linéaire par rapport à un vecteur dit de caractéristiques dépendant de l'entrée de f. Cela conduit au modèle

$$f(\varphi) = w^\top \eta(\varphi) + b, \tag{1.7}$$

où $\eta : \mathbb{R}^d \to \mathbb{R}^{n_h}$ est l'application qui projette les éléments de l'espace de régression dans l'espace des caractéristiques, $w \in \mathbb{R}^{n_h}$ est un vecteur de poids et $b \in \mathbb{R}$ est un biais. Alors, l'estimation de la fonction f à travers ses paramètres w et b peut être reformulée comme un problème d'optimisation sous contrainte [111] :

$$\begin{aligned} \min_{w,b,\bar{e}} \mathcal{J}(w, \bar{e}) &= \frac{1}{2} w^\top w + \frac{\gamma}{2} \sum_{t=\bar{n}+1}^{N} e(t)^2 \\ \text{sujet à} \quad & y(t) = w^\top \eta(\varphi(t)) + b + e(t),\ t = \bar{n}+1, \ldots, N, \end{aligned} \tag{1.8}$$

où $\bar{e} = \begin{bmatrix} e(\bar{n}+1) & \cdots & e(N) \end{bmatrix}^\top \in \mathbb{R}^{N-\bar{n}}$ et γ est un paramètre de régularisation. Selon la forme du critère $\mathcal{J}(w, \bar{e})$, le problème (1.8) vise à minimiser l'erreur du modèle (1.6) tout en garantissant une valeur raisonnable à la norme du vecteur w de (1.7). Le critère $\mathcal{J}(w, \bar{e})$ étant quadratique par rapport à w et $\bar{e}$, sa minimisation équivaut à celle du Lagrangien

$$\mathscr{L}(w, b, \bar{e}, \bar{\alpha}) = \mathcal{J}(w, \bar{e}) - \sum_{t=\bar{n}+1}^{N} \alpha(t)\big(w^\top \eta(\varphi(t)) + b + e(t) - y(t)\big),$$

où les $\alpha(t)$ sont les coefficients de Lagrange et $\bar{\alpha} = \begin{bmatrix} \alpha(\bar{n}+1) & \cdots & \alpha(N) \end{bmatrix}^\top \in \mathbb{R}^{N-\bar{n}}$. Les conditions nécessaires à l'obtention d'un minimum s'écrivent alors

$$\frac{\partial \mathscr{L}}{\partial w} = 0 \to w = \sum_{t=\bar{n}+1}^{N} \alpha(t)\eta(\varphi(t)), \tag{1.9}$$

$$\frac{\partial \mathscr{L}}{\partial b} = 0 \to \sum_{t=\bar{n}+1}^{N} \alpha(t) = 0, \tag{1.10}$$

$$\frac{\partial \mathscr{L}}{\partial e(t)} = 0 \to e(t) = \gamma^{-1}\alpha(t), \tag{1.11}$$

$$\frac{\partial \mathscr{L}}{\partial \alpha(t)} = 0 \to y(t) = w^\top \eta(\varphi(t)) + b + e(t), \tag{1.12}$$

En reportant les expressions (1.9) et (1.11) dans (1.12) et en tenant compte de (1.10), on obtient un système d'équations linéaires sous la forme

$$\left[\begin{array}{c|c} 0 & 1_N^\top \\ \hline 1_N & \Omega + \gamma^{-1} I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \bar{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right], \tag{1.13}$$

avec

$$1_N = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^\top \in \mathbb{R}^{N-\bar{n}}, \quad y = \begin{bmatrix} y(\bar{n}+1) & \cdots & y(N) \end{bmatrix}^\top \in \mathbb{R}^{N-\bar{n}},$$

et $\Omega \in \mathbb{R}^{N\times N}$ défini par $\Omega_{ij} = \eta(\varphi(i))^\top \eta(\varphi(j)) \ \forall\, i,j = 1,\ldots,N$. Pour résoudre le système d'équations (1.13), on a besoin de connaître les éléments de la matrice Ω qui apparaissent comme des produits scalaires canoniques dans l'espace de caractéristiques. Mais une évaluation directe nécessiterait une connaissance de η. Cependant, grâce au théorème de Mercer [116], Ω_{ij} peut être évalué par $\Omega_{ij} = K(\varphi(i), \varphi(j))$, où $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ est une certaine fonction noyau définie positive [105]. Cette fonction noyau peut être choisie de multiples façons. Des exemples sont le noyau à base radiale (RBF) du type Gaussien, $K(x,y) = \exp(-\|x-y\|_2^2/\sigma^2)$, le noyau polynomial de degré d, $K(x,y) = (1+x^\top y)^d$, le noyau du type perceptron multi-couches $K(x,y) = \tanh(\delta x^\top y + \mu)$ parametré par des constantes δ et μ [105].

La solution $\hat{f}_N$ au problème de régression s'obtient finalement en tout point $\varphi^* \in \mathbb{R}^d$ comme

$$\begin{aligned} \hat{f}_N(\varphi^*) &= \sum_{t=\bar{n}+1}^{N} \alpha(t) K(\varphi(t), \varphi^*) + b, \\ &= \bar{\alpha}^\top \eta_N(\varphi^*) + b \end{aligned} \tag{1.14}$$

où $\eta_N(\varphi^*) = \begin{bmatrix} K(\varphi(\bar{n}+1), \varphi^*) & \cdots & K(\varphi(N), \varphi^*) \end{bmatrix}^\top$ et α et b sont les solutions de l'équation (1.13). Une inspection de la solution (1.14) montre une analogie intéressante avec le modèle initial (1.7). On peut interpréter $\eta_N(\cdot)$ comme une estimée de la transformation $\eta(\cdot)$ basée sur N échantillons. Au vu des nombreuses possibilités de choix de la base, le concept de la méthode LS-SVM peut être regardé comme une approche unifiée. Suivant les différents choix du noyau $K(\cdot,\cdot)$, la relation (1.14) peut être assimilée à un développement de f sur une base de polynômes, de fonctions à base radiale (RBF), etc. Finalement, l'estimateur LS-SVM, à la régularisation près, est assez semblable à l'estimateur (1.5) des moindres carrés.

1.1.2 Optimisation directe de pondération

Considérons toujours un échantillon de données $\{\varphi(t), y(t)\}_{t=\bar{n}+1}^{N}$ générées par un modèle du type (1.6). Le bruit $e(t)$ est supposé être blanc Gaussien, de moyenne nulle et de variance σ_e^2 supposée connue. La méthode DWO (Direct Weight Optimization) [102] est une approximation non-paramétrique, basée

sur une estimation de la pondération optimale $w_{\varphi^*}(t)$, $t = \bar{n}+1, \ldots, N$, de façon à ce que le modèle [1]

$$\hat{f}(\varphi^*) = \sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t) y(t) \tag{1.15}$$

approxime au mieux la fonction f du modèle (1.6) au point φ^*. La fonction f est supposée appartenir à l'ensemble

$$\mathcal{F} = \left\{ f \in \mathcal{C}^1 : \|\nabla f(\varphi^* + h) - \nabla f(\varphi^*)\|_2 \leq L \|h\|_2 \, \forall \varphi^*, h \in \mathbb{R}^d \right\}$$

des fonctions continûment dérivables sur $\mathbb{R}^d$ et à gradient borné, au sens d'une certaine norme, par une constante Lipschitzienne L. La qualité de l'estimateur $\hat{f}$ pourrait être formulée comme la minimisation de l'espérance mathématique $\mathbb{E}\big(\big(f(\varphi^*) - \hat{f}(\varphi^*)\big)^2\big)$ du carré des erreurs. On définit ainsi le critère

$$W(f, \varphi^*, w_{\varphi^*}) = \mathbb{E}\Big(\big(f(\varphi^*) - \sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t) y(t)\big)^2\Big),$$

dans lequel $w_{\varphi^*} = \begin{bmatrix} w_{\varphi^*}(\bar{n}+1) & \cdots & w_{\varphi^*}(N) \end{bmatrix}^\top \in \mathbb{R}^{N-\bar{n}}$. Mais un problème est que $W(f, \varphi^*, w)$ dépend explicitement de f qui est inconnu. Donc pour procéder au calcul de $\hat{f}$, Roll *et al.* [102] proposent de suivre une approche mini-max c'est à dire, la minimisation de la borne supérieure de la fonction objectif (le pire cas). Cela revient à rechercher le vecteur de poids optimal $w^o_{\varphi^*}$ sous la forme

$$w^o_{\varphi^*} = \arg\min_{w_{\varphi^*}} \sup_{f \in \mathcal{F}} W(f, \varphi^*, w_{\varphi^*}). \tag{1.16}$$

En imposant aux poids de satisfaire la contrainte $\sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t) = 1$ et en exprimant linéairement $\varphi^* = \sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t)\varphi(t)$ comme le barycentre des couples $\{\varphi(t), w_{\varphi^*}(t)\}_{t=\bar{n}+1}^{N}$, on peut, en s'appuyant sur la définition de $\mathcal{F}$, montrer que (cf. [102])

$$W(f, x, w_{\varphi^*}) \leq \left(\frac{L}{2} \sum_{t=\bar{n}+1}^{N} |w_{\varphi^*}(t)| \, \|\varphi^* - \varphi(t)\|_2^2 \right)^2 + \sigma_e^2 \sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t)^2.$$

Comme conséquence de cela, le problème peut se réduire à une programmation quadratique du type

$$\min_{w_{\varphi^*}} \frac{L^2}{4} \left(\sum_{t=\bar{n}+1}^{N} |w_{\varphi^*}(t)| \, \|\varphi^* - \varphi(t)\|_2^2 \right)^2 + \sigma_e^2 \sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t)^2$$
$$\text{sujet à } \sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t) = 1, \; \sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t)\varphi(t) = \varphi^*$$

1. pour éviter des risques de confusion de la variable φ (de la fonction f) avec $\varphi(t)$, nous avons remplacé φ par φ^*.

qui, en remplaçant les $|w_{\varphi^*}(t)|$ par des variables auxiliaires $\rho(t) \geq |w_{\varphi^*}(t)|$, peut encore se reformuler sous la forme

$$\min_{w_{\varphi^*},\rho} \frac{L^2}{4}\left(\sum_{t=\bar{n}+1}^{N} \rho(t)\,\|\varphi^* - \varphi(t)\|_2^2\right)^2 + \sigma_e^2 \sum_{t=\bar{n}+1}^{N} \rho(t)^2$$
$$\text{sujet à } \rho(t) \geq \pm w_{\varphi^*}(t),\ \sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t) = 1,\ \sum_{t=\bar{n}+1}^{N} w_{\varphi^*}(t)\varphi(t) = \varphi^*$$

pour laquelle des méthodes de résolution efficaces existent [24]. L'approche DWO apparaît dans les travaux de Roll et al., [102], Nazin *et al.* [88], Bai *et al.* [7]. Dans [7], au lieu de minimiser la borne supérieure de l'erreur de simulation (1.16), il est proposé de rendre la plus petite possible la probabilité $\Pr\left(\left(f(\varphi^*) - \hat{f}(\varphi^*)\right)^2 \geq \delta\right)$ que cette erreur soit plus grande qu'un certain seuil prédéfini δ. Cette procédure conduit à l'obtention d'un estimateur optimal. Cependant, le modèle résultant de cette procédure est un modèle non-paramétrique complexe qui peut être physiquement difficile à interpréter.

1.2 Approche multi-modèles

Nous avons vu que les estimateurs non-paramétriques LS-SVM et DWO permettaient d'approcher efficacement la fonction non-linéaire f qui définit le modèle dynamique (1.6). On peut noter qu'ils partagent la caractéristique d'être des sommes pondérées de quantités élémentaires qu'on peut interpréter comme la contribution de chaque observation.

Cependant, une critique qu'on pourrait faire aux estimateurs LS-SVM et DWO est qu'ils ne sont pas assez transparents dans le sens où leur interprétabilité est limitée. Et pourtant, la finalité d'un modèle est de servir d'outil d'analyse et d'interprétation du système modélisé. De ce fait, le modèle doit, tout en étant assez précis, rester simple à appréhender. Mais il se trouve justement que la plupart des outils efficaces d'analyse ou de commande concernent des modèles linéaires. Il est donc naturel d'envisager une approximation du système non-linéaire (1.6) comme la combinaison linéaire de plusieurs modèles locaux linéaires, chaque modèle local se rapportant à une région spécifique du domaine de fonctionnement du système. Cette idée, connue dans la littérature comme approche multi-modèles, a fait l'objet de recherches soutenues [86], [62], [117]. Si l'approximation multi-modèles est attractive, il demeure en revanche des questions cruciales à savoir combien de modèles locaux sont nécessaires à une modélisation précise et comment estimer ces modèles. Nous verrons ici successivement la forme paramétrique et non paramétrique des multi-modèles encore appelés modèles de Takagi-Sugeno [112].

1.2.1 Forme paramétrique de multi-modèles affines

D'un point de vue mathématique, on peut regarder les différents modèles locaux ou sous-modèles comme résultant de linéarisations du système global autour de certains points de fonctionnement. En supposant que la fonction f définie par l'équation (1.6) appartient à la classe $\mathcal{C}^1$ des fonctions continûment dérivables, le développement de Taylor d'ordre un de f en un point c_j du domaine $\mathscr{X} \subset \mathbb{R}^d$ de f (supposé

être un compact) est donné par

$$f(\varphi) = f(c_j) + \nabla f(c_j)^\top (\varphi - c_j) + \varepsilon(c_j, \varphi), \tag{1.17}$$

où ∇f est le gradient de f, $\varepsilon(c_j, \varphi)$ fait référence aux termes d'ordre supérieur à un, avec $\lim_{\varphi \to c_j} \varepsilon(c_j, \varphi) = 0$. Pour un φ dans la boule de voisinage $B(c_j, \delta) = \{\varphi \in \mathscr{X} : \|\varphi - c_j\|_2 < \delta\}$ de c_j (où $\delta > 0$ est relativement petit), le terme non-linéaire $\varepsilon(c_j, \varphi)$ peut être négligé. Ainsi, f peut être approximé par la partie linéaire de (1.17). En suivant ce principe, $\mathscr{X}$ peut être décomposé en un ensemble de s régions entrelacées $B(c_j, \delta)$, $j = 1, \cdots, s$ sur lesquelles f est approché par un modèle local linéaire. Avec

$$\theta_j = \begin{bmatrix} \nabla f(c_j)^\top & f(c_j) - \nabla f(c_j)^\top c_j \end{bmatrix}^\top \in \mathbb{R}^{d+1},$$
$$\bar{\varphi}(t) = \begin{bmatrix} \varphi(t)^\top & 1 \end{bmatrix}^\top \in \mathbb{R}^{d+1},$$

on peut écrire

$$f(\varphi(t)) \simeq \theta_j^\top \bar{\varphi}(t) \quad \text{pour} \quad \varphi(t) \in B(c_j, \delta).$$

Si $\varphi(t)$ appartient à l'intersection de plusieurs régions, il peut être nécessaire de compter dans $f(\varphi(t))$ la contribution de chacun des sous-modèles correspondants. En conséquence, nous considérons le modèle suivant :

$$f(\varphi(t)) = \sum_{j=1}^{s} \mu_j(t) \Big(\theta_j^\top \bar{\varphi}(t) \Big), \tag{1.18}$$

dans lequel s est le nombre de sous-modèles, θ_j le vecteur de paramètres du j-ème sous-modèle dont $\mu_j(t)$ est le poids. La pondération est telle que

$$0 \leq \mu_j(t) \leq 1 \quad \text{et} \quad \sum_{j=1}^{s} \mu_j(t) = 1, \; t = \bar{n} + 1, \cdots, N.$$

Un choix classique des fonctions poids est la forme RBF gaussienne [117]

$$\mu_j(t) = \frac{r_j(t)}{\sum_{i=1}^{s} r_i(t)}, \tag{1.19}$$
$$r_j(t) = \exp\Big(- \|\varphi(t) - c_j\|_{\Sigma_j}^2 \Big), \tag{1.20}$$

où $\|x\|_{\Sigma_j}^2 = x^\top \Sigma_j x$, Σ_j est une matrice définie positive que nous choisissons sous une forme diagonale $\Sigma_j = \text{diag}(w_j^1, \cdots, w_j^d)$. Les c_j sont appelés centres des fonctions RBF et peuvent, d'une certaine manière, être regardés ici comme les points de linéarisation.

L'apprentissage des paramètres du modèle (1.18) repose généralement sur deux types de critères : le critère global et le critère local [86], [133]. Le premier tente aveuglement de minimiser l'erreur globale entre le modèle et les données sans tenir compte de l'interprétabilité locale du modèle résultant. Par conséquent, la structure pondérée du modèle (1.18) n'est pas assez reflétée dans son estimée. Contrairement au critère

global, le critère d'apprentissage local présente l'avantage de renforcer les aptitudes et les qualités locales du modèle qui en résulte.

Ici, nous aimerions proposer une technique d'estimation du modèle (1.18) qui renforce le sens local (dans l'espace de régression $\mathbb{R}^d$ considéré muni d'une certaine norme) des différentes composantes de la structure multi-modèles. De cette manière, on peut espérer que le modèle obtenu soit plus facile à interpréter physiquement. Pour cela, nous formulons le critère suivant

$$\mathcal{J}\left(\{\theta_j, c_j, \Sigma_j\}_{j=1}^{s}\right) = \sum_{t=\bar{n}+1}^{N}\sum_{j=1}^{s}\mu_j(t)\left(\left(y(t)-\theta_j^\top\bar{\varphi}(t)\right)^2 + \gamma\left\|\varphi(t)-c_j\right\|_{\Sigma_j}^2\right), \tag{1.21}$$

avec γ un paramètre de régularisation. Ce critère vise à trouver un compromis entre la classification des vecteurs de régression $\varphi(t)$ autour des centres c_j et la minimisation de l'erreur entre la sortie du j-ème sous-modèle et la sortie mesurée sur le système.

Il doit être clair que même s'il existe un fondement théorique solide [112] quant à la possibilité d'approcher la fonction non-linéaire f avec une précision arbitraire en utilisant un modèle de la forme (1.18), l'estimation de ses sous-modèles constitutifs reste une tâche délicate. En effet, le modèle (1.18) a une identifiabilité assez limitée puisqu'il pourrait souffrir d'un risque de compensation entre les différents éléments impliqués dans la somme [62]. Plusieurs approches existent dans la littérature [118], [23], [28] pour l'estimation du modèle (1.18).

Dans le but de trouver un ensemble de paramètres $\{\theta_j, c_j, \Sigma_j\}_{j=1}^{s}$ qui minimise le coût (1.21), nous suivons ici une procédure cyclique et itérative, similaire à l'algorithme de classification des K-means [40]. Il est important de noter qu'un tel algorithme n'est que sous-optimal dans le sens où sa convergence vers un optimum est fortement subordonnée à son initialisation.

Etant donné les observations entrée-sortie $\{u(t), y(t)\}_{t=1}^{N}$, nous construisons l'ensemble $\{\varphi(t), y(t)\}_{t=\bar{n}+1}^{N}$, avec $\bar{n} = \max(n_a, n_b)$, à partir de la définition (1.1) de $\varphi(t)$ et des ordres n_a et n_b supposés connus. Alors, les étapes d'itération de notre algorithme sont décrites comme suit :

(a) *Initialisation :* Choisir γ, s et initialiser les paramètres $\{\theta_j, c_j, \Sigma_j\}_{j=1}^{s}$. Il peut être nécessaire d'utiliser des heuristiques élaborées pour réaliser l'initialisation de ces paramètres.

(b) *Mise à jour des poids :* Calculer les poids $\mu_j(t)$ en utilisant les formules (1.19) et (1.20).

(c) *Adaptation des c_j :* Etant donné les paramètres θ_j, nous pouvons mettre à jour les centres en exploitant les conditions d'ordre un, nécessaires à l'obtention d'un minimum.
En faisant

$$\frac{\partial\mathcal{J}}{\partial c_j} = \sum_{t=\bar{n}+1}^{N}\left(\frac{\partial\mu_j(t)}{\partial c_j}J_j(t) + \mu_j(t)\frac{\partial J_j(t)}{\partial c_j}\right) = 0,$$

il vient

$$c_j = \frac{\sum_t \alpha_j(t)\varphi(t)}{\sum_t \alpha_j(t)},\ j = 1,\ldots,s, \tag{1.22}$$

avec

$$\alpha_j(t) = \mu_j(t)\Big(1 - \frac{1}{\gamma}(1-\mu_j(t))J_j(t)\Big),$$
$$J_j(t) = \big(y(t) - \theta_j^\top \bar{\varphi}(t)\big)^2 + \gamma \left\|\varphi(t) - c_j\right\|^2_{\Sigma_j}.$$

(d) *Mise à jour des Σ_j :* On écrit de nouveau la première condition dont la réalisation est nécessaire à l'obtention d'un minimum

$$\frac{\partial \mathcal{J}}{\partial w_j} = \sum_{t=\bar{n}+1}^{N} \left(\frac{\partial \mu_j(t)}{\partial w_j} J_j(t) + \mu_j(t) \frac{\partial J_j(t)}{\partial w_j} \right) = 0$$

ce qui donne, après développement,

$$w_j = \left(\sum_{t=\bar{n}+1}^{N} \delta_j(t) z_j(t) z_j(t)^\top \right)^{-1} \sum_{t=\bar{n}+1}^{N} \lambda_j(t) z_j(t),\ j = 1, \ldots, s,$$

avec

$$\delta_j(t) = \mu_j(t) \left(\gamma - (1-\mu_j(t)) \big(y(t) - \theta_j^\top \bar{\varphi}(t)\big)^2 \right),$$
$$\lambda_j(t) = \gamma \mu_j(t) \big(1 - \mu_j(t)\big),$$
$$z_j(t) = (\varphi(t) - c_j) \odot (\varphi(t) - c_j),$$

le symbole $\odot$ désignant le produit élément par élément de deux vecteurs (produit de Hadamard).

(e) *Mise à jour des θ_j :* Répéter l'étape *(b)* en utilisant les nouveaux centres calculés aux étapes *(c)* et *(d)* avant de calculer θ_j, $j = 1, \ldots, s$. De la relation

$$\frac{\partial \mathcal{J}}{\partial \theta_j} = \sum_{t=\bar{n}+1}^{N} \mu_j(t) \left(y(t) - \theta_j^\top \bar{\varphi}(t) \right) \bar{\varphi}(t) = 0,$$

on obtient, sous réserve de l'inversibilité de $\Phi D_j \Phi^\top$,

$$\theta_j = \left(\Phi D_j \Phi^\top \right)^{-1} \Phi D_j Y,\ j = 1, \ldots, s, \tag{1.23}$$

où

$$D_j = \operatorname{diag}\big(\mu_j(\bar{n}+1), \cdots, \mu_j(N)\big) \in \mathbb{R}^{(N-\bar{n})\times(N-\bar{n})},$$
$$\Phi = \Big[\bar{\varphi}(\bar{n}+1), \cdots, \bar{\varphi}(N)\Big] \in \mathbb{R}^{(d+1)\times(N-\bar{n})},$$
$$Y = \Big[y(\bar{n}+1), \cdots, y(N)\Big]^\top \in \mathbb{R}^{(N-\bar{n})}.$$

(f) Aller à l'étape *(b)* jusqu'à la convergence. Nous dirons que l'algorithme a convergé lorsque

$\left\|\Theta^{(r+1)} - \Theta^{(r)}\right\|_2 < \varepsilon \left\|\Theta^{(r)}\right\|_2$, pour un certain $\varepsilon > 0$, avec Θ un vecteur formé de tous les vecteurs du type $\begin{bmatrix}\theta_j^\top & c_j^\top & w_j^\top\end{bmatrix}^\top$, $j = 1, \ldots, s$, et r un compteur du nombre d'itérations.

Comme formulé par l'équation (1.22), le centre c_j peut être vu comme la moyenne des régresseurs $\varphi(t)$ pondérés par les $\alpha_j(t)$. Quant à l'expression (1.23) de θ_j, elle s'apparente à l'estimateur des moindres carrés pondérés. On peut noter que si les fonctions poids étaient connues, l'estimation des paramètres θ_j qui représentent les modèles locaux ne serait plus qu'un problème de régression linéaire.

1.2.2 Forme non-paramétrique de multi-modèles affines

Dans la sous-section 1.2.1, on a vu que l'estimation du modèle (1.18) nécessitait une optimisation à la fois sur les centres c_j des fonctions RBF, les matrices Σ_j et sur les paramètres θ_j avec fixation *a priori* du nombre de modèles locaux. Cette optimisation non-linéaire s'est avérée extrêmement ardue et parfois inefficace puisque l'optimalité des estimées n'est pas garantie. Pour limiter l'ampleur de ce problème, on pourrait par exemple assigner systématiquement (ou en recourant à une heuristique appropriée), des valeurs aux c_j et aux Σ_j, $j = 1, \cdots, s$. Etant donné les paramètres c_j et Σ_j ainsi fixés, les θ_j peuvent être obtenus de façon optimale puisque le coût est linéaire par rapport à ces derniers.

Si nous regardons le modèle (1.18) comme résultant de la superposition de plusieurs approximations linéaires locales (autour des c_j), alors on peut envisager l'approche suivante. Plutôt que de considérer seulement le comportement local dans le voisinage de quelques points (qui seraient idéalement distribués), nous allons procéder à une estimation autour de chaque point de mesure disponible. Cette approche a déjà été utilisée dans [46] en amont d'une procédure d'identification de systèmes affines par morceaux.

Pour commencer, nous définissons $\mathcal{X}_t = \left\{k_t^1, \cdots, k_t^{n_v}\right\}$ comme l'ensemble qui collectionne dans l'ordre croissant, l'indice t de $\varphi(t)$ et les indices des $n_v - 1$ plus proches voisins de $\varphi(t)$ au sens de la distance euclidienne. Le choix du nombre n_v joue un rôle déterminant à la fois dans la précision recherchée pour le modèle à estimer et dans la capacité de celui-ci à être généralisé. Soit

$$\Phi_t = \begin{bmatrix}\bar{\varphi}(k_t^1) & \cdots & \bar{\varphi}(k_t^{n_v})\end{bmatrix} \quad \text{et} \quad \mathcal{Y}_t = \begin{bmatrix}y(k_t^1) & \cdots & y(k_t^{n_v})\end{bmatrix}^\top.$$

Dans le voisinage de $\varphi(t)$ la fonction f peut être approchée par l'hyperplan [1] $y = \theta(t)^\top \varphi^*$ tangent à l'hypersurface d'équation $y = f(\varphi^*)$:

$$\theta(t) = \left(\Phi_t \Phi_t^\top\right)^{-1} \Phi_t \mathcal{Y}_t.$$

Le modèle global peut alors être écrit comme suit :

$$\hat{f}_N(\varphi^*) = \sum_{t=\bar{n}+1}^{N} \mu(\varphi^*, t)\left(\theta(t)^\top \bar{\varphi^*}\right), \tag{1.24}$$

avec $\bar{\varphi^*} = \begin{bmatrix}\varphi^{*\top} & 1\end{bmatrix}^\top$ et $\mu(\varphi^*, t)$, une certaine fonction rapidement décroissante du terme $\|\varphi^* - \varphi(t)\|_2$.

1. Nous avons remplacé ici la variable générique φ de la la fonction f pour éviter des risques de confusion avec $\varphi(t)$.

Comme dans le cas précédent, nous pouvons choisir des fonctions poids sous forme d'une RBF gaussienne que nous pourrons ensuite normaliser de façon à ce que leur somme vaille un : $\mu(\varphi^*,t)=\dfrac{r(\varphi^*,t)}{\sum_{i=1}^{s} r(\varphi^*,t)}$, avec $r(\varphi^*,t)=\exp\big(-\frac{\|\varphi^*-\varphi(t)\|_2^2}{\sigma^2}\big)$. Cependant, lorsque l'échantillon de donnée $\{\varphi(t)\}_{t=\bar{n}+1}^{N}$ est distribué sur le domaine d'intérêt $\mathscr{X}$ d'une manière suffisamment dense et uniforme, on peut juste poser

$$\mu(\varphi^*,t)=\begin{cases}1 & \text{si} \quad t=t_{\varphi^*}\\ 0 & \text{sinon}\end{cases}$$

avec $t_{\varphi^*}=\arg\min_{t=\bar{n}+1,\dots,N}\|\varphi^*-\varphi(t)\|_2$. Dans le cas où t_{φ^*} n'est pas déterminé de façon unique par cette expression, on peut prendre simplement t_{φ^*} comme un élément quelconque de l'ensemble $\big\{t_\varphi : \|\varphi(t_\varphi)-\varphi(t)\|_2=\min_{t=\bar{n}+1,\dots,N}\|\varphi^*-\varphi(t)\|_2\big\}$. Ainsi, on obtient

$$\hat{f}_N(\varphi^*)\simeq\theta(t_{\varphi^*})^\top\bar{\varphi^*}. \tag{1.25}$$

Ainsi, on peut approximer la fonction non-linéaire f grâce à une commutation entre un certain nombre de modèles affines. L'équation (1.25) est un modèle non-paramétrique au même titre que les modèles LS-SVM (1.14) et DWO (1.15) mais avec l'avantage que les sous-modèles sont affines et donc plus faciles à appréhender.

1.2.3 Exemple numérique

Nous illustrons dans cette sous-section, les performances des méthodes discutées à la section 1.2. Dans ce but, nous empruntons à l'article [118], le système entrée-sortie suivant

$$y(t)=\frac{y(t-1)y(t-2)\big(y(t-1)+4.5\big)}{1+y(t-1)^2+y(t-2)^2}+u(t-1). \tag{1.26}$$

Dans ce dernier article, cet exemple a été utilisé pour valider une méthode d'estimation de multi-modèles consistant en agrégation de sous-modèles d'état affines.

Avec une entrée d'excitation choisie comme un signal du type carrés multiples (cf. Figure 1.1), nous générons un échantillon de 1000 données dont 500 pour l'estimation d'un modèle et les 500 autres pour la validation de ce modèle. Le bruit $e(t)$ est tel que le Rapport Signal sur Bruit (RSB) vaille 30 dB. Le paramètre de réglage γ du critère (1.21) est choisi égal à 5. Nous obtenons alors à la figure 1.2 les sorties du multi-modèle paramétrique (1.18) (cf. Sous-section 1.2.1) en fixant le nombre de sous-modèles respectivement à 1 puis à 2. Le critère [18]

$$\text{FIT}=\left(1-\frac{\|\hat{y}-y\|_2}{\|y-\bar{y}\|_2}\right)\times 100\%$$

est utilisé pour mesurer la similitude entre la sortie $y=\begin{bmatrix}y(1) & \cdots & y(N)\end{bmatrix}^\top\in\mathbb{R}^N$ mesurée sur le système réel et la sortie $\hat{y}=\begin{bmatrix}\hat{y}(1) & \cdots & \hat{y}(N)\end{bmatrix}^\top\in\mathbb{R}^N$ reconstruite à partir du modèle. Dans cette formule, $\bar{y}$

est la moyenne des mesures $y(t)$, $t = 1, \ldots, N$, où N est le nombre de points de mesure disponibles.

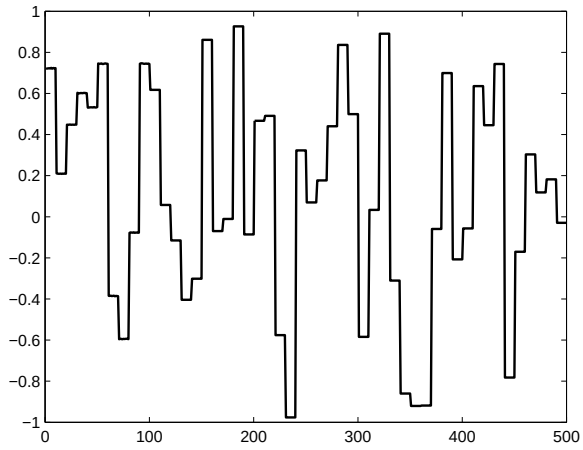
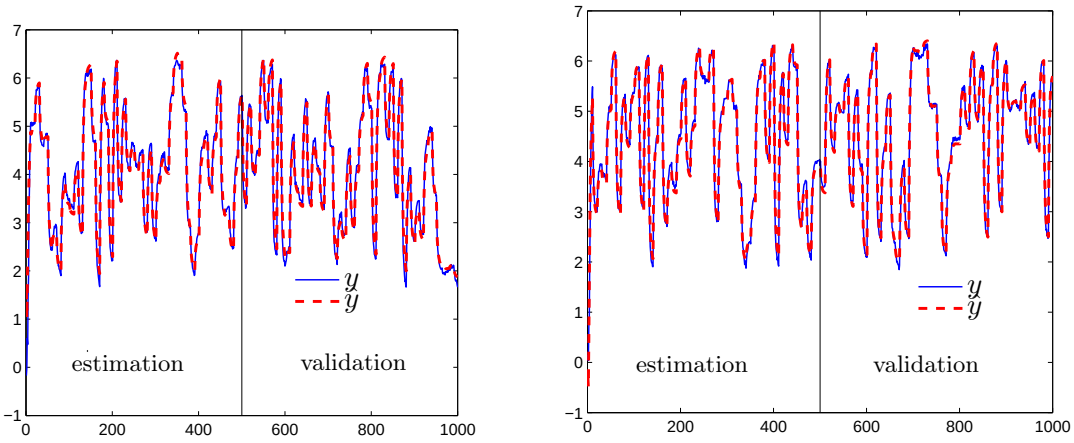


FIGURE 1.1 – Entrée d'excitation ayant servi à la génération des données d'identification.



(a) 1 sous-modèle : FIT = 85%.

(b) 2 sous-modèles : FIT = 91%.

FIGURE 1.2 – Identification et validation d'un système non-linéaire : la sortie $\hat{y}$ du multi-modèle (pointillés rouge) est une bonne approximation de la sortie système réel y (trait plein bleu).

Dans une deuxième expérience, nous considérons pour l'approximation du modèle non-linéaire (1.26), l'identification d'un multi-modèle non paramétrique de la forme (1.25). Pour cela, l'entrée d'excitation est prise comme précédemment. Un bruit de sortie est additionné sur la sortie de façon à réaliser un RSB de 30 dB. Nous utilisons alors les 100 premières données pour l'estimation, avec $n_v = 30$, et les 900 données restantes pour la validation. Les résultats de cette expérience sont représentés sur la figure 1.3. Ceux-ci montrent la capacité d'un modèle de la forme (1.25) à approximer le système (1.26).

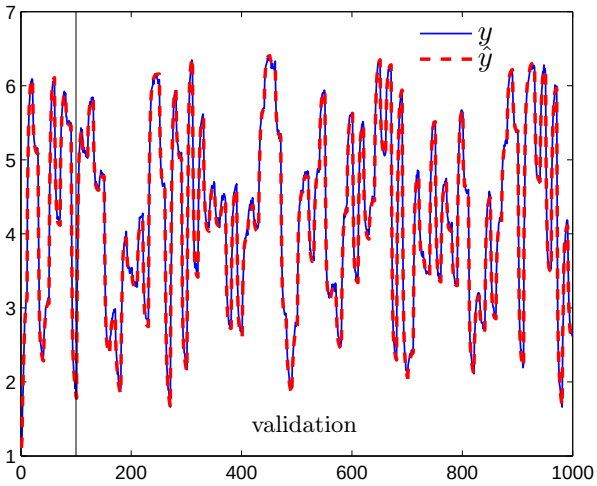


FIGURE 1.3 – Sortie réelle (trait plein) et sortie reconstruite (pointillé) à partir d'un multi-modèle non paramétrique du type (1.25) : FIT = 94%.

1.3 Conclusion

Dans ce chapitre, nous avons étudié le problème qui consiste à inférer le modèle mathématique, potentiellement non-linéaire, qui pourrait expliquer au mieux un échantillon de données $\{\varphi(t), y(t)\}_{t=\bar{n}+1}^{N}$. Sans rentrer dans les détails algorithmiques des différentes méthodes existantes dans la littérature, nous avons résumé l'idée centrale qui inspire toute technique de modélisation non-linéaire. Nous avons vu qu'une non-linéarité régulière pouvait être efficacement approximée par un développement fini sur une base de fonctions non-linéaires élémentaires connues *a priori* ou non. Partant de ce principe, nous avons insisté sur l'intérêt de développer la fonction non-linéaire sur une base de fonctions linéaires ou affines : c'est l'approche multi-modèle. L'avantage manifeste de ce choix est qu'on pourra ainsi disposer d'un modèle relativement simple et plus exploitable c'est-à-dire plus facile à analyser et à interpréter. Cette affirmation tire sa justification dans le fait qu'il existe une théorie très développée pour l'analyse de systèmes linéaires. Ainsi, deux méthodes d'identification de formes paramétrique et non-paramétrique de multi-modèles affines ont été discutées et validées sur un exemple de simulation.
Dans le prochain chapitre, nous considérons le cas particulier de multi-modèles affines où la pondération des sous-modèles est binaire.

2

Identification de systèmes hybrides : un état de l'art

Sommaire

2.1 Introduction

Nous avons vu dans le chapitre précédent qu'un système non-linéaire continu peut être modélisé par l'agrégation d'un nombre fini de modèles locaux linéaires ou affines. Dans le cas particulier où la pondération de cette agrégation est binaire, le système global ainsi modélisé peut être regardé comme commutant entre les différents sous-systèmes considérés.

En fait, il existe dans la nature, des systèmes dynamiques dans lesquels interagissent à la fois des phénomènes de nature continue et événementielle. Ces systèmes sont connus sous la dénomination de systèmes dynamiques hybrides. Une illustration du principe de fonctionnement d'un système hybride est présentée [1] sur la figure 2.1.

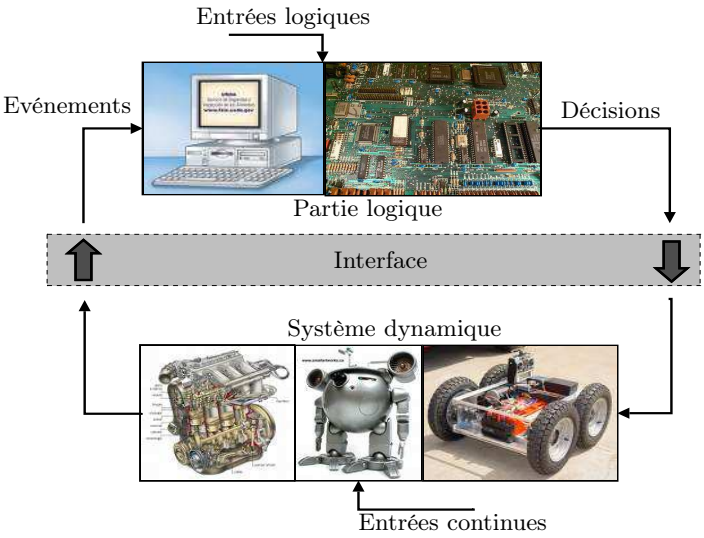


FIGURE 2.1 – Schéma de principe d'un système hybride : interaction entre composants logiques et processus continus.

Les exemples de systèmes hybrides sont abondants dans la vie réelle : les systèmes autonomes ou intelligents de façon générale, les systèmes de contrôle automatique de la vitesse automobile, le pilote automatique dans la navigation aérienne, le phénomène biologique de multiplication et de division de cellules, les robots d'exploration spatiale, etc. En somme, les systèmes hybrides s'apparentent à tout système comportant des phases, des états, des transitions, des modes. La loi de planification ou de coordination des différents changements de modes du système global, peut être une entrée exogène ou événementielle, elle peut aussi être fonction de l'état du système ou du temps, déterministe ou même totalement aléatoire.

Il existe dans la littérature une variété de modèles hybrides qui sont utilisés pour décrire des processus physiques dont le comportement peut être perçu comme étant à la fois continu et événementiel. Quelques exemples de modèles sont les modèles MLD (Mixed Logical Dynamical) [19], les modèles LC (Linear Complementarity) [56, 114], les modèles ELC (Extended Linear Complementarity) [36, 37], les modèles MMPS (Max-Min-Plus-Scaling) [37], les modèles PWA (PieceWise Affine) [108]. Bien qu'il existe parfois

1. Ce schéma s'inspire d'une illustration intéressante : http ://www.dii.unisi.it/ bemporad/.

des équivalences entre ces différentes formes de représentation [55], chacune d'elles possède des propriétés spécifiques qui lui sont propres. Ainsi, pour décrire un système physique donné, le choix de la classe de modèle appropriée pourra se faire en fonction de l'étude que l'on souhaite faire de ce système.

Bien souvent, la notion de système hybride peut apparaître comme une pure abstraction de modélisation. Par exemple, pour représenter certains systèmes non linéaires complexes (continus), il peut être préférable, comme nous l'avons antérieurement indiqué dans le Chapitre 1, de décomposer leur domaine de fonctionnement en plusieurs sous-régions sur lesquelles le système global peut être approché par un modèle linéaire ou affine. Ainsi, le système non-linéaire considéré qui est en fait un système continu, devient par modélisation, un système hybride commutant entre des sous-modèles linéaires.

Dans ce chapitre, nous faisons un état de l'art détaillé des méthodes existantes dans la littérature pour l'identification de systèmes hybrides. Un autre état de l'art sur ce sujet peut être trouvé dans [90]. La majorité de ces méthodes sont dédiées à la classe des systèmes affines par morceaux (PWA), c'est-à-dire des systèmes hybrides pour lesquels les commutations entre les sous-modèles sont déterminées par une partition polyhédrale de l'espace état-entrée (ou de l'espace de régression lorsqu'il s'agit d'un modèle entrée-sortie) [46], [67]. Concrètement, les méthodes existantes s'intéressent généralement à l'estimation d'un modèle de régression du type ARX par morceaux (PWARX). Cependant, il existe aussi quelques méthodes pour l'identification de systèmes commutants représentés par des modèles d'état [93], [119], [58], [22]. Ces dernières méthodes requièrent généralement l'hypothèse importante que les instants de commutation soient séparés par une certaine période minimum.

Le plan du chapitre est le suivant. Dans la section 2.2, nous présentons quelques méthodes développées dans la littérature pour l'estimation de modèles PWARX (PieceWise ARX). Nous décrivons plus particulièrement l'approche à base de classification [46], la méthode de l'erreur bornée [18], la méthode bayésienne [67]. Nous mentionnons aussi au passage les méthodes présentées dans [99], [87], [85], [107]. La section 2.3 est consacrée à la présentation de quelques méthodes pour l'estimation de modèles du type HHARX (Hinging Hyperplanes ARX). Après avoir rappelé rapidement le principe de l'algorithme HFA de Breiman [25], nous donnons un aperçu de la méthode d'optimisation mixte en nombres entiers [101]. Finalement, dans la section 2.4, nous présentons des travaux relatifs à l'identification de modèles linéaires commutants du type entrée-sortie SARX (Switched ARX) [131], [77] et du type représentation d'état SLSS (Switched Linear State Space models) [58], [119].

Notons que les modèles affines par morceaux PWARX constituent une classe particulière de modèles hybrides pour lesquels les sous-modèles affines sont définis sur une partition polyhédrale du domaine des régresseurs. Les modèles HHARX peuvent être regardés comme formant une sous-classe des modèles PWARX. La particularité de ces derniers modèles (HHARX) est que l'application affine par morceaux qui définit la relation entrée-sortie est continue alors qu'elle peut être discontinue dans le cas général. Quant aux modèles SARX ou SLSS, ils font référence à une classe plus large de modèles hybrides pour lesquels le mécanisme qui commande le passage d'un sous-modèle à un autre peut être quelconque.

2.2 Identification de modèles PWARX (PieceWise ARX)

Un système dynamique hybride, affine par morceaux du type ARX (PWARX) [46], est un modèle entrée-sortie défini de la façon suivante

$$y(t) = f(\varphi(t)) + e(t), \tag{2.1}$$

où $y(t) \in \mathbb{R}$ désigne la sortie du système. Le vecteur de régression $\varphi(t)$, supposé appartenir à un certain polyèdre borné $\mathscr{X}$ de $\mathbb{R}^d$, est défini par

$$\varphi(t) = \begin{bmatrix} y(t-1) & \cdots & y(t-n_a) & u(t)^\top & u(t-1)^\top & \cdots & u(t-n_b)^\top \end{bmatrix}^\top, \tag{2.2}$$

où $u(t) \in \mathbb{R}^{n_u}$ est l'entrée du système, n_a et n_b sont ses ordres et $d = n_a + n_u(n_b+1)$ est la longueur du vecteur $\varphi(t)$. Dans l'équation (2.1), $e(t)$ est l'erreur de prédiction, supposée être une séquence indépendante et identiquement distribuée (i.i.d.) de moyenne nulle et f est une fonction affine par morceaux définie comme suit :

$$f(\varphi) = \begin{cases} \theta_1^\top \bar{\varphi} & \text{si} \quad \varphi \in \mathscr{X}_1 \\ \quad \vdots & \\ \theta_s^\top \bar{\varphi} & \text{si} \quad \varphi \in \mathscr{X}_s, \end{cases} \tag{2.3}$$

avec $\bar{\varphi} = \begin{bmatrix} \varphi^\top & 1 \end{bmatrix}^\top$ et $\theta_i \in \mathbb{R}^{d+1}$. Les régions $\{\mathscr{X}_i\}_{i=1}^s$ réalisent une partition polyhédrale du domaine fermé et borné $\mathscr{X}$, c'est-à-dire :

$$\mathscr{X} = \cup_{i=1}^s \mathscr{X}_i \quad \text{et} \quad \mathscr{X}_i \cap \mathscr{X}_j = \emptyset, \text{ pour } i \neq j. \tag{2.4}$$

Le modèle (2.1) est dit PieceWise Auto-Regressive eXogenous (PWARX), c'est-à-dire ARX par morceaux. Les régions $\mathscr{X}_i$ sont des polyhèdres de l'espace des régresseurs $\varphi(t)$ c'est-à-dire, des domaines délimités par un système d'inéquations linéaires du type $H_i\varphi \preceq d_i$, $\preceq$ désignant une inégalité élément par élément et $H_i \in \mathbb{R}^{m_i \times d}$ et $d_i \in \mathbb{R}^{m_i}$. Nous désignons par état discret λ, l'application qui associe à un instant $t \in \mathbb{Z}$, l'indice $i \in \{1, \ldots, s\}$ du sous-modèle actif à cet instant : $\lambda_t = i \iff \varphi(t) \in \mathscr{X}_i$. Dans le présent manuscrit, cet indice pourrait être encore appelé classe ou mode. Mais les expressions du type régime opératoire, sous-modèle ou encore modèle local désigneront plutôt le comportement local (rapporté par exemple à l'une des régions $\mathscr{X}_i$ de l'espace de régression) du système.

A titre illustratif nous représentons sur la figure 2.2, l'exemple d'un modèle statique (d'ordre zéro) hybride. La fonction entrée-sortie, définie sur un domaine $\mathscr{X} = [-3,\ 3]$ est affine par morceaux :

$$f(\varphi) = \begin{cases} \begin{bmatrix} 0.7 & 0.2 \end{bmatrix} \begin{bmatrix} \varphi \\ 1 \end{bmatrix} \text{ si } \varphi \in \mathscr{X}_1 = [-3,\ 0] \\ \begin{bmatrix} -0.4 & 2 \end{bmatrix} \begin{bmatrix} \varphi \\ 1 \end{bmatrix} \text{ si } \varphi \in \mathscr{X}_2 = \,]\,0,\ 3\,] \end{cases}$$

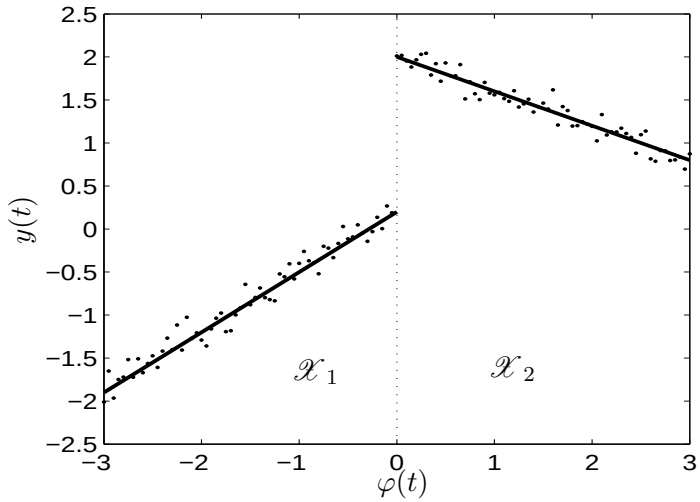


FIGURE 2.2 – Exemple de système PWA

Le problème d'identification des systèmes PWARX peut génériquement se formuler de la façon suivante :

Problème 2.1. *Etant donné les mesures entrées-sorties* $\{u(t), y(t)\}_{t=1}^{N}$ *générées par un système affine par morceaux du type (2.1), estimer à la fois le nombre* s *de sous-modèles (ou d'états discrets), les ordres* n_a *et* n_b*, la partition* $\{\mathscr{X}_i\}_{i=1}^{s}$ *de l'espace des régresseurs* $\varphi(t)$ *et les vecteurs de paramètres* $\{\theta_i\}_{i=1}^{s}$ *associés à chaque région.*

Ce problème est particulièrement délicat parce qu'il vise la double identification des régions et des paramètres du sous-modèle actif sur chaque région, la segmentation des données en régions disjointes étant proprement inséparable de la phase d'estimation des vecteurs $\{\theta_i\}_{i=1}^{s}$. En effet la connaissance des paramètres θ_i décrivant chaque sous-modèle du système permettrait de partitioner immédiatement (estimation de l'état discret) l'ensemble des données. De même, si les séquences de données disponibles étaient séparées par sous-modèles générateurs (ou par sous-région), il ne resterait plus qu'à estimer les paramètres de chacun d'eux en recourant aux méthodes classiques de régression. Dans [58], la difficulté de ce problème est comparée à celle du très célèbre problème de *la poule et de l'oeuf.*

2.2.1 Méthodes basées sur la classification

La résolution du problème 2.1 comporte une étape de classification, notamment dans la phase de partionnement des données de régression. En se focalisant sur cette remarque, Ferrari-Trecate *et al.* proposent dans [46] une solution partielle [1] au problème 2.1 qui combine des méthodes de régression linéaire, de classification et de reconnaissance des formes. Cette méthode d'identification de PWARX procède de la façon suivante :

1. Le nombre de modèles locaux s et les ordres n_a et n_b sont supposés connus *a priori.*

1. Les ordres et le nombre de sous-modèles ne sont pas estimés mais supposés disponibles *a priori.*

2. Pour tout instant t, on collecte dans un ensemble $\mathscr{C}_t = \left\{k_t^1, \cdots, k_t^{n_v}\right\}$ dans l'ordre croissant, l'indice t de $\varphi(t)$ et ceux de ses $n_v - 1$ plus proches voisins au sens de la distance euclidienne. Soit

$$\Phi_t = \begin{bmatrix} \bar{\varphi}(k_t^1) & \cdots & \bar{\varphi}(k_t^{n_v}) \end{bmatrix} \text{ et } \quad \mathcal{Y}_t = \begin{bmatrix} y(k_t^1) & \cdots & y(k_t^{n_v}) \end{bmatrix}^\top. \tag{2.5}$$

Les ensembles $\mathscr{C}_t$ sont dits ensembles de données locales. Cette idée repose sur l'observation intuitive que des données d'un même ensemble local ainsi défini sont vraisemblablement issues d'une même région de la partition $\{\mathscr{X}_i\}_{i=1}^s$. Mais il ne saurait vraiment en être ainsi en réalité. Parmi les ensembles locaux constitués, certains peuvent renfermer seulement des données issues d'un seul et même sous-modèle, (ils sont dits ensembles locaux purs), d'autres par contre seront mixtes en ce sens qu'ils collectionneront des données issues de plusieurs sous-modèles différents (ils sont dits ensembles locaux mixtes). Le nombre $n_v = \text{card}(\mathscr{C}_t)$ est un paramètre réglable qui va influer d'une façon décisive sur les performances de l'algorithme.

3. On estime ensuite sur chaque $\mathscr{C}_t$, par l'estimateur des moindres carrés, un vecteur de paramètres θ_t^{MC}.

$$\theta_t^{\text{MC}} = \left(\Phi_t^\top \Phi_t\right)^{-1} \Phi_t^\top \mathcal{Y}_t. \tag{2.6}$$

Bien évidemment, il est supposé dans la formule (2.6) que $\text{rang}(\Phi_t) = d + 1$, c'est-à-dire que $\Phi_t^\top \Phi_t$ est inversible. La covariance empirique [74] des estimées θ_t^{MC} est alors donnée par la formule

$$V_t = \frac{\text{SSR}_t}{n_v - (d+1)} \left(\Phi_t^\top \Phi_t\right)^{-1}, \tag{2.7}$$

$$\text{où} \quad \text{SSR}_t = \mathcal{Y}_t^\top \left(I - \Phi_t \left(\Phi_t^\top \Phi_t\right)^{-1} \Phi_t^\top\right) \mathcal{Y}_t \tag{2.8}$$

est la somme des carrés des résidus du problème de régression.

On calcule aussi la moyenne m_t des régresseurs contenus dans $\mathscr{C}_t$ ainsi que leur variance Q_t :

$$m_t = \frac{1}{n_v} \sum_{k \in \mathscr{C}_t} \varphi(k), \tag{2.9}$$

$$Q_t = \sum_{k \in \mathscr{C}_t} (\varphi(k) - m_t)(\varphi(k) - m_t)^\top. \tag{2.10}$$

Désormais, pour représenter le couple $(\varphi(t), y(t))$ dans un objectif de classification, on considère le vecteur de caractéristiques $\xi_t = \begin{bmatrix} \left(\theta_t^{\text{MC}}\right)^\top & m_t^\top \end{bmatrix}^\top$. Ce vecteur peut être considéré comme la moyenne d'un nuage de caractéristiques distribuées selon une distribution gaussienne de variance

$$R_t = \begin{bmatrix} V_t & 0 \\ 0 & Q_t \end{bmatrix}.$$

La matrice R_t ou du moins son inverse mesure la dispersion autour de la moyenne ξ_t, des données

contenues dans le domaine local $\mathscr{C}_t$. Si l'ensemble $\mathscr{C}_t$ est pur, la covariance V_t ne reflétera que l'effet du bruit et sera donc très probablement plus petite que dans le cas où les données sont mixtes dans $\mathscr{C}_t$. Dans le premier cas, les données sont intégralement générées par le même sous-modèle qui est effectivement linéaire alors que dans le deuxième cas elles sont issues de un ou plusieurs sous-modèles donc d'un modèle non linéaire (l'estimation de θ_t^{MC} est donc grossière dans ce cas). De plus, des régresseurs proches et tassés sont plus susceptibles d'appartenir au même modèle local que des régresseurs distants et éparpillés (information portée par Q_t). Ces arguments confortent bien le choix du vecteur de caractéristiques ξ_t.

4. Les vecteurs $\{\xi_t\}_{t=1}^N$ sont ensuite classifiés en s classes disjointes $\mathscr{D}_j$ au moyen d'une technique « sous-optimale » de classification dérivée de l'algorithme K-means [40] et pouvant autoriser une attraction de minima locaux selon son initialisation. Chaque classe $\mathscr{D}_j$ consiste en un groupement des indices temporels des vecteurs du type ξ_t associé à leur moyenne ω_j encore appelé centre de la classe. Le partitionnement en classes est réalisé par la minimisation du critère

$$J\left(\{\mathscr{D}_i\}_{i=1}^s, \{\omega_i\}_{i=1}^s\right) = \sum_{i=1}^{s} \sum_{j \in \mathscr{D}_i} \|\xi_j - \omega_i\|^2_{R_j^{-1}} \tag{2.11}$$

qui est en fait un problème de programmation non convexe dont on ne sait isoler en réalité qu'une solution sous-optimale. La norme employée dans (2.11) est définie par $\|x\|_{R_j^{-1}} = \sqrt{x^\top R_j^{-1} x}$.

5. Etant donné la bijectivité de l'application qui, à une paire de données $(\varphi(t), y(t))$ associe un vecteur de caractéristiques ξ_t, la classification précédente équivaut à une segmentation de l'espace des données originales en groupes $\{\mathcal{F}_j\}_{j=1}^s$. Ainsi, à partir des résultats de la classification, on peut faire de nouveau recours aux moindres carrés pour finalement calculer les vecteurs de paramètres θ_j, $j = 1, \ldots, s$.

Il reste maintenant à déterminer les paramètres H_i et d_i qui fixent les frontières des régions. Il s'agit de trouver pour tous i et j, $i \neq j$, l'hyperplan qui sépare les points contenus dans $\mathscr{X}_i$ de ceux dans $\mathscr{X}_j$ (quand les deux sont adjacents). Les méthodes d'apprentissage statistique comme les Support Vector Machines (SVM) [116] offrent un potentiel intéressant pour accomplir cette tâche [46].

Les potentialités d'approximation de la méthode de Ferrari-Trecate *et al.* sont illustrées par la figure 2.3. Le domaine de définition de la fonction non-linéaire (ici une fonction sinus) est décomposé en une partition d'intervalles représentés sur la figure par une bande de couleurs. Sur chacun de ces intervalles un modèle affine est utilisé pour approximer la non-linéarité. En dernière remarque, il est à noter que la méthode de Ferrari-Trecate *et al.* ne garantit pas l'obtention de l'optimum global du problème d'optimisation du critère mixte de classification-identification. Ses performances globales sont liées au choix du paramètre de réglage n_v et à la procédure d'initialisation de l'algorithme de classification. De plus, le nombre s de modèles locaux et les ordres n_a et n_b doivent être connus *a priori* avec plus ou moins d'exactitude. Le lecteur intéressé pourra se référer à [65] pour quelques commentaires supplémentaires sur cette méthode.

Il existe d'autres méthodes d'identification de systèmes linéaires par morceaux qui s'appuient sur le concept de classification de données. Par exemple dans [87], Nakada *et al.* utilisent une méthode de classification statistique pour séparer complètement les données selon les régions avant d'employer des méthodes de régression linéaire pour obtenir les paramètres correspondants. Ils suggèrent aussi une

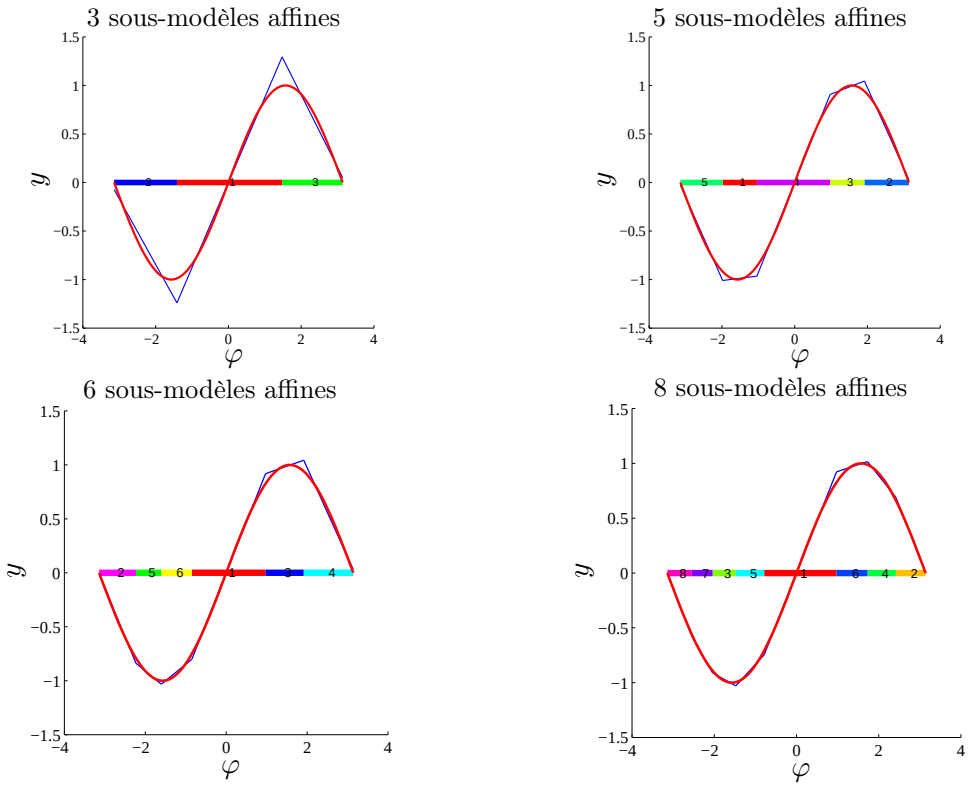


FIGURE 2.3 – Approximation d'une fonction sinusoïdale par des modèles locaux affines par morceaux. Ces graphes ont été tracés avec le Toolbox Matlab développé par Ferrari-Trecate [45] pour l'identification de systèmes hybrides. On voit sur cet exemple que plus il y a de sous-modèles, plus l'erreur d'approximation est petite. Cependant, un nombre important de sous-modèles pourrait être extrêmement ardu à estimer.

technique d'extraction du nombre de sous-modèles lorsque les ordres sont connus.

Une autre catégorie assez variée de méthodes alterne entre l'assignation des données aux régions et l'estimation simultanée des paramètres, en adoptant une technique d'apprentissage des poids sur un modèle du type Takagi-Sugeno [99], en combinant l'algorithme des moindres carrés récursifs avec une règle de décision a *posteriori* [8], en résolvant un problème de partition MIN PFS (Minimum Partition into Feasible Subsystems) [18], ou en recourant à une inférence bayésienne [67]. Ces deux dernières méthodes feront, pour des raisons de popularité, l'objet d'une présentation plus détaillée dans les deux sous-sections suivantes.

2.2.2 Approche de l'erreur bornée

Dans cette approche, le problème 2.1 est traité comme un problème de partitionnement en un nombre minimum de sous-systèmes réalisables (MIN PFS) [3], [17]. Le principe de la méthode consiste à imposer à l'erreur $e(t)$ de l'équation (2.1) d'être bornée par un nombre $\delta > 0$, soit

$$|y(t) - f(\varphi(t))| \leq \delta, \ \forall t = \bar{n}+1, \ldots, N, \tag{2.12}$$

avec f définie comme en (2.3) et $\bar{n} = \max(n_a, n_b)$. Sous cette contrainte, le problème 2.1 se relit comme suit. Etant donné l'ensemble de données $\{\varphi(t), y(t)\}_{t=\bar{n}+1}^{N}$ et un nombre $\delta > 0$, estimer le plus petit entier s, un ensemble de paramètres $\{\theta_i\}_{i=1}^{s}$, une partition polyédrale $\{\mathscr{X}_i\}_{i=1}^{s}$ de l'espace de régression $\mathscr{X} = \cup_{i=1}^{s} \mathscr{X}_i \subset \mathbb{R}^d$ tel que le modèle (2.1) satisfasse la condition (2.12).
Le paramètre réglable δ permet de fixer la complexité en même temps que la précision du modèle recherché. Pour un δ fixé, la segmentation des $N - \bar{n}$ inégalités complémentaires

$$\left| y(t) - \theta_{\lambda_t}^{\top} \bar{\varphi}(t) \right| \leq \delta,\ t = \bar{n}+1, \ldots, N, \tag{2.13}$$

en un nombre minimum s de sous-modèles réalisables est un problème numériquement complexe dit « NP-difficile » [1]. On peut noter que le paramètre δ joue ici un rôle semblable à celui du paramètre n_v dans la méthode de la section 2.2.1. Le choix de ce paramètre de réglage est déterminant pour les performances de l'algorithme. Par exemple, plus δ est grand, moins il y aura de sous-modèles dans la partition obtenue. Par contre un δ petit conduit, sous la contrainte (2.13), à un nombre important de sous-modèles.

Nous donnons maintenant un aperçu de la méthode de l'erreur bornée qui a été développée par Bemporad *et al.* [17]. Cet algorithme (résumé sur la figure 2.4) comporte trois étapes.
La première étape comprend une classification initiale des données entrée-sortie concomitamment à l'estimation du nombre de sous-modèles ainsi que des paramètres correspondant à chacun d'eux. Cette étape est fondée sur l'algorithme *glouton* de Amaldi et Mattaveli [3] pour la résolution du problème MIN PFS. Le principe de ce dernier algorithme consiste à subdiviser le problème de la partition minimum mentionné ci-dessus en une série de sous-problèmes dits MAX FS, chacun d'eux faisant référence à la détermination d'un vecteur θ satisfaisant au nombre maximum d'inégalités du type $\left| y(t) - \theta^{\top} \bar{\varphi}(t) \right| \leq \delta$. En partant de l'ensemble des données $\mathscr{D} = \{(\varphi(t), y(t)), t = \bar{n}+1, \ldots, N\}$, on commence par trouver un premier θ noté θ_1 tel que le cardinal de $\mathscr{D}_1 = \left\{(\varphi(t), y(t)) : \left| y(t) - \theta_1^{\top} \bar{\varphi}(t) \right| \leq \delta \right\}$ soit maximum. Ensuite, en excluant dans $\mathscr{D}$ les éléments de $\mathscr{D}_1$, on répète la même procédure sur le reste des données et ainsi de suite jusqu'à ce qu'on obtienne, pour un certain s^*, un ensemble $\mathscr{D} \setminus \{\mathscr{D}_1 \cup \cdots \cup \mathscr{D}_{s^*}\}$ réalisable. Alors, le nombre de sous-modèles est automatiquement obtenu comme $s = s^* + 1$. Pour plus de détails sur l'extraction des vecteurs θ_j, le lecteur intéressé pourra se reporter à [18]. On peut noter que cette approche est essentiellement une heuristique qui ne garantit pas l'obtention de la meilleure partition possible des données.
Ainsi, il est proposé dans la deuxième étape, une procédure corrective de la classification initialement obtenue à la première étape et une amélioration des valeurs des paramètres estimés. En partant de ces valeurs initiales, on applique une procédure itérative alternant entre la classification des données $\varphi(t)$ et la mise à jour des vecteurs θ_j.
La dernière étape concerne la recherche d'une partition polyhédrale de l'espace de régression. Cela est réalisé en ayant recours à des méthodes de séparation linéaire de données en deux ou plusieurs classes [20], [21], [105].

1. Non-deterministic Polynomial-time-difficile.

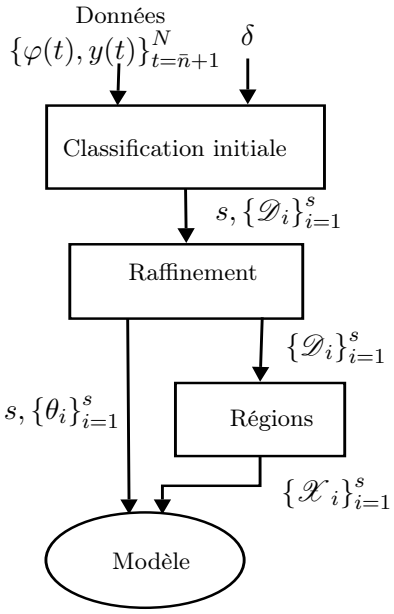


FIGURE 2.4 – Schéma de l'algorithme de l'erreur bornée. Ici, l'ensemble $\mathscr{D}_i$ est défini par $\mathscr{D}_i = \left\{(\varphi(t), y(t)) : \left|y(t) - \theta_i^\top \bar{\varphi}(t)\right| \leq \delta\right\}$ tandis que $\mathscr{X}_i$ fait référence à un polyèdre de la partition (2.4).

2.2.3 Approche bayésienne

Juloski *et al.* proposent dans [67], une approche fondée sur l'inférence bayésienne pour l'identification des systèmes PWARX. En supposant que le nombre s de modes et les ordres n_a et n_b sont donnés *a priori*, le principe de cette méthode peut être résumé de la façon suivante. Chaque vecteur de paramètres θ_i est considéré comme une variable aléatoire à laquelle on substitue, dans la procédure d'estimation, la fonction de densité de probabilité (f.d.p.) $p_{\theta_i}(\cdot)$ qui la caractérise. L'objectif de cette démarche est de se placer dans un espace qui permette l'application des règles de mise à jour bayésienne. Notons $p_{\theta_i}(\theta; t)$ la f.d.p. continue de θ_i à l'instant (ou l'étape d'itération) t, définie sur un espace dense $\Theta_i \subset \mathbb{R}^{d+1}$, avec $d = n_a + n_u(n_b + 1)$. La théorie du filtrage particulaire [5] permet alors d'approcher cette f.d.p. de la façon suivante.

$$\hat{p}_{\theta_i}(\theta; t) = \sum_{l=1}^{M} w_{i,l}(t)\delta(\theta - \theta_{i,l}(t)), \tag{2.14}$$

où δ correspond à la distribution de Dirac, $\{w_{i,l}(t)\}_{i=1,l=1}^{i=s,l=M} > 0$ représentent des poids associés aux $\theta_{i,l}(t)$ et vérifiant $\sum_{l=1}^{M} w_{i,l}(t) = 1$. L'échantillon fini $\{\theta_{i,l}(t)\}_{l=1}^{M}$ est supposé être généré sur l'espace Θ_i selon la distribution donnée par $p_{\theta_i}(\theta; t)$. De toute évidence, la fiabilité d'une telle approximation est étroitement liée au nombre M de particules. Soit $p((\varphi(t), y(t))|\hat{\lambda}_t = i)$ la probabilité que le mode i ait généré la donnée $(\varphi(t), y(t))$. L'état discret $\hat{\lambda}_t$ correspond alors au mode qui maximise cette probabilité, c'est-à-dire,

$$\hat{\lambda}_t = \arg\max_i p((\varphi(t), y(t))|\hat{\lambda}_t = i), \tag{2.15}$$

avec

$$p((\varphi(t), y(t))|\hat{\lambda}_t = i) = \int_{\Theta_i} p((\varphi(t), y(t))|\theta) p_{\theta_i}(\theta; t-1) d\theta \tag{2.16}$$

et

$$p\left((\varphi(t), y(t))\,|\theta\right) = p_e\left(y(t) - \theta^\top \bar{\varphi}(t)\right), \tag{2.17}$$

où p_e est la f.d.p. *a priori* du bruit additif $e(t)$ du modèle (2.1), supposé être une séquence indépendante et identiquement distribuée. En s'aidant de la formule (2.14), une approximation (par discrétisation) du critère de décision (2.16) peut s'écrire

$$p((\varphi(t), y(t))|\hat{\lambda}_t = i) \simeq \sum_{l=1}^{M} w_{i,l}(t-1) p_e\left(y(t) - \bar{\varphi}(t)^\top \theta_{i,l}(t-1)\right). \tag{2.18}$$

Ainsi, la connaissance des couples particules-poids $\{\theta_{i,l}(t-1),\ w_{i,l}(t-1)\}$ *a priori* suffit à la détermination de l'état discret $\hat{\lambda}_t$ selon la règle (2.15). Une fois $\hat{\lambda}_t$ connu, il faut procéder à la mise à jour de la f.d.p. $p_{\theta_{\hat{\lambda}_t}}(\cdot; t)$, les autres f.d.p. $p_j(\cdot; t)$, $j = 1, \ldots, s$, $j \neq \hat{\lambda}_t$, restant inchangées à cette étape. Cela peut être réalisé en utilisant la règle d'inférence bayesienne c'est-à-dire, en faisant

$$p_{\theta_{\hat{\lambda}_t}}(\theta; t) = \frac{p_e\left(y(t) - \theta^\top \bar{\varphi}(t)\right) p_{\theta_{\hat{\lambda}_t}}(\theta; t-1)}{\int_{\Theta_i} p_e\left(y(t) - \theta^\top \bar{\varphi}(t)\right) p_{\theta_{\hat{\lambda}_t}}(\theta; t-1)}. \tag{2.19}$$

Au vu de l'approximation particulaire (2.14), calculer $p_{\theta_{\hat{\lambda}_t}}(\theta; t)$ revient à calculer $\left\{\theta_{\hat{\lambda}_t,l}(t),\ w_{\hat{\lambda}_t,l}(t)\right\}$. Pour ce faire, une variante de l'algorithme de filtrage particulaire SIR (Sample Importance Resampling) [5] est utilisée.

L'ensemble de la procédure d'identification est résumé dans l'algorithme 2.1. Cet algorithme affecte séquentiellement les données aux modes tout en estimant les paramètres de chaque sous-modèle. [1] Une fois que l'algorithme a parcouru toutes les données disponibles, on s'attend à avoir des f.d.p., qui par suite des mises à jour succesives opérées séquentiellement pour chaque donnée, s'apparentent enfin à d'assez bonnes approximations de la réalité. Ainsi, et toujours selon le principe qu'une donnée est attribuée au mode ayant la plus forte probabilité (de l'avoir générée), une classification finale est réalisée selon (2.15) grâce aux f.d.p. $p_{\theta_i}(\cdot; N)$. Après cela, une méthode de programmation linéaire [21] est utilisée pour estimer les hyperplans frontières des régions $\mathscr{X}_i = \{\varphi(t) : \lambda_t = i\}$.

Une caractéristique assez évidente de la démarche bayésienne présentée est la complexification qui accompagne l'idée de substituer à chaque variable θ_i, une fonction $p_{\theta_i}(\cdot)$ que l'on sait continue et non-linéaire. L'approximation de celle-ci par une fonction à support fini nécessite, pour être efficace, que M soit très grand et qu'un domaine probable de variation de θ_i soit connu. Aussi, on estime les f.d.p. au prix d'un effort de calcul accru mais sans pour autant atténuer sensiblement le caractère « sous-optimal » de l'algorithme. Par ailleurs, l'initialisation d'un nombre aussi élevé de variables $\{\theta_{i,l}\}$, $i = 1, \ldots, s$, $l = 1, \ldots, M$ n'est pas sans incidence sur les performances qu'on pourrait obtenir [8].

1. A l'origine, cet algorithme fonctionne en hors ligne : il y a au moins une itération sur l'ensemble des données.

Algorithme 2.1 Classification bayésienne

1. Initialisation :
   - Fournir les f.d.p. *a priori* $p_{\theta_i}(\cdot;0)$ pour $i=1,\ldots,s$,
   - Fournir $p_e(\cdot)$
2. **Pour** tout $t=1,\ldots,N$
   - Affecter $(\varphi(t),y(t))$ au mode $\hat{\lambda}_t$ défini par (2.15)
   - Evaluer la f.d.p. *a posteriori* $p_{\theta_{\hat{\lambda}_t}}(\cdot;t)$ du paramètre $\theta_{\hat{\lambda}_t}$ comme suit :
     - Diversification des particules :
     $$\theta_{\hat{\lambda}_t,l}(t)=\theta_{\hat{\lambda}_t,l}(t-1)+\varepsilon_l,$$
     avec $\varepsilon_l\sim N(0,\Sigma_\varepsilon)$, $l=1,\ldots,M$.
     - Calcul des poids :
     $$w_{\hat{\lambda}_t,l}(t)=p\left((\varphi(t),y(t))|\theta_{\hat{\lambda}_t,l}(t)\right)=p_e\left(y(t)-\bar{\varphi}(t)^\top\theta_{\hat{\lambda}_t,l}(t)\right),$$
     $l=1,\ldots,M$.
     - Normalisation des poids :
     $$w_{\hat{\lambda}_t,l}(t):=\frac{w_{\hat{\lambda}_t,l}(t)}{\sum_{l=1}^M w_{\hat{\lambda}_t,l}(t)},\ l=1,\ldots,M.$$
     - Rééchantillonnage : De la distribution (2.14) calculée avec $\left\{\theta_{\hat{\lambda}_t,l}(t),w_{\hat{\lambda}_t,l}(t)\right\}$, on génère ensuite indépendamment M points pour former de nouvelles particules uniformément pondérées $\left(\theta_{\hat{\lambda}_t,l}(t),\ 1/M\right)$.
   - Pour tout $j\neq\hat{\lambda}_t$, $p_{\theta_j}(\cdot;t)=p_{\theta_j}(\cdot;t-1)$.
3. Aller à l'étape 2 tant que $t<N$.

2.3 Identification de modèles HHARX (Hinging Hyperplanes ARX)

En dehors des modèles PWARX, d'autres structures de modèles sont utilisées pour la représentation de systèmes dynamiques hybrides. C'est le cas par exemple des modèles du type *jonction d'hyperplans*[1] HHARX qui forment une sous-classe spéciale de modèles PWARX pour laquelle la fonction affine par morceaux (2.3) est continue.

2.3.1 Les hyperplans de Breiman

Une fonction *gond*[2] (Hinge Function en anglais) $y=h(\varphi)$ est constituée de deux demi-hyperplans joints en forme de livre ouvert comme illustré sur la figure 2.5 [25]. L'intersection des deux hyperplans est désignée par le vocable *gond*. Si $y=\bar{\varphi}^\top\theta^+$ et $y=\bar{\varphi}^\top\theta^-$, avec $\bar{\varphi}=\begin{bmatrix}\varphi^\top & 1\end{bmatrix}^\top\in\mathbb{R}^{d+1}$, désignent les équations des deux hyperplans, alors l'expression des fonctions HF est soit $h(\varphi)=\max(\bar{\varphi}^\top\theta^+,\bar{\varphi}^\top\theta^-)$,

1. Ici, le terme ARX fait référence à la manière dont le régresseur $\varphi(t)$ est construit, c'est-à-dire, comme une concaténation d'entrées de sorties passées sur un certain horizon fini. 2. par analogie au gond d'une porte.

soit $h(\varphi) = \min(\bar{\varphi}^\top\theta^+, \bar{\varphi}^\top\theta^-)$ avec θ^- et θ^+, les vecteurs de paramètres à déterminer et φ, le vecteur de régression. L'intersection des deux hyperplans (*gond*) est alors donnée par $\bar{\varphi}^\top\Delta = 0$ avec $\Delta = \theta^+ - \theta^-$. Les modèles *Hinging Hyperplanes* (HH) ont été introduits par Breiman [25] comme outils de modélisation non-linéaire. Comme la majorité des techniques de régression non linéaire, l'approche HH consiste à développer la fonction à approximer sur une base de fonctions élémentaires, en l'occurrence des fonctions du type HF. Breiman démontre que toute fonction f définie et continue sur un support compact peut être approchée avec une précision arbitraire sur une base de HFs par

$$\hat{f}(\varphi) = \sum_{i=1}^{s} h_i(\varphi), \tag{2.20}$$

avec s, le nombre de fonctions de la base, $h_i(\varphi) = \min(\bar{\varphi}^\top\theta_i^+, \bar{\varphi}^\top\theta_i^-)$ ou $h_i(\varphi) = \max(\bar{\varphi}^\top\theta_i^+, \bar{\varphi}^\top\theta_i^-)$. Lorsque φ est un régresseur de la forme (2.2), le modèle (2.20) est dit modèle HHARX. Le modèle (2.20) garantit l'existence d'une borne supérieure de l'erreur d'approximation qui est inversement proportionnelle au nombre s de noeuds du réseau HH :

$$\left\| f(\varphi) - \sum_{i=1}^{s} h_i(\varphi) \right\|^2 \leq \frac{(2R)^4 c^2}{s}, \tag{2.21}$$

R étant le rayon de la sphère contenant le compact domaine de définition, c une constante définie par $c = \int \|\omega\|^2 |f(\omega)| \, d\omega < \infty$. On peut noter au passage qu'en principe, la borne supérieure de l'erreur d'approximation décroît avec le nombre de noeuds. En revanche, un nombre important de fonctions h_i peut s'avérer très fastidieux à estimer en pratique.

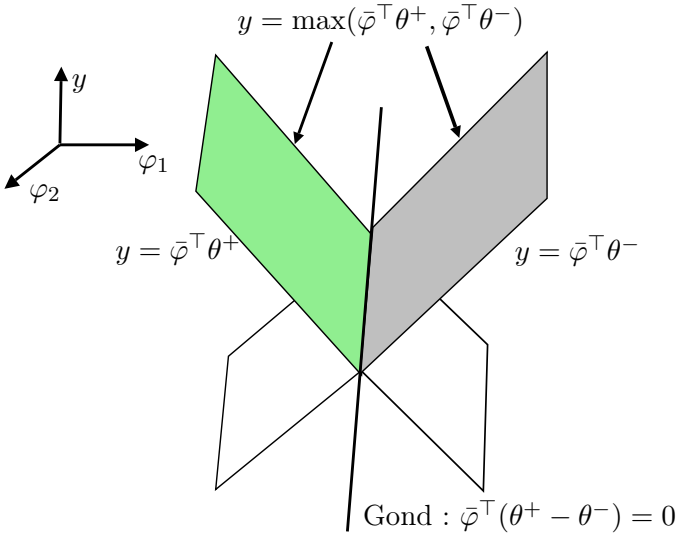


FIGURE 2.5 – Représentation d'une fonction gond (HF).

Remarque 2.1 (Modèles canoniques de Chua)**.** *Les formes canoniques des modèles affines par morceaux ont été introduites par Chua et Kang dans [32, 71] pour l'approximation d'une fonction non-linéaire f*

définie sur un compact de l'espace euclidien $\mathbb{R}^n$. Cette structure canonique est donnée par

$$f(\varphi) = \theta_0^\top \bar{\varphi} + \sum_{j=1}^{s} \mu_j \left|\theta_j^\top \bar{\varphi}\right|, \tag{2.22}$$

avec $\bar{\varphi} = \begin{bmatrix}\varphi^\top & 1\end{bmatrix} \in \mathbb{R}^{d+1}$ et μ_j, $j = 1, \ldots, s$, des poids scalaires. Notons que cette forme de représentation est possible pour toute fonction prenant des valeurs scalaires mais son existence n'est pas garantie pour toute fonction à valeurs dans $\mathbb{R}^m$, avec $m \geq 2$ [31, 50]. En fait, il existe une équivalence entre les modèles HH de Breiman (2.20) et les représentations canoniques de Chua (2.22) [100].

2.3.2 Algorithme HFA (Hinge Finding Algorithm)

Pour déterminer le *gond* Δ à partir d'une collection de données $\{\varphi(t), y(t)\}_{t=1}^N$, Breiman [25] propose l'algorithme suivant, appelé algorithme HFA.

- Initialisation : une valeur arbitraire est attribuée à Δ, soit $\Delta^{(0)}$. Les échantillons $\{\bar{\varphi}(t), y(t)\}_{t=1}^N$ d'observations sont alors partitionnés en deux ensembles $S_+ = \left\{\bar{\varphi} : \bar{\varphi}^\top \Delta^{(0)} > 0\right\}$ et $S_- = \left\{\bar{\varphi} : \bar{\varphi}^\top \Delta^{(0)} \leq 0\right\}$. On calcule alors θ^+ et θ^- respectivement par

$$\theta^+ = \Big(\sum_{\bar{\varphi}(t) \in S_+} \bar{\varphi}(t)\bar{\varphi}(t)^\top \Big)^{-1} \sum_{\bar{\varphi}(t) \in S_+} \bar{\varphi}(t) y(t), \tag{2.23}$$

$$\theta^- = \Big(\sum_{\bar{\varphi}(t) \in S_-} \bar{\varphi}(t)\bar{\varphi}(t)^\top \Big)^{-1} \sum_{\bar{\varphi}(t) \in S_-} \bar{\varphi}(t) y(t). \tag{2.24}$$

- On répète l'étape précédente avec $\Delta^{(1)} = \theta^+ - \theta^-$ jusqu'à la convergence de l'algorithme vers un optimum local Δ^*.

Une fois Δ^* obtenu, on peut calculer les θ^+ et θ^- correspondants en utilisant les formules (2.23)-(2.24) et ainsi obtenir le premier HF h_1. Supposons maintenant qu'on veuille estimer la J-ème HF h_J, connaissant $h_1, \cdots, h_{J-1}$. Il suffit alors d'appliquer l'algorithme HFA aux échantillons de données $\{\bar{\varphi}(t), \tilde{y}_J(t)\}_{t=1}^N$ avec $\tilde{y}_J(t) = y(t) - \sum_{j=1}^{J-1} h_j(\varphi(t))$, $\tilde{y}_1(t) = y(t)$.

Notons que l'algorithme HFA pour la détermination de h_J tente de minimiser en fait le critère

$$V_N(\theta) = \frac{1}{2} \sum_{t=1}^{N} \big(\tilde{y}_J(t) - h_J(\varphi(t), \theta)\big)^2.$$

Donc, $\hat{\theta} = \arg\min_\theta V_N(\theta)$ avec $\theta = \begin{bmatrix}(\theta^+)^\top & (\theta^-)^\top\end{bmatrix}^\top \in \mathbb{R}^{2(d+1)}$. La routine HFA pour l'estimation du *gond* présente l'avantage très attractif d'être simple mais une analyse poussée montre qu'elle souffre de problèmes de convergence. En effet sa convergence vers un optimum c'est-à-dire le vrai *gond* n'est pas garantie puisque celle-ci est assez tributaire du point d'initialisation. Pucar et Sjöberg [97] ont montré que cet algorithme correspond en réalité à un cas particulier de l'algorithme d'optimisation de Newton.

En appliquant cette dernière méthode à $V_N(\theta)$, on obtient la loi itérative suivante :

$$\theta^{(i+1)} = \theta^{(i)} + \delta\theta^{(i)},$$

avec $\delta\theta^{(i)} = -\left(\nabla^2 V_N(\theta)\right)^{-1}\nabla V_N(\theta) = \theta_{Br}^{(i+1)} - \theta^{(i)}$, où $\theta_{Br}^{(i+1)}$ est le vecteur de paramètres qui aurait été obtenu par l'algorithme HFA de Breiman. Il est suggéré par conséquent dans [97] de lui apporter des modifications correctives (version amortie de l'algorithme de Newton) à l'algorithme de Breiman. Cette amélioration consiste dans l'introduction d'un facteur μ selon

$$\theta^{(i+1)} = \theta^{(i)} + \mu\delta\theta^{(i)},$$

offrant ainsi un degré de liberté supplémentaire dans la recherche de l'optimum. La paramètre μ doit être choisi de manière à garantir $V_N(\theta^{(i+1)}) \leq V_N(\theta^{(i)})$.

2.3.3 Approche de la programmation mixte en nombres entiers

Une caractéristique de la plupart des algorithmes présentés jusqu'ici pour l'identification de systèmes linéaires affines par morceaux est leur incapacité à fournir un optimum global au critère d'identification-classification. La méthode proposée dans [101] par Roll *et al.* s'affranchit de cette difficulté mais au prix d'un coût calculatoire important. La principale contribution des auteurs consiste dans une subtile reformulation du problème 2.1 en un problème de programmation linéaire ou quadratique mixte en nombres entiers (MILP/MIQP) [104], [33]. Cela est motivé par le fait qu'il existe des outils assez efficaces pour traiter de façon optimale le problème de MILP/MIQP. L'algorithme est développé pour des modèles ARX du type hyperplans de Breiman (HHARX) [25] :

$$y(t) = g(\varphi(t), \Theta) + e(t), \tag{2.25}$$

où

$$g(\varphi(t), \Theta) = \theta_0^\top \bar{\varphi}(t) + \sum_{i=1}^{s} s_i \max(\theta_i^\top \bar{\varphi}(t), 0)$$
$$\Theta = \begin{bmatrix} \theta_0 & \dots & \theta_s \end{bmatrix} \in \mathbb{R}^{(d+1)\times s},$$

avec $s_i = +1$ ou $s_i = -1$ selon le signe de la fonction max et $\bar{\varphi}(t)$ est, comme précédemment défini par $\bar{\varphi}(t) = \begin{bmatrix} \varphi(t)^\top & 1 \end{bmatrix}^\top \in \mathbb{R}^{d+1}$. Le lecteur intéressé pourra se référer à [100] ou à [32] pour plus de détails sur ce modèle. En remarquant que pour tout z, $-z + \max\{z, 0\} = \max\{-z, 0\}$, on peut noter que le modèle de l'équation (2.25) n'est pas globalement identifiable puisque certains termes peuvent se compenser. Cette compensation potentielle de termes entre eux pourrait autoriser une multiplicité de solutions $\{\theta_j\}_{j=1}^{s}$. Pour lever cette difficulté, il est possible d'introduire des contraintes supplémentaires

sur les paramètres θ_j de façon à particulariser la solution attendue. Soient par exemple les contraintes

$$w^\top \theta_1 \geq \cdots \geq w^\top \theta_s \geq 0, \tag{2.26}$$

avec w un vecteur non nul quelconque dans $\mathbb{R}^{d+1}$. Partant du modèle (2.25), le problème se rapporte désormais à l'estimation des fonctions $\max(\theta_i^\top \bar{\varphi}(t), 0)$, $i = 0, \ldots, s$.

En considérant un critère basé sur la norme 1, l'optimum Θ^* est donné par

$$\Theta^* = \arg\min V(\Theta) = \sum_{t=\bar{n}+1}^{N} |y(t) - g(\varphi(t), \Theta)| \tag{2.27}$$

sujet aux contraintes

$$\begin{cases} \theta^{j-} \leq \theta_j \leq \theta^{j+}, & j = 1, \ldots, s, \\ w^\top \theta_j \geq 0, & j = 1, \ldots, s, \end{cases}$$

où $\bar{n} = \max(n_a, n_b)$. Les quantités w, θ^{j-} et θ^{j+} sont fixées *a priori*. Afin de convertir (2.27) en un problème de Programmation Linéaire en Nombres Entiers MILP [104], [33], on introduit les variables binaires (auxiliaires) δ_{it} et continues z_{it} définies par

$$\delta_{it} = 0 \Longleftrightarrow \theta_i^\top \bar{\varphi}(t) \leq 0, \tag{2.28}$$

$$z_{it} = \max(\theta_i^\top \bar{\varphi}(t), 0) = \theta_i^\top \bar{\varphi}(t)\delta_{it}, \tag{2.29}$$

$$i = 1, \ldots, s, \quad t = \bar{n} + 1, \ldots, N.$$

Les équations (2.28) et (2.29) sont équivalentes à

$$\begin{cases} z_{it} \geq 0, \\ z_{it} \leq M_{it}^\theta \delta_{it}, \\ \theta_i^\top \bar{\varphi}(t) \leq z_{it}, \\ (1 - \delta_{it}) m_{it}^\theta + z_{it} \leq \theta_i^\top \bar{\varphi}(t), \end{cases} \tag{2.30}$$

avec M_{it}^θ et m_{it}^θ, respectivement les bornes supérieures et inférieures de $\theta_i^\top \bar{\varphi}(t)$. La proposition suivante résume l'analogie entre le problème d'estimation du modèle (2.25) et celui de programmation mixte en nombres entiers.

Proposition 2.1 ([101])**.** *Rechercher un optimum au problème (2.27) revient à trouver l'optimum du MILP suivant :*

$$\min_{\varepsilon_t, \theta_i, z_{it}, \delta_{it}} \sum_{t=\bar{n}+1}^{N} \varepsilon_t$$

sujet à

$$\begin{cases} \varepsilon_t \geq 0, \\ \varepsilon_t \geq y(t) - \theta_0^\top \bar{\varphi}(t) - \sum_{i=1}^{s} s_i z_{it}, \\ \varepsilon_t \geq \theta_0^\top \bar{\varphi}(t) + \sum_{i=1}^{s} s_i z_{it} - y(t), \\ \delta_{it} \in \{0,1\}\,,\ 0 \leq z_{it} \leq M_{it}^{\theta}, \quad \theta^{j-} \leq \theta_j \leq \theta^{j+}, \\ (2.26)\ et\ (2.30), \end{cases}$$

où δ_{it}, z_{it}, θ_i, ε_t désignent les variables d'optimisation, $t = \bar{n}+1, \ldots, N$, $i = 1, \ldots, s$ et N, s, s_i, $y(t)$, $\varphi(t)$ fournis.

Le problème d'identification est ainsi résolu au travers de la programmation linéaire (norme 1 dans l'équation (2.27)) ou quadratique (norme 2 dans l'équation (2.27)) en nombres entiers [104], [33]. Nous terminons ce paragraphe en mentionnant que l'approche des MILP/MIQP pour l'identification des modèles PWARX, bien qu'optimale est numériquement très complexe [101]. De plus, l'application de cette méthode requiert la connaissance des ordres n_a et n_b et du nombre s de sous-modèles. Une autre difficulté de la méthode est relative au choix des bornes θ^{j-}, θ^{j+}, M_{it}^{θ} et m_{it}^{θ} qui définissent l'espace de recherche.

2.4 Identification de systèmes à commutations

2.4.1 Estimation algébro-géométrique de modèles SARX (Switched ARX)

Contrairement aux algorithmes précédents qui sont dédiés à l'identification de modèles PWARX et HHARX, la méthode proposée par R. Vidal *et al.* dans [131] s'applique à l'identification de modèles à commutations (SARX). Dans le cas des systèmes PWARX, les lois qui gouvernent la commutation du système d'un sous-modèle à un autre est une fonction du régresseur (loi de commutation définie par une partition polyhédrale de l'espace de régression) tandis que pour les SARX, le mécanisme des commutations peut être d'origine totalement arbitraire. Cela confère aux modèles commutants un caractère plus général.

Afin de présenter le principe de la méthode algébro-géométrique de R. Vidal, considérons le modèle SISO SARX suivant

$$y(t) = \sum_{i=1}^{n_a(\lambda_t)} a_{\lambda_t}^i y(t-i) + \sum_{i=0}^{n_b(\lambda_t)} b_{\lambda_t}^i u(t-i) + e(t), \tag{2.31}$$

où λ_t désigne l'état discret du système, $\lambda_t \in \{1, \ldots, s\}$, s étant le nombre d'états discrets. L'état discret est ici l'application qui associe à chaque instant t, l'indice du modèle local en activité à cet instant. Pour simplifier, nous supposerons que $n_a(1) = \cdots = n_a(s) = n_a$ et $n_b(1) = \cdots = n_b(s) = n_b$, c'est-à-dire que

tous les sous-modèles ont les mêmes ordres. Nous définissons alors

$$\bar{\varphi}(t) = \begin{bmatrix} -y(t) & y(t-1) & \dots & y(t-n_a) & u(t) & u(t-1) & \dots & u(t-n_b) \end{bmatrix}^\top \in \mathbb{R}^K, \tag{2.32}$$

$$\theta_i = \begin{bmatrix} 1 & a_i^1 & \dots & a_i^{n_a(i)} & b_i^0 & b_i^1 & \dots & b_i^{n_b(i)} \end{bmatrix}^\top \in \mathbb{R}^K, \tag{2.33}$$

avec $K = n_a + n_b + 2$. Alors, le modèle (2.31) peut se lire comme suit : pour tout $t > n_a$, il existe au moins un $i \in \{1, \dots, s\}$ tel que l'équation $\theta_i^\top \bar{\varphi}(t) = 0$ soit vérifiée. En d'autres termes, $\bar{\varphi}(t)$ appartient à une union $\mathcal{H}_1 \cup \dots \cup \mathcal{H}_s$ d'hyperplans homogènes $\mathcal{H}_i = \left\{x \in \mathbb{R}^K : \theta_i^\top x = 0\right\}$ de l'espace des régresseurs $\mathbb{R}^K$. Cet état de faits peut se traduire par la condition suivante dite *contrainte de découplage hybride* :

$$\prod_{i=1}^{s} \left(\theta_i^\top \bar{\varphi}(t)\right) = 0 \quad \forall\, t > \bar{n} = \max(n_a, n_b). \tag{2.34}$$

Soit le polynôme homogène P_s de degré s en K variables définie de la façon suivante

$$\begin{aligned} P_s(z) = \prod_{i=1}^{s} \left(\theta_i^\top z\right) &= \sum h_{s_1,\cdots,s_K} z_1^{s_1} \cdots z_K^{s_K} \\ &= h^\top \nu_s(z), \end{aligned} \tag{2.35}$$

où $\nu_s : \mathbb{R}^K \to \mathbb{R}^{M_s(K)}$, $M_s(K) = \binom{K+s-1}{s} = \dfrac{(K+s-1)!}{s!(K-1!)}$, est l'application de Veronese [52] de degré s qui associe à $z \in \mathbb{R}^K$, le vecteur de tous les monômes de degré s, $z_1^{s_1} \cdots z_K^{s_K}$, $0 \le s_j \le s$ pour tout j et $s_1 + \dots + s_K = s$, organisés dans l'ordre *lexicographique* décroissant. Ici, $h \in \mathbb{R}^{M_s(K)}$ et appelé *vecteur de paramètres hybride*, est formé de tous les coefficients des monômes de $P_s(z)$. En fait, comme vecteur coefficient du polynôme $(\theta_1^\top z) \cdots (\theta_s^\top z)$, le vecteur h est le produit tensoriel symétrique [52] des paramètres θ_i, $i = 1, \dots, s$, c'est-à-dire $h = \mathrm{Sym}\,(\theta_1 \otimes \dots \otimes \theta_s)$.

Avec le polynôme $P_s(z)$ défini en (2.35), la contrainte de découplage hybride (2.34) peut se réécrire

$$P_s\big(\bar{\varphi}(t)\big) = h^\top \nu_s\big(\bar{\varphi}(t)\big) = 0,\ t = \bar{n}+1, \dots, N, \tag{2.36}$$

En appliquant la relation (2.36) à l'ensemble des données entrées-sorties $\{\bar{\varphi}(t)\}_{t=\bar{n}+1}^N$, on obtient le système d'équations linéaires suivant

$$L_s h = 0, \tag{2.37}$$

avec

$$L_s = \begin{bmatrix} \nu_s(\bar{\varphi}_{\bar{n}+1}) & \cdots & \nu_s(\bar{\varphi}_N) \end{bmatrix}^\top \in \mathbb{R}^{(N-\bar{n})\times M_s(K)} \quad \text{et} \quad h \in \mathbb{R}^{M_s(K)}. \tag{2.38}$$

Pour déterminer le nombre s d'états discrets, on utilise le théorème suivant.

Theorème 2.1 ([125]). *Soit $L_j \in \mathbb{R}^{(N-\bar{n})\times M_j(K)}$ définie comme en (2.38) mais avec l'application de Veronese de degré j, ν_j. Si $N \ge M_s(K) + \bar{n} - 2$ et si les points $\{\bar{\varphi}(t)\}_{t=\bar{n}+1}^N$ sont distribués d'une façon*

générique[1] *sur les hyperplans* $\mathcal{H}_i$ *avec au moins* $K-1$ *points par hyperplan, alors :*

$$\text{rang}(L_j) = \begin{cases} > M_j(K)-1, & j < s \\ = M_j(K)-1, & j = s \\ < M_j(K)-1 & j > s \end{cases}$$

Le nombre d'états discrets (ou de sous-modèles) est donc donné par

$$s = \max\{j : \text{rang}(L_j) = M_j(K)-1\}. \tag{2.39}$$

Disposant désormais du nombre d'états discrets s, on va maintenant procéder à l'estimation des paramètres θ_i. Considérons à cet effet la dérivée de $P_s(z)$

$$\nabla P_s(z) = \frac{\partial P_s(z)}{\partial z} = \sum_{j=1}^{s} \prod_{l\neq j} (\theta_l^\top z)\theta_j. \tag{2.40}$$

Si z est dans $\mathcal{H}_j \setminus \cup_{\substack{i=1 \\ i\neq j}}^{s} \mathcal{H}_i$ c'est-à-dire, si z est dans l'hyperplan $\mathcal{H}_j$ à l'exclusion de tous les autres hyperplans $\mathcal{H}_i$, $i = 1,\ldots,s$, $i \neq j$, alors $\theta_j^\top z = 0$ et donc $\nabla P_s(z)$ est réduit seulement au terme $\prod_{l\neq j}(\theta_l^\top z)\theta_j$, qui est proportionnel au vecteur θ_j (cf. Figure 2.6). Le premier élément de θ_j étant égal à l'unité, on déduit :

$$\theta_j = \left.\frac{\nabla P_s(z)}{e_1^\top \nabla P_s(z)}\right|_{z\in\mathcal{H}_j},$$

où e_i est un vecteur de $\mathbb{R}^K$ possédant 1 à la $i$ème entrée et et 0 partout ailleurs. Cette formule montre qu'on peut estimer θ_j à partir de la dérivée du polynôme multivariable P_s mais à condition de connaître au moins un certain point z_j dans chaque hyperplan $\mathcal{H}_j, j = 1,\ldots,s$. Si z_0 est un élément non nul quelconque de $\mathbb{R}^K$ et v un élément de $\mathbb{R}^K$ non colinéaire à z_0 et vérifiant $\theta_j^\top v \neq 0\ \forall j$, $\mathcal{L} = \{z_0 + \alpha v, \alpha \in \mathbb{R}\}$ définit une droite affine dans l'espace $\mathbb{R}^K$. On peut facilement, compte tenu des contraintes imposées, montrer que la droite $\mathcal{L}$ non parallèle à aucun des hyperplans $\mathcal{H}_j$ admet un unique point d'intersection $z_j = z_0 + \alpha_j v$ avec chacun d'eux, α_j étant une racine du polynôme monovariable $Q_s(\alpha) = P_s(z_0 + \alpha v)$ [131]. Ainsi, on peut,

- en fixant $z_0 \in \mathbb{R}^K \neq 0$, $v \in \mathbb{R}^K$ tel que , $v \neq \lambda z_0\ \forall\lambda$ et $P_s(v) \neq 0$,
- en trouvant les racines α_i de $Q_s(\alpha)$ et en posant $z_j = z_0 + \alpha_j v$,

calculer tous les vecteurs de paramètres

$$\theta_j = \frac{\nabla P_s(z_j)}{e_1^\top \nabla P_s(z_j)},\ j = 1,\ldots,s. \tag{2.41}$$

Dans [126], il est suggéré de choisir z_0 et v de la façon suivante : $z_0 = e_{K-1}$ et $v = e_K$. Notons néanmoins que la recherche des racines du polynôme monovariable $Q_s(\alpha)$ peut vite devenir fastidieuse

1. C'est-à-dire, avec au moins $M_s(K)$ points en position générale. Un ensemble de points d'un espace euclidien $\mathbb{R}^K$ sont dits en position (configuration) générale, s'il n'existe aucun sous-groupe de $K+1$ points dans cet ensemble qui soit inclus dans un sous-espace de dimension $K-1$ (hyperplan).

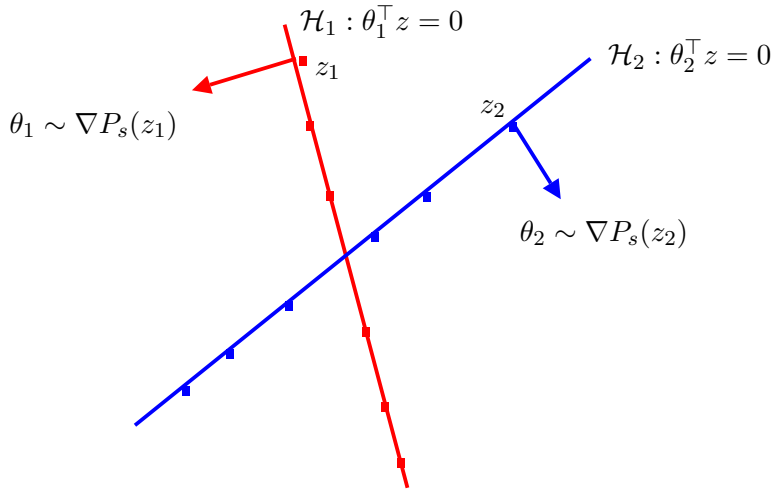


FIGURE 2.6 – Exemple de deux sous-modèles avec $\bar{\varphi} \in \mathbb{R}^2$.

si s devient grand. Un résumé de l'ensemble de la procédure d'identification algébro-géométrique de systèmes commutants est fourni dans l'Algorithme 2.2.

Algorithme 2.2 Résumé de l'algorithme algébro-géométrique

1. Former à partir des données, la matrice L_s définie en (2.38),
2. Obtenir h par exemple comme le dernier vecteur singulier (à droite) de L_s,
3. Pour tout $j = 1, \ldots, s$, trouver un point z_j vérifiant $z_j \in \mathcal{H}_j \setminus \cup_{\substack{i=1 \\ i \neq j}}^{s} \mathcal{H}_i$ en suivant par exemple une technique comme celle décrite ci-dessus,
4. Trouver ensuite les vecteurs de paramètres θ_j, $j = 1, \ldots, s$, par la formule (2.41).

Une fois s et θ_j obtenus, on peut reconstruire l'état discret du système c'est-à-dire déterminer pour chaque instant $t \geq \bar{n}$, l'indice du modèle local actif à ce instant.

$$\lambda_t = \arg \min_{i=1,\ldots,s} (\theta_i^\top \bar{\varphi}(t))^2. \tag{2.42}$$

Remarque 2.2. *Si les ordres n_a et n_b des différents sous-modèles sont parfaitement connus, alors il n'est pas nécessaire de connaître exactement le nombre s de sous-modèles pour identifier les paramètres. En fait, on a juste besoin de disposer d'une borne supérieure $\bar{s}$ de s. En paramétrant alors la matrice L_s définie par l'équation (2.38) avec $\bar{s} \geq s$, on peut toujours trouver le vrai paramètre hybride h à une constante multiplicative près, pourvu qu'il n' y ait pas de répétition de facteurs (d'ordre un) dans $P_{\bar{s}}$ (cf. [126]).*

Dans cette présentation de la méthode algébro-géométrique, il a été supposé que les données sont parfaites, c'est-à-dire non perturbées par du bruit. En présence de bruit, la contrainte de découplage hybride (2.34) n'est plus rigoureusement vérifiée. En fait, le produit considéré peut même être très éloigné

de zéro. Cela pourra avoir le désagréable effet de dégrader les performances de l'algorithme algébro-géométrique. Par exemple, le nombre de sous-modèles ne peut plus être obtenu aussi facilement que dans le théorème 2.1. De même, une optimisation peut être nécessaire au calcul des vecteurs d'évaluation z_j utilisés dans (2.41). Pour plus de commentaires, nous nous référons à [124]. Dans le cadre de ce manuscrit, nous développerons au chapitre 5, une extension de cette méthode algébrique à l'estimation de systèmes MIMO commutants, dans le contexte général où les ordres sont inconnus et peuvent différer d'un sous-modèle à un autre et le nombre de sous-modèles est également inconnu.

2.4.2 Estimation de modèles d'état commutants

Les méthodes présentées jusqu'ici sont relatives à l'identification de modèles entrée-sortie, affines par morceaux ou commutants du type ARX. Ce sont généralement des modèles entrée-sortie consistant en une prédiction de la sortie future à partir des observations passées. Ces modèles sont souvent assez peu commodes pour décrire un système multivariable. C'est certainement la raison qui a poussé quelques chercheurs à faire des propositions pour l'identification (par les méthodes des sous-espaces) de systèmes MIMO commutants décrits par des représentations d'état [93], [92], [119], [22]. Les modèles considérés sont de la forme

$$\begin{cases} x(t+1) &= A_{\lambda_t}x(t) + B_{\lambda_t}u(t) + w(t) \\ y(t) &= C_{\lambda_t}x(t) + D_{\lambda_t}u(t) + v(t), \end{cases} \tag{2.43}$$

où les vecteurs $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^{n_u}$, $y(t) \in \mathbb{R}^{n_y}$ sont respectivement l'état, l'entrée et la sortie du système. L'indice $\lambda_t \in \{1, \ldots, s\}$ fait référence à l'état discret du système, s est le nombre d'états discrets du système hybride et n est l'ordre des sous-modèles. Les matrices $A_{\lambda_t}, B_{\lambda_t}, C_{\lambda_t}, D_{\lambda_t}$ indexées par l'état discret sont les paramètres à estimer.
Une difficulté supplémentaire dans l'identification d'un modèle du type (2.43) par rapport à l'identification d'un modèle commutant entrée-sortie, est que l'état continu $x(t)$ est inconnu. Ainsi, il devient assez fastidieux de déterminer une quelconque partition de l'espace état-entrée. Du fait de cette difficulté, les méthodes disponibles pour l'estimation d'un modèle du type (2.43) se bornent essentiellement à l'application des méthodes d'identification des sous-espaces [115] entre deux instants de commutation consécutifs. Pour ce faire, on a besoin de supposer que les instants de commutation sont séparés par une certaine durée minimum appelée temps de séjour. A la différence des méthodes entrée-sortie, les méthodes d'identification de modèles d'état commutants s'appuient sur des techniques de détection de sauts de modèles. A titre indicatif, nous présentons dans le reste de cette section (Sous-sections 2.4.3 et 2.4.4), les méthodes développées respectivement dans [119] et [93].

2.4.3 Méthode de V. Verdult et M. Verhaegen

La méthode présentée dans [119] est basée sur une application de la technique d'identification MOESP [120] à l'estimation des sous-modèles d'un modèle commutant du type (2.43). Dans ce but, les auteurs font l'hypothèse importante que l'état discret λ_t, $t = 1, \ldots, N$, ainsi que le nombre de sous-modèles s sont complètement connus. Les instants de commutation sont supposés être séparés par une période

minimum $\tau_{\text{dwell}} > f$ où f est un certain entier vérifiant $f > n$. De plus, il est supposé que le bruit de procédé $w(t)$ est nul dans le modèle (2.43). La séquence $\{v(t)\}$ du bruit de sortie est de moyenne nulle et est non corrélée avec l'entrée $\{u(t)\}$.

Si l'on se restreint à un intervalle du type $[t,\, t+f-1]$ où un seul sous-modèle linéaire i est actif, alors on peut écrire l'équation

$$y_f(t) = \Gamma_f^{(i)} x(t) + H_f^{(i)} u_f(t) + v_f(t), \tag{2.44}$$

où

$$\Gamma_f^{(i)} = \begin{bmatrix} C_i \\ C_i A_i \\ \vdots \\ C_i A_i^{f-1} \end{bmatrix} \in \mathbb{R}^{f n_y \times n}, \quad H_f^{(i)} = \begin{bmatrix} D_i & 0 & \cdots & 0 \\ C_i B_i & D_i & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ C_i A_i^{f-2} B_i & \cdots & C_i B_i & D_i \end{bmatrix} \in \mathbb{R}^{f n_y \times f n_u}$$

et

$$\begin{aligned} u_f(t) &= \begin{bmatrix} u(t)^\top & \cdots & u(t+f-1)^\top \end{bmatrix}^\top \in \mathbb{R}^{f n_u}, \\ y_f(t) &= \begin{bmatrix} y(t)^\top & \cdots & y(t+f-1)^\top \end{bmatrix}^\top \in \mathbb{R}^{f n_y}, \\ v_f(t) &= \begin{bmatrix} v(t)^\top & \cdots & v(t+f-1)^\top \end{bmatrix}^\top \in \mathbb{R}^{f n_y}. \end{aligned} \tag{2.45}$$

Maintenant, pour des raisons de commodité dans la présentation, posons

$$Y = \begin{bmatrix} y_f(1) & \cdots & y_f(N) \end{bmatrix} \quad \text{et} \quad U = \begin{bmatrix} u_f(1) & \cdots & u_f(N) \end{bmatrix}.$$

Nous définissons également $Y^{(i)}$, $i = 1, \ldots, s$, comme la matrice dont les colonnes consistent en une collection de tous les vecteurs $y_f(t)$ tels que t soit dans $\mathscr{I}_i = \{t : \lambda_k = i \, \forall k \in [t,\, t+f-1]\}$. D'une façon similaire à $Y^{(i)}$, nous formons $U^{(i)}$ et $V^{(i)}$ à partir de $u_f(t)$ et $v_f(t)$ respectivement. Alors, à partir de l'équation (2.44), on peut écrire

$$Y^{(i)} = \Gamma_f^{(i)} X^{(i)} + H_f^{(i)} U^{(i)} + V^{(i)}, \quad i = 1, \ldots, s.$$

Puisque les données impliquées dans cette équation sont relatives au même sous-modèle linéaire i, on peut procéder à l'identification du sous-modèle i en utilisant les méthodes classiques d'identification des sous-espaces. Dans [119], cela est effectué au moyen de l'algorithme PI-MOESP [120] dont le principe peut être décrit comme suit :

1) Réaliser la factorisation RQ suivante des matrices de données

$$\begin{bmatrix} U^{(i)} \\ Z^{(i)} \\ Y^{(i)} \end{bmatrix} = \begin{bmatrix} R_{11}^{(i)} & 0 & 0 \\ R_{21}^{(i)} & R_{22}^{(i)} & 0 \\ R_{31}^{(i)} & R_{32}^{(i)} & R_{33}^{(i)} \end{bmatrix} \begin{bmatrix} Q_1^{(i)} \\ Q_2^{(i)} \\ Q_3^{(i)} \end{bmatrix},$$

où $Z^{(i)}$ est une matrice d'instruments [74] construite à partir des entrées passées.

2) Alors on peut, sous certaines conditions de persistence d'excitation [120], montrer que les relations asymptotiques

$$\begin{aligned}\lim_{N^{(i)}\to\infty}\frac{1}{\sqrt{N^{(i)}}}R_{31}^{(i)} &= \lim_{N^{(i)}\to\infty}\frac{1}{\sqrt{N^{(i)}}}\left(\Gamma_f^{(i)}X^{(i)}(Q_1^{(i)})^\top + H_f^{(i)}R_{11}^{(i)}\right)\\ \lim_{N^{(i)}\to\infty}\frac{1}{\sqrt{N^{(i)}}}R_{32}^{(i)} &= \lim_{N^{(i)}\to\infty}\frac{1}{\sqrt{N^{(i)}}}\Gamma_f^{(i)}X^{(i)}(Q_2^{(i)})^\top,\end{aligned} \tag{2.46}$$

où $N^{(i)} = \text{card}\,(\mathscr{I}_i)$, sont vérifiées.

3) Bien que $N^{(i)}$ ne soit pas infini, il est possible de supposer que ces égalités sont approximativement satisfaites (sans les limites) pour $N^{(i)}$ assez grand. Ainsi,

$$\begin{aligned}R_{31}^{(i)} &= \Gamma_f^{(i)}X^{(i)}(Q_1^{(i)})^\top + H_f^{(i)}R_{11}^{(i)},\\ R_{32}^{(i)} &= \Gamma_f^{(i)}X^{(i)}(Q_2^{(i)})^\top.\end{aligned} \tag{2.47}$$

Si $\text{rang}\left(X^{(i)}(Q_2^{(i)})^\top\right) = n$ (condition de persistence d'excitation), alors l'espace colonne de $\Gamma_f^{(i)}$ est égal à celui de $R_{32}^{(i)}$. Pour recouvrer une base de ce sous-espace, une décomposition en valeurs singulières de $R_{32}^{(i)}$ peut être réalisée, soit

$$R_{32}^{(i)} = \begin{bmatrix}\mathcal{U}^{(i)} & \mathcal{U}_\perp^{(i)}\end{bmatrix}\begin{bmatrix}\mathcal{S}^{(i)} & 0\\ 0 & \mathcal{S}_\perp^{(i)}\end{bmatrix}\begin{bmatrix}\mathcal{V}^{(i)}\\ \mathcal{V}_\perp^{(i)}\end{bmatrix},$$

où $\mathcal{S}^{(i)}$ est une matrice diagonale contenant les n valeurs singulières dominantes. On peut alors (à une multiplication à droite par une matrice inversible près) poser arbitrairement $\widehat{\Gamma}^{(i)} = \mathcal{U}^{(i)}$. Il en résulte, grâce à la structure de $\Gamma_f^{(i)}$ que

$$\hat{C}_i = \mathcal{U}^{(i)}(1:n_y,:) \quad \text{et} \quad \hat{A}_i = \left[\mathcal{U}^{(i)}(1:(f-1)n_y,:)\right]^\dagger \mathcal{U}^{(i)}(f+1:fn_y,:),$$

où le symbole † fait référence à la pseudo-inverse.

4) Etant donné les estimées $\hat{A}_i$ et $\hat{C}_i$, on s'intéresse à l'extraction de $\hat{B}_i$ et $\hat{D}_i$ à partir de la première équation de (2.47). En multipliant celle-ci à gauche par $(\mathcal{U}_\perp^{(i)})^\top$ et à droite par $(R_{11}^{(i)})^{-1}$, il vient

$$(\mathcal{U}_\perp^{(i)})^\top R_{31}^{(i)}(R_{11}^{(i)})^{-1} = (\mathcal{U}_\perp^{(i)})^\top H_f^{(i)}.$$

En considérant les partitions

$$(\mathcal{U}_\perp^{(i)})^\top R_{31}^{(i)}(R_{11}^{(i)})^{-1} = \begin{bmatrix}M_1^{(i)} & \cdots & M_f^{(i)}\end{bmatrix} \quad \text{et} \quad (\mathcal{U}_\perp^{(i)})^\top = \begin{bmatrix}L_1^{(i)} & \cdots & L_f^{(i)}\end{bmatrix},$$

avec $M_q^{(i)} \in \mathbb{R}^{fn_y\times n_u}$ et $L_q^{(i)} \in \mathbb{R}^{fn_y\times n_y}$, $q = 1,\ldots,s$, il est possible d'exploiter facilement la forme de

$H_f^{(i)}$ pour écrire

$$\begin{bmatrix} L_1^{(i)} & L_2^{(i)} & \cdots & L_{f-1}^{(i)} & L_f^{(i)} \\ L_2^{(i)} & L_3^{(i)} & \cdots & L_f^{(i)} & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ L_f^{(i)} & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} I_{n_y} & 0 \\ 0 & \mathcal{U}^{(i)}\big(1:(f-1)n_y,:\big) \end{bmatrix} \begin{bmatrix} \hat{D}_i \\ \hat{B}_i \end{bmatrix} = \begin{bmatrix} M_1^{(i)} \\ M_2^{(i)} \\ \vdots \\ M_f^{(i)} \end{bmatrix},$$

d'où les matrices $\hat{B}_i$ et $\hat{D}_i$, $i = 1, \ldots, s$ peuvent être déduites par simple calcul.

Les matrices d'état $(\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i)$, $i = 1, \ldots, s$, ainsi estimées pour les sous-modèles ne sont pas nécessairement définies par rapport à la même base de coordonnées de l'état continu $x(t)$. Il convient donc de les rapporter à des bases concordantes (cf. Chapitre 4). Pour ce faire, il est proposé dans [119], de calculer des matrices de changement de base en exploitant les valeurs prises par l'état $x(t)$ aux différents instants de commutation (qui sont supposés être exactement connus). Cela s'inspire d'une idée présentée premièrement dans [128]. Pour présenter leur méthode, considérons un instant de commutation τ où le système commute d'un sous-modèle i à un sous-modèle j. Alors on peut calculer l'état $x(\tau)$ de deux manières différentes en utilisant

– d'une part, les matrices du sous-modèle i (départ)

$$\begin{aligned} \hat{x}^i(\tau) &= A_i^f x(\tau - f) + \Delta_f^{(i)} U(:, \tau - f) \\ &= A_i^f (\Gamma_f^{(i)})^\dagger \Big(Y(:, \tau - f) - H_f^{(i)} U(:, \tau - f) \Big) + \Delta_f^{(i)} U(:, \tau - f) \\ &= A_i^f (\Gamma_f^{(i)})^\dagger Y(:, \tau - f) + \Big(\Delta_f^{(i)} - A_i^f (\Gamma_f^{(i)})^\dagger H_f^{(i)} \Big) U(:, \tau - f), \end{aligned} \tag{2.48}$$

où la relation $x(\tau - f) = (\Gamma_f^{(i)})^\dagger \big(Y(:, \tau - f) - H_f^{(i)} U(:, \tau - f) \big)$ a été utilisée, avec

$$\Delta_f^{(i)} = \begin{bmatrix} A_i^{f-1} B_i & \cdots & A_i B_i & B_i \end{bmatrix}.$$

– d'autre part, les matrices du sous-modèle j (arrivée)

$$\hat{x}^j(\tau) = (\Gamma_f^{(j)})^\dagger \big(Y(:, \tau) - H_f^{(j)} U(:, \tau) \big). \tag{2.49}$$

En fait, ces deux états $\hat{x}^i(\tau)$ et $\hat{x}^j(\tau)$ représentent l'état du système (2.43) à l'instant τ mais relativement à des bases différentes. Ainsi, il existe une matrice T_{ij} inversible d'ordre n telle que

$$\hat{x}^i(\tau) = T_{ij} \hat{x}^j(\tau).$$

Si T_{ij} était connue, il suffirait alors de changer les matrices (A_j, B_j, C_j, D_j) du sous-modèle j en $(T_{ij} A_j T_{ij}^{-1}, T_{ij} B_j, C_j T_{ij}^{-1}, D_j)$ pour ramener les matrices de ce dernier sous-modèle dans une base qui concorde avec celle du sous-modèle i. Mais comme T_{ij} n'est pas connue, elle va être calculée en considérant l'ensemble de toutes les commutations $i \to j$ et $j \to i$. Soient $\tau_1, \tau_2, \cdots$, les instants de ces

commutations. Alors, en calculant prudemment les états $\hat{x}^i$ et $\hat{x}^j$ (la formule (2.48) est utilisée pour le sous-modèle de départ tandis que la formule (2.49) est appliquée pour le sous-modèle d'arrivée de la commutation), on peut écrire

$$\begin{bmatrix} \hat{x}^i(\tau_1) & \hat{x}^i(\tau_2) & \cdots \end{bmatrix} = T_{ij} \begin{bmatrix} \hat{x}^j(\tau_1) & \hat{x}^j(\tau_2) & \cdots \end{bmatrix}. \tag{2.50}$$

Si la matrice $\begin{bmatrix} \hat{x}^j(\tau_1) & \hat{x}^j(\tau_2) & \cdots \end{bmatrix}$ est de rang n, on peut déterminer T_{ij} à partir de l'équation (2.50). En généralisant cette procédure à l'ensemble de toutes les commutations survenant dans le système, il est possible de rapporter les matrices de tous les sous-modèles estimés à des bases d'état qui concordent (cf. [119] pour plus de commentaires).

En pratique, la méthode d'identification de modèles commutants présentée dans [119] pourrait souffrir de quelques inconvénients :

- L'hypothèse selon laquelle les instants de commutation sont parfaitement connus est forte car, en pratique cela n'est généralement pas le cas.
- A moins que le niveau de bruit dans les données ne soit très faible, cette méthode ne permet pas de ramener réellement les modèles dans des bases *cohérentes*. L'approximation de l'état à partir de matrices estimées (donc entachées de bruit) et de données bruitées comme dans les formules (2.48) et (2.49), peut créer une déviation par rapport à la base commune convenue.

Si l'on considère par exemple un système simple d'ordre 2, commutant *une seule fois* entre deux sous-modèles, alors on ne peut pas déterminer les matrices T_{ij} par la relation (2.50) et cela, indépendamment de la taille de l'échantillon de données disponibles. En fait, l'application de la formule (2.50) pour la détermination des matrices T_{ij} requiert que le système commute un certain nombre minimum de fois (supérieur ou égal à n). Au-delà de la méthode proposée dans [119], la détermination d'un modèle global commutant à partir des matrices de ses sous-modèles individuels est un problème très délicat.

2.4.4 Méthode de K. M. Pekpe *et al.*

Le modèle considéré dans [93] est différent de la forme classique du modèle à commutations donnée en (2.43). Il s'agit d'un ensemble de s sous-modèles linéaires évoluant indépendamment les uns des autres, c'est-à-dire, ne partageant pas le même état continu. Les équations de ces sous-modèles sont données par

$$\begin{cases} x^i(t+1) &= A_i x^i(t) + B_i u(t) + w^i(t) \\ y^i(t) &= C_i x^i(t) + D_i u(t) + v^i(t), \quad i = 1, \ldots, s, \end{cases} \tag{2.51}$$

où $x^i(t) \in \mathbb{R}^{n_i}$ est l'état du sous-modèle i, $w^i(t) \in \mathbb{R}^{n_i}$ est le bruit agissant sur cet état, $y^i(t) \in \mathbb{R}^{n_y}$ et $v^i(t) \in \mathbb{R}^{n_y}$ sont respectivement la sortie du sous-modèle i et le bruit associé. Le vecteur $u(t) \in \mathbb{R}^{n_u}$ est l'entrée commune à tous les sous-modèles. Dans cette étude, les auteurs de l'article [93] supposent que les bruits $v^i(t)$ et $w^i(t)$ sont des bruits blancs gaussiens non correlés avec l'entrée $u(t)$. La sortie du

système global est donnée à chaque instant par une des sorties des différents sous-modèles, soit,

$$y(t) = \sum_{i=1}^{s} p_i(t) y_i(t),$$

avec pour tout t,

$$p_i(t) \in \{0, 1\} \quad \text{et} \quad \sum_{i=1}^{s} p_i(t) = 1.$$

Avec $f \ll \tau_{\text{dwell}}$ un entier petit par rapport au temps de séjour τ_{dwell}, définissons les vecteurs $u_{f+1}(t)$ et $w_f^i(t)$ comme dans l'équation (2.45) à partir de $u(t)$ et $w^i(t)$ respectivement. Alors pour tout $t > f$, on a

$$y^i(t) = C_i A_i^f x^i(t-f) + H_{i,f} u_{f+1}(t-f) + K_{i,f} w_f^i(t-f) + v^i(t), \tag{2.52}$$

où

$$H_{i,f} = \begin{bmatrix} C_i A_i^{f-1} B_i & \cdots & C_i B_i & D_i \end{bmatrix} \in \mathbb{R}^{n_y \times (f+1) n_u},$$

$$K_{i,f} = \begin{bmatrix} C_i A_i^{f-1} & \cdots & C_i A_i & C_i \end{bmatrix} \in \mathbb{R}^{n_y \times f n_i}.$$

Si chaque sous-modèle du modèle (2.51) est stable, alors pour f suffisamment grand, on peut négliger le terme $C_i A_i^f x^i(t-f)$ dans l'équation (2.52) de sorte qu'il reste seulement

$$y^i(t) \simeq H_{i,f} u_{f+1}(t-f) + K_{i,f} w_f^i(t-f) + v^i(t). \tag{2.53}$$

En écrivant maintenant l'équation précédente pour tout $t \in [k,\ k+N-1]$, avec $k \geq f+1$ et $N \gg f+1$ un certain entier, on peut former la relation matricielle suivante

$$Y_{k,N}^{(i)} \simeq H_{i,f} U_{k-f,f+1,N} + K_{i,f} W_{k-f,f,N}^{(i)} + V_{k,N}^{(i)}, \tag{2.54}$$

avec

$$Y_{k,N}^{(i)} = \begin{bmatrix} y(k) & \cdots & y(k+N-1) \end{bmatrix} \in \mathbb{R}^{n_y \times N},$$

$$V_{k,N}^{(i)} = \begin{bmatrix} v^i(k) & \cdots & v^i(k+N-1) \end{bmatrix} \in \mathbb{R}^{n_y \times N},$$

$$W_{k-f,f,N}^{(i)} = \begin{bmatrix} w_f^i(k-f) & \cdots & w_f^i(k+N-f-1) \end{bmatrix} \in \mathbb{R}^{f n_i \times N},$$

$$U_{k-f,f+1,N} = \begin{bmatrix} u_{f+1}(k-f) & \cdots & u_{f+1}(k+N-f-1) \end{bmatrix} \in \mathbb{R}^{(f+1) n_u \times N}.$$

Dans la notation $U_{p,q,r}$, p désigne l'indice temporel du premier élément de la matrice $U_{p,q,r}$, q fait référence au nombre d'instants concernés par chaque colonne de $U_{p,q,r}$ (qn_u lignes) et r est le nombre de colonnes. Contrairement à la méthode de la sous-section 2.4.3, les instants de commutation sont supposés inconnus dans [93]. Ainsi, pour pouvoir obtenir les paramètres de chaque sous-modèle, les auteurs commencent

par déterminer les poids $p_i(t)$. Cela est effectué en suivant une stratégie de détection de changement. En fait, si on projette l'équation précédente orthogonalement sur le complément orthogonal de l'espace des lignes de la matrice $U_{k-f,f+1,N}$, on obtient [1]

$$Y^{(i)}_{k,N}\Pi^{\perp}_{U_{k-f,f+1,N}} \simeq \left(K_{i,f}W^{(i)}_{k-f,f,N} + V^{(i)}_{k,N}\right)\Pi^{\perp}_{U_{k-f,f+1,N}},$$

où $\Pi^{\perp}_{U_{k-f,f+1,N}}$ est la matrice de projection définie par

$$\Pi^{\perp}_{U_{k-f,f+1,N}} = I_N - U^{\top}_{k-f,f+1,N}(U_{k-f,f+1,N}U^{\top}_{k-f,f+1,N})^{-1}U_{k-f,f+1,N}.$$

Il ne reste donc plus qu'une matrice de résidus formée seulement de bruit. En notant

$$\varepsilon(k) = Y^{(i)}_{k,N}\Pi^{\perp}_{U_{k-f,f+1,N}}e_N \in \mathbb{R}^{n_y}, \tag{2.55}$$

avec $e_N = \begin{bmatrix}0 & \cdots & 0 & 1\end{bmatrix}^{\top} \in \mathbb{R}^N$ la dernière colonne de cette matrice de résidus, une stratégie de détection de changement peut être mise en place au travers d'une inspection de la moyenne de $\varepsilon(k)$, $k = f+1, f+2, \ldots$ En procédant récursivement avec la fenêtre glissante $[k,\ k+N-1]$, avec k variant [2] de $f+1$ à $f+M$, cette moyenne devrait être nulle tant qu'il n'y a pas de commutation et non nulle en cas de changement de mode [93].

Une fois la séquence de commutation (c'est-à-dire les poids $p_i(t)$) complètement estimée, on peut procéder au calcul des paramètres de chaque sous-modèle. Pour cela, on forme des matrices blocs de données comme à l'équation (2.54) sur l'horizon $[f+1,\ f+M]$, où $f+M$ est le plus grand indice temporel de donnée disponible.

$$Y^{(i)}_{f+1,M} \simeq H_{i,f}U_{f+1,f+1,M} + K_{i,f}W^{(i)}_{f+1,f,M} + V^{(i)}_{f+1,M}$$

En minimisant alors le critère

$$\left\|Y^{(i)}_{f+1,M}P_i - H_{i,f}U_{f+1,f+1,M}P_i\right\|^2_F, \quad i = 1,\ldots,s,$$

où $P_i = \text{diag}\,(p_i(f+1),\cdots,p_i(f+M)) \in \mathbb{R}^{M\times M}$, et $\|\cdot\|$ fait référence à la norme de Frobenius, on obtient

$$\hat{H}_{i,f} = Y^{(i)}_{f+1,M}P_iU^{\top}_{f+1,f+1,M}\left(U_{f+1,f+1,M}P_iU^{\top}_{f+1,f+1,M}\right)^{-1}, \quad i = 1,\ldots,s.$$

Cependant, il est important de noter que, à cause du bruit affectant les données, des erreurs de détection sont probables. De ce fait, le nombre s de sous-modèles pourrait être surestimé lors de la détermination des poids $p_i(t)$. Donc, des procédures correctives de classification ou de fusion peuvent être appliquées aux paramètres $\hat{H}_{i,f}$ dans le but de réduire ces erreurs (cf. [93] pour plus de détails). Etant donné les matrices $\hat{H}_{i,f}$ dites paramètres de Markov, les ordres des différents sous-modèles ainsi que leurs

1. En fait, $\Pi^{\perp}_{U_{k-f,f+1,N}}$ est calculable si l'entrée $\{u(t)\}$ et l'entier N sont tels que $\text{rang}(U_{k-f,f+1,N}) = (f+1)n_u$.
2. Nous convenons que $f+M$ est la taille de l'échantillon de données entrée-sortie $\{u(t), y(t)\}$ disponibles.

matrices respectives (A_i, B_i, C_i, D_i), i=1,. . . ,s, peuvent être déterminées en appliquant l'algorithme ERA (Eigenvalue Realization Algoithm) [93].

Une caractéristique notable du modèle (2.51) est qu'il ne souffre pas du problème de concordance de bases d'état comme le modèle usuel (2.43) de systèmes commutants (cf. Sous-section 2.4.3). En effet, les différents sous-modèles linéaires n'interagissent pas mais évoluent indépendamment les uns des autres si bien qu'on peut représenter chacun d'eux dans une base arbitraire. Mais l'application de la méthode présentée dans [93] pour l'identification d'un tel modèle (selon la formulation des auteurs) peut nécessiter un temps de séjour très important [1]. Cela tient au fait que, pour pouvoir négliger le terme $C_i A_i^f x^i(t-f)$ dans l'équation (2.52), il est nécessaire que le rayon spectral $\rho(A)$ de A soit très petit ou que $f \ll \tau_{\text{dwell}}$ soit très grand. De plus, le calcul de $\Pi^{\perp}_{U_{k-f,f+1,N}}$ est possible si $\text{rang}(U_{k-f,f+1,N}) = (f+1)n_u$. Avec f très grand, cette condition devient forte car elle nécessite alors que l'entrée $u(t)$ soit fréquentiellement beaucoup plus riche que dans la méthode de la sous-section 2.4.3 par exemple.

2.5 Conclusion

Nous avons fait dans ce chapitre une synthèse significative des techniques d'identification de systèmes hybrides qui ont été développées dans la littérature. On a vu que la plupart de ces méthodes s'appliquent à l'identification de modèles entrée sortie représentés par des modèles ARX, généralement SISO. En général, les méthodes présentées font au moins l'une des hypothèses simplificatrices suivantes :

- les ordres des différents sous-modèles sont égaux et sont connus *a priori* [46, 67, 87, 131, 101, 99],
- le nombre de sous-modèles est connu *a priori* [46, 67, 101, 131, 99],
- les instants de commutation sont connus *a priori* [119],
- il existe un certain temps de séjour minimum dans chaque état discret [93, 119, 22],

Les travaux présentés dans [46, 67, 87, 131, 101, 99] considèrent l'identification de modèles ARX, SISO mais parfois MISO tandis que les contributions [93, 119, 22] s'intéressent à des modèles d'état avec l'hypothèse d'un temps de séjour minimum.

Cependant, peu de travaux concernent l'identification de systèmes MIMO commutants, sans connaissance *a priori* ni des ordres (ceux-ci pouvant différer d'un sous-modèle à un autre) des sous-modèles, ni du nombre de sous-modèles, ni des instants de commutation [93, 119, 22]. La suite du manuscrit est consacrée au développement de méthodes qui puissent surmonter simultanément ces challenges. Nous considérons d'abord des modèles d'état au Chapitre 4 puis des modèles ARX commutants au Chapitre 5. Mais l'identification de modèles d'état pour des systèmes commutants requiert certaines propriétés essentielles comme la possibilité de fixer ou d'avoir un contrôle sur la base de coordonnées de l'état du système et l'extraction des ordres des différents sous-modèles surtout quand ces ordres peuvent être différents. Pour cela, nous commençons au chapitre suivant par proposer quelques nouvelles méthodes des sous-espaces (qui sont plus appropriées, du fait d'un certain nombre de propriétés, au cas de systèmes commutants)

1. En fait, selon la nature particulière du modèle (2.51), il est possible de s'affranchir de cette hypothèse de temps de séjour minimum. Puisqu'il n'y a pas d'interaction entre les sous-modèles (l'évolution de chaque sous-modèle ne dépendant pas des autres sous-modèles), on peut, en assimilant par exemple le modèle (2.53) à un modèle entrée-sortie du type $y(t) = Q_{\lambda_t} u_{f+1}(t-f) + \textit{bruit}$, surmonter l'hypothèse de temps de séjour.

pour l'identification d'un seul modèle linéaire. Au chapitre 4, ces méthodes seront généralisées à un ensemble de plusieurs modèle linéaires, commutant entre eux.

3

Identification de modèles d'états linéaires

Sommaire

3.1 Introduction

Après avoir fait dans le chapitre 2 un état de l'art des techniques d'identification de systèmes hybrides, nous allons maintenant présenter, à partir du présent chapitre, les principaux développements de cette thèse. Comme nous l'avons annoncé en introduction, l'objectif ultime est bien entendu de pouvoir estimer les paramètres d'un modèle hybride linéaire à partir d'un certain nombre d'observations entrée-sortie. Puisqu'un système linéaire hybride est concrètement une collection de sous-systèmes linéaires en interaction, nous commençons par considérer l'identification d'un seul système et plus précisément, d'un modèle linéaire d'état à partir de données de mesures. Les algorithmes développés nous permettront au Chapitre 4 de traiter le problème plus complexe de l'identification de systèmes hybrides dont les différents sous-systèmes sont représentés par des modèles d'états linéaires.

Dans ce chapitre, nous présentons quelques algorithmes d'identification de modèles d'état pour la description de systèmes linéaires. Par rapport à la littérature existante [72], [115], ces méthodes possèdent entre autres les caractéristiques intéressantes de ne nécessiter aucune Décomposition en Valeurs Singulières (DVS) et de permettre un choix *a priori* de la base des matrices à estimer dans l'espace d'état. Ces qualités, comme nous le verrons, sont essentielles dans la résolution des équations de systèmes linéaires hybrides.
La méthode dite du propagateur [84], [80] est introduite, révisée puis étendue sous certaines hypothèses, à l'identification de systèmes multivariables. Nous établissons ensuite un rapprochement entre la famille des méthodes d'estimation de modèles d'état et celle des méthodes dédiées à l'identification des modèles entrée-sortie. Il est démontré, à travers la proposition d'une technique originale de construction de modèles d'état de systèmes multivariables sous forme canonique, que la seconde famille peut présenter quelques avantages en termes de charge de calcul, d'implémentation récursive ou de nombre de paramètres à estimer. Finalement, une troisième méthode d'identification de modèles d'état est proposée pour suppléer aux insuffisances des deux premières. Elle consiste en une transformation du problème d'estimation de sous-espaces en un problème de moindres carrés. L'identification des sous-espaces est ainsi converti en un problème de régression classique avec un régresseur complètement connu.

3.2 Motivations et formulation du problème

3.2.1 Motivations

L'identification de systèmes dynamiques linéaires a fait, ces dernières années, l'objet d'un développement remarquable sous l'impulsion des méthodes dites des sous-espaces (S-S) [115], [72], [123]. Le principal apport de ces méthodes par rapport aux techniques plus traditionnelles d'erreur de prédiction [74] est de permettre l'estimation directe de représentations minimales d'état non nécessairement canoniques. Cela a le mérite d'éviter la fastidieuse tâche qu'est la construction de modèles d'état pour des systèmes multi-entrées-multi-sorties (MIMO) à partir d'équations entrée-sortie. La représentation d'état est souvent préférée au modèle entrée-sortie (E-S) pour la compacité avec laquelle elle encode les paramètres des systèmes multivariables. De plus, elle se prête plus naturellement, compte tenu des

méthodes algébriques et analytiques disponibles dans la littérature, à l'analyse de systèmes en termes d'observabilité, de commande et de filtrage.

Cependant, l'application des méthodes des sous-espaces à l'identification de certains types de systèmes comme les systèmes composites avec des sous-modèles linéaires [118], les systèmes à commutations [119], [9] où à l'identification récursive [76], [89] pourraient s'accompagner d'un certain nombre de difficultés techniques. En effet, la majorité des méthodes des sous-espaces connues passent par une étape de DVS pour décider arbitrairement de la base de la réalisation à estimer. Un problème important est que cette DVS est difficile à appliquer, par exemple dans le cadre de l'identification en ligne de systèmes, parce qu'elle est coûteuse en charge de calcul et techniquement ardue à mettre à jour de façon récursive [76], [81]. Un deuxième problème est lié à la multiplicité des représentations d'état possibles pour un même système. Cela est assez problématique, par exemple dans le cas des systèmes multi-modèles ou hybrides, où les réalisations d'état des différents sous-modèles à estimer doivent être relatives à la même base (ou à des bases « cohérentes ») [9], [118], [119]. Or l'utilisation de la DVS ne garantit pas une base d'estimation constante au cours d'une procédure d'identification en ligne par exemple. Elle ne permet pas non plus d'obtenir les différents sous-modèles composant un système multi-modèles dans la même base d'état. Cela pourrait ainsi fausser une éventuelle reconstruction du comportement entrée-sortie [113].

Notre objectif dans ce chapitre est de proposer des méthodes basées sur les techniques sous-espaces et qui soient applicables à l'identification de systèmes à commutations. Pour cela, nous proposons de remédier aux problèmes mentionnés ci-dessus à travers la présentation de trois nouveaux algorithmes d'identification de modèles d'état multivariables.

Le premier algorithme consiste en une revisite puis en une extension de la méthode du propagateur [82], [80], [84] à l'identification de systèmes MIMO. Cela se fait grâce à un partitionnement particulier de la matrice d'observabilité de façon à isoler une base de l'espace des lignes de cette dernière matrice. En tirant alors avantage de ce partitionnement, nous présentons également une méthode pour l'estimation de l'ordre du système aussi bien hors-ligne que dans un contexte en-ligne.

Une alternative aux méthodes classiques S-S, comme suggérée dans [109], pourrait être de recourir à des structures intermédiaires de la forme E-S ou à des représentations canoniques qui présentent notamment l'avantage d'être directes en ce sens qu'un régresseur est immédiatement disponible pour l'estimation. Cette caractéristique des modèles E-S peut constituer un atout remarquable en cas d'implémentation récursive, permettant ainsi d'éviter le recours à la DVS généralement nécessaire dans le cadre des méthodes des sous-espaces. Notre deuxième algorithme établit donc un parallèle entre les méthodes d'identification des sous-espaces et les méthodes à erreur de prédiction. A partir de données supposées générées par un système représenté par un modèle d'état, nous dérivons un modèle du type ARMAX [74]. Une fois que cette structure est correctement estimée, il est possible de retourner à une réalisation d'état du système générateur des données. Une méthode originale de passage des relations entrée-sortie à une réalisation dans la base canonique d'observabilité est alors présentée.

Le troisième algorithme, comme les deux précédents, exploite le concept de transformation similaire dans l'espace d'état pour fixer la base de représentation. Cela est effectué par l'introduction d'une *matrice de pondération* (notée Λ_f), qui, multipliée par la matrice d'observabilité étendue, ne détruit pas

les propriétés de rang de cette dernière. Contrairement aux deux premières méthodes dont l'application requiert une hypothèse un peu restrictive sur la matrice de dynamique, notre troisième méthode est plus générale. Nous montrons que, sous l'hypothèse d'observabilité du système considéré, si la matrice de pondération est générée de façon aléatoire suivant une distribution uniforme par exemple, la méthode proposée permet de retrouver presque sûrement une réalisation consistante du système.

Le plan du chapitre est le suivant. La sous-section 3.2.2 formule le problème d'identification des sous-espaces. Dans la section 3.3, nous revisitons la méthode du propagateur pour l'identification de systèmes MISO. Dans la section 3.4, nous généralisons la méthode du propagateur à des systèmes MIMO. Après cela, nous proposons dans la section 3.5 d'étudier l'alternative d'estimation directe de réalisations canoniques d'état de systèmes. Finalement, dans la section 3.6, nous généralisons le concept d'identification structurée de modèles d'état qui constitue aussi la charpente de nos deux premières méthodes. Des simulations numériques permettent à la section 3.7, d'illustrer la faisabilité de nos méthodes.

3.2.2 Formulation du problème

Considérons un système linéaire à temps invariant, représenté par le modèle d'état à temps discret suivant.

$$\left\{ \begin{array}{rcl} x(t+1) & = & Ax(t) + Bu(t) + w(t) \\ y(t) & = & Cx(t) + Du(t) + v(t), \end{array} \right. \tag{3.1}$$

où $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^{n_u}$, $y(t) \in \mathbb{R}^{n_y}$ sont respectivement, les vecteurs d'état, d'entrée et de sortie. Les vecteurs $w(t) \in \mathbb{R}^n$ et $v(t) \in \mathbb{R}^{n_y}$ symbolisent les bruits de système et de mesure qui seront supposés blancs. A, B, C, D sont les matrices du système relativement à une certaine base de l'espace d'état.

Le problème d'identification peut alors se formuler de la façon suivante : étant donné des mesures entrée-sortie $\{u(t), y(t)\}$, générées par un système du type (3.1), estimer à une transformation similaire près, les matrices (A, B, C, D).

Pour commencer, définissons les vecteurs

$$u_f(t) = \begin{bmatrix} u(t)^\top & \cdots & u(t+f-1)^\top \end{bmatrix}^\top \in \mathbb{R}^{fn_u} \tag{3.2}$$

et $y_f(t) \in \mathbb{R}^{fn_y}$, $w_f(t) \in \mathbb{R}^{fn}$, $v_f(t) \in \mathbb{R}^{fn_y}$, de la même manière que $u_f(t)$, f étant un entier quelconque.

Parallèlement à ces vecteurs de signaux, nous définissons aussi les matrices de paramètres

$$\Gamma_f = \begin{bmatrix}(C)^\top & (CA)^\top & \cdots & (CA^{f-1})^\top\end{bmatrix}^\top \in \mathbb{R}^{fn_y \times n},$$

$$H_f = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ CA^{f-2}B & \cdots & CB & D \end{bmatrix} \in \mathbb{R}^{fn_y \times fn_u}, \tag{3.3}$$

$$G_f = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ CA^{f-2} & \cdots & C & 0 \end{bmatrix} \in \mathbb{R}^{fn_y \times fn}.$$

Nous utiliserons aussi parfois les notations fonctionnelles du type $\Gamma_f(A, C)$ et $H_f(A, B, C, D)$ pour désigner respectivement les matrices Γ_f et H_f de l'équation (3.3). De l'équation récurrente (3.1), on obtient facilement, par substitutions successives sur un horizon temporel f,

$$y_f(t) = \Gamma_f x(t) + H_f u_f(t) + e_f(t),\ t \geq 1 \tag{3.4}$$

avec $e_f(t) = G_f w_f(t) + v_f(t)$.

Posons maintenant

$$\begin{aligned} X_{t,N} &= \begin{bmatrix} x(t) & \cdots & x(t+N-1)\end{bmatrix} \in \mathbb{R}^{n \times N}, \\ U_{t,f,N} &= \begin{bmatrix} u_f(t) & \cdots & u_f(t+N-1)\end{bmatrix} \in \mathbb{R}^{fn_u \times N}, \\ Y_{t,f,N} &= \begin{bmatrix} y_f(t) & \cdots & y_f(t+N-1)\end{bmatrix} \in \mathbb{R}^{fn_y \times N}, \\ E_{t,f,N} &= \begin{bmatrix} e_f(t) & \cdots & e_f(t+N-1)\end{bmatrix} \in \mathbb{R}^{fn_y \times N}, \end{aligned} \tag{3.5}$$

où N est un entier tel que $f \ll N$ et défini de façon compatible avec le nombre total de mesures entrée-sortie disponibles. Alors, l'équation (3.4) peut, en posant $t = f + 1$, se mettre sous la forme compacte

$$Y_{f+1,f,N} = \Gamma_f X_{f+1,N} + H_f U_{f+1,f,N} + E_{f+1,f,N}. \tag{3.6}$$

Dans la littérature de l'identification par les méthodes des sous-espaces, les matrices de données de la forme $Y_{1,f,N}$ et $Y_{f+1,f,N}$ sont souvent référencées comme respectivement les données *passées* et *futures* [115]. Nous employons ici la même terminologie. Afin d'alléger les notations nous utilisons dans l'ensemble du manuscrit, du moins chaque fois que cela sera possible, les notations simplifiées

$$Y = Y_{f+1,f,N},\quad U = U_{f+1,f,N},\quad E = E_{f+1,f,N},\quad X = X_{f+1,N}.$$

Ainsi, l'équation (3.6) peut s'écrit

$$Y = \Gamma_f X + H_f U + E. \tag{3.7}$$

Partant de cette équation de blocs de données, les méthodes dites des sous-espaces s'attachent à extraire d'abord, soit la séquence d'état X, soit la matrice d'observabilité étendue Γ_f au moyen de techniques de projection géométrique et de DVS [115, 123, 98]. Une seconde étape consiste à déduire de Γ_f ou de X, les matrices du système dans une base arbitraire de coordonnées de l'espace d'état. L'efficacité de ces méthodes a été expérimentée et reconnue au travers de nombreuses applications réelles [68], [35].

Cependant, comme déjà indiqué dans l'introduction de ce chapitre, les méthodes S-S existantes sont difficilement applicables à l'identification de certains types de systèmes. Elles souffrent en effet de quelques problèmes techniques liés à la multiplicité des bases de coordonnées possibles pour l'état du système (problématique dans le cas des systèmes composites avec plusieurs sous-modèles linéaires locaux) [118] et de la DVS qui possède un coût d'implémentation important, notamment dans les applications récursives [76].

Afin de remédier à ces problèmes, en vue d'une application aux systèmes à commutations (cf. Chapitre 4), nous développons dans ce chapitre de nouvelles méthodes d'identification de systèmes MIMO appartenant à la famille des méthodes des sous-espaces. Mais avant d'entrer dans les détails des différentes méthodes présentées dans ce chapitre, nous commençons par remarquer qu'une identification consistante des matrices impliquées dans l'équation (3.7) n'est possible que si les données de mesure satisfont à certaines propriétés de richesse dans un certain sens. Ainsi, nous commençons par introduire les conditions nécessaires à l'obtention d'estimées consistantes. Pour cela, considérons la propriété suivante qui caractérise la richesse fréquentielle de l'entrée exogène $\{u(t)\}$.

Définition 3.1 (entrée suffisamment excitante)**.** *Le signal d'entrée* $\{u(t)\} \subset \mathbb{R}^{n_u}$ *est dit Suffisamment Excitant (SE) d'ordre au moins* l*, s'il existe un entier* N_0 *et un indice temporel* t_0 *tel que* $\text{rang}(U_{t,l,N_0}) = ln_u$ *pour tout* $t \leq t_0$*. Cette propriété de* $\{u(t)\}$ *est exprimée en disant simplement que* $\{u(t)\}$ *est SE*(l)*.*

La définition 3.1 diffère de la conception classique de la persistence d'excitation [74] qui concerne des séquences infinies de signaux. Ici, par analogie avec la formulation donnée dans [4], nous caractérisons cette propriété sur un horizon fini, indépendamment de la nature stochastique ou déterministe de $\{u(t)\}$. Cette démarche se justifie par le fait qu'en pratique, le nombre de données disponibles pour l'identification est généralement fini. C'est pourquoi nous préférons ici l'expression entrée *suffisamment* excitante à entrée *continuellement* excitante.

Si l'on réécrit l'équation (3.7) sous la forme

$$Y = \begin{bmatrix} \Gamma_f & H_f \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} + E, \tag{3.8}$$

alors, on voit que pour estimer les paramètres, il faudrait que la matrice $\begin{bmatrix} X^\top & U^\top \end{bmatrix}^\top$ soit de rang plein ligne. Cette propriété peut être obtenue comme une conséquence de la définition 3.1.

Proposition 3.1. *Supposons le système (3.1) atteignable et considérons que les termes $w(t)$ et $v(t)$ sont identiquement nuls. Alors l'assertion suivante est vraie.*
Si $\{u(t)\}$ est SE$(n+f)$ dans le sens de la Définition 3.1, alors

$$\text{rang}\left(\begin{bmatrix} X_{f+1,N} \\ U_{f+1,f,N} \end{bmatrix}\right) = n + fn_u,$$

où $N \geq N_0 + n$.

Preuve. Une preuve est donnée dans l'annexe A.1. ■

Un résultat similaire à la Proposition 3.1 était déjà formulé dans [60] mais avec une définition différente de la propriété de la persistence d'excitation. Pour une utilisation future, nous énonçons aussi la proposition suivante.

Proposition 3.2. *Supposons le système (3.1) atteignable et observable et considérons que les bruits $w(t)$ et $v(t)$ sont identiquement nuls dans (3.1). Alors, les assertions suivantes sont équivalentes.*

1. $\text{rang}\left(\begin{bmatrix} X \\ U \end{bmatrix}\right) = n + fn_u.$
2. $\text{rang}(X\Pi_U^{\perp}) = n$, *où*
$$\Pi_U^{\perp} = I_N - U^{\top}(UU^{\top})^{-1}U, \tag{3.9}$$
I_N étant la matrice identité d'ordre N.
3. $\text{rang}\left(\begin{bmatrix} Y \\ U \end{bmatrix}\right) = n + fn_u.$
4. $\text{rang}(Y\Pi_U^{\perp}) = n.$

Preuve. La preuve de la proposition est basée sur le lemme suivant.

Lemme 3.1 ([60])**.** *Considérons la matrice bloc définie par*

$$S = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

avec $A \in \mathbb{R}^{n\times n}$ and $D \in \mathbb{R}^{m\times m}$. Si D est non singulière, alors

$$\text{rang}(S) = m + \text{rank}\left(A - BD^{-1}C\right). \tag{3.10}$$

1) $\Longleftrightarrow$ 2) : Commençons par remarquer que

$$\begin{bmatrix} X \\ U \end{bmatrix}\begin{bmatrix} X \\ U \end{bmatrix}^{\top} = \begin{bmatrix} XX^{\top} & XU^{\top} \\ UX^{\top} & UU^{\top} \end{bmatrix}.$$

Alors, en utilisant l'identité (3.10), on a

$$\begin{aligned}\operatorname{rang}\left(\begin{bmatrix}X\\U\end{bmatrix}\right) &= fn_u + \operatorname{rang}\left(XX^\top - XU(UU^\top)^{-1}X^\top\right)\\ &= fn_u + \operatorname{rang}\left((X\Pi_U^\perp)(X\Pi_U^\perp)^\top\right)\\ &= fn_u + \operatorname{rang}\left(X\Pi_U^\perp\right),\end{aligned}$$

d'où on peut conclure que

$$\operatorname{rang}\left(\begin{bmatrix}X\\U\end{bmatrix}\right) = n + fn_u \Longleftrightarrow \operatorname{rang}(X\Pi_U^\perp) = n.$$

3) $\Longleftrightarrow$ 1) : Le résultat se déduit immédiatement de la relation

$$\begin{bmatrix}Y\\U\end{bmatrix} = \underbrace{\begin{bmatrix}\Gamma_f & H_f\\ 0 & I_{fn_u}\end{bmatrix}}_{M}\begin{bmatrix}X\\U\end{bmatrix} \tag{3.11}$$

car le système étant observable (c'est-à-dire $\operatorname{rang}(\Gamma_f) = n$), la matrice M est de rang plein colonne.

4) $\Longleftrightarrow$ 3) : Il suffit d'appliquer la même démarche que dans la preuve de l'équivalence 1) $\Longleftrightarrow$ 2). ■

Après avoir présenté les propriétés d'une excitation persistante, nous pouvons maintenant nous focaliser sur le problème d'identification du système (3.1). Cela nous conduit à rappeler brièvement la méthode dite du propagateur et destinée justement à l'estimation d'un sous-espace engendré par une certaine collection d'observations.

3.3 Méthode du Propagateur

3.3.1 Principe de la méthode

La méthode du propagateur a été développée dans [84] pour le traitement de signaux d'antennes et adaptée récemment par G. Mercère dans [80] à l'identification récursive des sous-espaces. Afin de présenter le principe de cette méthode, considérons le problème qui consiste à « estimer » le sous-espace vectoriel engendré par les colonnes d'une certaine matrice $\Gamma \in \mathbb{R}^{p\times n}$, $p \geq n$, de rang n à partir d'observations $z(1), \cdots, z(N) \in \mathbb{R}^p$ appartenant à ce sous-espace. Bien évidemment, nous entendons par estimation de sous-espace, la détermination d'une base quelconque de ce dernier. Alors, pour tout t, il existe un vecteur $x(t)$ tel que

$$z(t) = \Gamma x(t). \tag{3.12}$$

Notons d'abord que cette équation ne détermine pas Γ de façon unique puisque si Γ est solution de (3.12), alors pour $T \in \mathbb{R}^{n\times n}$, non singulière, ΓT^{-1} est aussi solution de (3.12) pour peu que $x(t)$ soit pris égal à $\bar{x}(t) = Tx(t)$. Si les n premières lignes de Γ sont linéairement indépendantes, alors Γ peut

être partitionnée de la façon suivante

$$\Gamma = \begin{bmatrix} \Gamma^1 \\ \Gamma^2 \end{bmatrix},$$

où $\Gamma^1 \in \mathbb{R}^{n\times n}$ est une matrice carrée non singulière et $\Gamma^2 \in \mathbb{R}^{(p-n)\times n}$. Puisque Γ est de rang n, on peut écrire Γ^2 comme une combinaison linéaire des lignes de Γ^1. Il existe donc une unique matrice $\Psi \in \mathbb{R}^{(p-n)\times n}$ telle que $\Gamma^2 = \Psi\Gamma^1$. Ainsi, on a

$$\begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} \Gamma^1 \\ \Psi\Gamma^1 \end{bmatrix} x(t) = \begin{bmatrix} I_n \\ \Psi \end{bmatrix} \bar{x}(t),$$

avec $\bar{x}(t) = \Gamma^1 x(t)$ et $z(t) = \begin{bmatrix} z_1(t)^\top & z_2(t)^\top \end{bmatrix}^\top$ est une partition de $z(t)$ conforme à celle de Γ. Il suffit maintenant de trouver la matrice Ψ, appelée propagateur [84], [79] pour disposer d'une base de l'espace des colonnes de Γ. De l'équation précédente, il est immédiat que Ψ vérifie $z_2(t) = \Psi z_1(t)$ pour tout t, d'où il vient

$$\hat{\Psi} = \arg\min_{\Psi} \|Z_2 - \Psi Z_1\|_F^2,$$

où $\|\cdot\|_F$ est la norme matricielle de Frobenius, Z_1 et Z_2 résultent d'une partition de $Z = \begin{bmatrix} z(1), \cdots, z(N) \end{bmatrix}$ semblable à celle de $z(t)$.

3.3.2 Adaptation à l'identification de systèmes MISO

Nous allons maintenant illustrer la méthode du propagateur [80] sur le problème d'identification de la matrice d'observabilité étendue Γ_f à partir de l'équation (3.7). Pour cela, supposons que le système (3.1) soit un système monosortie (MISO) et notons $c^\top = C \in \mathbb{R}^{1\times n}$. Nous faisons pour l'instant abstraction du bruit, ce qui signifie que nous considérons $E = 0$. Afin d'éliminer le terme HU dans (3.7), nous commençons par projeter orthogonalement l'ensemble de cette équation sur le sous-espace orthogonal à l'espace des lignes de U. Il vient :

$$Y\Pi_U^\perp = \Gamma_f X \Pi_U^\perp,$$

avec $\Pi_U^\perp$ définie comme en (3.9). Le système étant MISO et observable, si nous considérons $f > n$, alors la matrice $\Gamma_n(A, c^\top)$ constituée des n premières lignes de Γ_f est carrée et non singulière. En suivant le principe de la méthode du propagateur, on peut alors partitionner Γ_f de la façon suivante

$$\Gamma_f = \begin{bmatrix} \Gamma_n(A, c^\top) \\ \mathscr{P}_f \Gamma_n(A, c^\top) \end{bmatrix}, \tag{3.13}$$

avec $\mathscr{P}_f \in \mathbb{R}^{(fn_y-n)\times n}$ et $\Gamma_n(A, c^\top) = \begin{bmatrix} c & A^\top c & \cdots & (A^\top)^{n-1} c \end{bmatrix}^\top$.
Par suite

$$\begin{bmatrix} Y_{f+1,n,N} \\ Y_{f+n+1,f-n,N} \end{bmatrix} \Pi_U^\perp = \begin{bmatrix} I_n \\ \mathscr{P}_f \end{bmatrix} \Gamma_n(A, c^\top) X \Pi_U^\perp, \tag{3.14}$$

avec $Y_{i,j,N}$ définie comme en (3.5). Comme dans la sous-section précédente, il nous suffit de calculer $\mathscr{P}_f$ pour obtenir le sous-espace de $\mathbb{R}^{fn_y}$ engendré par les colonnes de Γ_f. De l'équation précédente, on obtient

$$Y_{f+n+1,f-n,N}\Pi_U^{\perp} = \mathscr{P}_f Y_{f+1,n,N}\Pi_U^{\perp}. \tag{3.15}$$

Selon les propositions 3.1 et 3.2, en faisant l'hypothèse que le signal d'excitation $\{u(t)\}$ est suffisamment excitant d'ordre $f+n$, on obtient que

$$\text{rang}\left(Y_{f+1,n,N}\Pi_U^{\perp}\right) = \text{rang}\left(\Gamma_n(A,c^\top)X\Pi_U^{\perp}\right) = n.$$

Par conséquent $\mathscr{P}_f$ peut être déterminée de façon consistante à partir de (3.15) par

$$\mathscr{P}_f = Y_{f+n+1,f-n,N}\Pi_U^{\perp}Y_{f+1,n,N}^\top\left(Y_{f+1,n,N}\Pi_U^{\perp}Y_{f+1,n,N}^\top\right)^{-1} \tag{3.16}$$

et la matrice d'observabilité étendue Γ_f peut être ainsi obtenue dans une base particulière de l'état par

$$\bar{\Gamma}_f = \Gamma_f(\bar{A},\bar{C}) = \begin{bmatrix} I_n \\ \mathscr{P}_f \end{bmatrix}, \tag{3.17}$$

où $\bar{A} = \Gamma_n(A,c^\top)A\Gamma_n(A,c^\top)^{-1}$ et $\bar{C} = C\Gamma_n(A,c^\top)^{-1}$. Etant donné $\bar{\Gamma}_f$, l'extraction des matrices $\bar{A}$ et $\bar{C}$ (que nous notons aussi $\bar{c}^\top$ pour mettre en exergue le fait qu'il s'agit d'un vecteur ligne) correspondantes dans la base considérée s'opère traditionnellement par

$$\bar{A} = \left(\bar{\Gamma}_f^{\uparrow}\right)^{\dagger}\bar{\Gamma}_f^{\downarrow}, \quad \text{et} \quad \bar{C} = \bar{\Gamma}_f(1:n_y,:), \tag{3.18}$$

avec $n_y = 1$ dans le présent cas, $\bar{\Gamma}_f^{\uparrow} = \bar{\Gamma}_f(1:(f-1)n_y,:)$, $\bar{\Gamma}_f^{\downarrow} = \bar{\Gamma}_f(n_y+1:fn_y,:)$; le symbole $\dagger$ faisant référence à l'inverse de Moore-Penrose.

Mais une inspection de la forme particulière (3.17) de $\bar{\Gamma}_f$ permet de retrouver bien plus simplement les matrices $\bar{A}$ et $\bar{c}^\top$ que par l'équation (3.18). En effet, il est facile de voir que $\bar{\Gamma}_n = I_n$ et donc $\bar{\Gamma}_f(2:n+1,:) = \bar{\Gamma}_n\bar{A} = \bar{A}$. Par ailleurs, en utilisant le théorème de Cayley-Hamilton, on peut constater que la dernière ligne de $\bar{A}$ est donnée par

$$\begin{aligned}\mathscr{P}_f(1,:) &= \bar{c}^\top\bar{A}^n \\ &= \bar{c}^\top\left(-a_0I_n - a_1\bar{A} - \cdots - a_{n-1}\bar{A}^{n-1}\right) \\ &= \begin{bmatrix}-a_0 & \cdots & -a_{n-1}\end{bmatrix}\bar{\Gamma}_n \\ &= \begin{bmatrix}-a_0 & \cdots & -a_{n-1}\end{bmatrix},\end{aligned}$$

où les a_i, $i = 0,\ldots,n-1$, sont les coefficients du polynôme caractéristique de $\bar{A}$ défini par

$$p_{\bar{A}}(z) = \det\left(zI - \bar{A}\right) = a_0 + a_1z + \cdots + a_{n-1}z^{n-1} + z^n.$$

Finalement, on a

$$\bar{A} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix}. \tag{3.19}$$

$$\bar{c}^\top = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \tag{3.20}$$

La forme (3.19) de $\bar{A}$ est dite forme compagne horizontale [1] [69] de la matrice A. Ainsi, seule la première ligne de $\mathscr{P}_f$ est nécessaire à détermination des matrices $\bar{A}$ et $\bar{c}^\top$. On peut donc poser $f = n+1$ pour éviter d'estimer inutilement des lignes supplémentaires de $\mathscr{P}_f$. Une fois que $\bar{A}$ et $\bar{c}^\top$ sont connues, les matrices $\bar{B}$ et $\bar{d}^\top$ peuvent être obtenues par régression linéaire (voir par exemple [76] pour plus de commentaires).

Remarque 3.1. *Il est bien connu que pour un système donné, la représentation d'état (A, B, C, D) n'est pas déterminée de façon unique puisque pour toute matrice non singulière T, $(TAT^{-1}, TB, CT^{-1}, D)$ réalise aussi le comportement du même système. Ainsi, dans la méthode du propagateur, nous avons réalisé un changement de coordonnées (transformation similaire) en posant*

$$\begin{aligned} x(t) \leftarrow \bar{x}(t) &= \Gamma_n(A, c^\top)x(t), \forall\, t \\ A \leftarrow \bar{A} &= \Gamma_n(A, c^\top)A\Gamma_n(A, c^\top)^{-1}, \\ c^\top \leftarrow \bar{c}^\top &= c^\top\Gamma_n(A, c^\top)^{-1}. \end{aligned}$$

En exprimant A et $c^\top$ relativement à une base donnée, la matrice B qui en résulte par régression linéaire est nécessairement exprimée dans la même base.

3.4 Extension de la méthode du Propagateur aux systèmes multivariables (MIMO)

La méthode du propagateur discutée dans la section précédente est basée sur un partitionnement de la matrice d'observabilité étendue sous la forme (3.13). Ainsi, l'application de cette méthode suppose que les n premières lignes de Γ_f (à une permutation de lignes près) sont linéairement indépendantes. Il faut donc pouvoir repérer à l'avance dans Γ_f (sans pour autant connaître cette dernière), les positions de n lignes linéairement indépendantes qu'on pourrait ramener dans les n premières positions grâce à une permutation de lignes. Cependant, dans le cas des systèmes MIMO, la caractérisation d'un tel nombre de lignes linéairement indépendantes est un problème délicat puisque Γ_f n'est pas connu. Dans cette

1. Chez certains auteurs, l'appellation *forme compagne* est réservée plutôt à la transposée $\bar{A}^\top$ de $\bar{A}$.

section, nous développons, sous certaines hypothèses qui seront précisées dans la suite, une extension de la méthode dite du propagateur à l'identification de systèmes MIMO.

Les résultats qui vont suivre seront d'une grande importance au Chapitre 4 puisque certains seront notamment généralisés à l'identification plusieurs modèles linéaires en interaction. Pour cette raison, nous allons insister sur certains aspects jugés importants pour l'application envisagée.

La méthode du propagateur étant, dans sa forme dédiée à l'identification [82], théoriquement fondée seulement pour l'estimation de systèmes MISO [1], nous proposons ici de l'étendre à des systèmes MIMO. Dans ce but, nous rapportons l'observabilité des dynamiques d'état du système MIMO qui est initialement distribuée sur l'ensemble des sorties, à une seule sortie auxiliaire définie comme une combinaison linéaire de toutes les sorties. De cette sortie auxiliaire, on peut ainsi observer la totalité des dynamiques du système, c'est-à-dire que la matrice d'observabilité qui est relative à cette sortie est de rang plein. Par conséquent, un système MISO ayant le même état $x(t)$ que le système MIMO original peut être estimé par la méthode du propagateur. On en déduit ensuite les matrices du système initial.

Nous considérons le modèle suivant [2] (c'est-à-dire le modèle (3.1) sans bruit de processus $w(t)$)

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ \quad y(t) = Cx(t) + Du(t) + v(t), \end{cases} \tag{3.21}$$

où nous rappelons que $u(t) \in \mathbb{R}^{n_u}$, $y(t) \in \mathbb{R}^{n_y}$, $x(t) \in \mathbb{R}^n$ et $v(t) \in \mathbb{R}^{n_y}$ sont respectivement les vecteurs d'entrée, de sortie, d'état et de bruit de sortie. (A, B, C, D) sont les matrices du système relativement à une certaine base de coordonnées de l'espace d'état. Nous considérons que $\{x(t)\}$, $\{u(t)\}$, $\{y(t)\}$ et $\{v(t)\}$, $t \in \mathbb{Z}$ sont des processus stochastiques ergodiques et faiblement stationnaires [74].

Le problème d'identification considéré peut maintenant s'énoncer de la façon suivante. Des réalisations $\{u(t)\}_{t=1}^{N_s}$ et $\{y(t)\}_{t=1}^{N_s}$ des processus d'entrée et de sortie générées par un modèle de la forme (3.21) sur un horizon fini mais suffisamment large, estimer la dimension minimum n du processus d'état et les matrices (A, B, C, D) du modèle (3.21) à une transformation similaire près.

Nous commençons par faire un certain nombre d'hypothèses.

- A1. Le processus d'entrée $\{u(t)\}$ est ergodique, stationnaire et est suffisamment excitant d'ordre au moins $f + n$, avec f défini comme à la section 3.2.2.
- A2. Le bruit de sortie $\{v(t)\}$ est un bruit blanc de moyenne nulle et est statistiquement non correlé avec le processus d'entrée $\{u(t)\}$. Plus explicitement, pour tous t, s, on a $\mathbb{E}\left[v(t)v(s)^\top\right] = \delta_{ts}\sigma_v^2 I_{n_y}$ et $\mathbb{E}\left[u(t)v(s)^\top\right] = 0$, où $\mathbb{E}[\,]$ indique l'espérance mathématique et δ est le nombre de Kronecker.
- A3. La matrice A du modèle (3.21) est asymptotiquement stable c'est-à-dire que toutes les valeurs propres de A sont strictement localisées à l'intérieur du cercle unité.
- A4. Le modèle (3.21) est minimal c'est-à-dire que, (A, B) est atteignable et (A, C) est observable.

1. Cette méthode peut néanmoins s'appliquer à certains MIMO dont les n premières lignes de la matrice d'observabilité étendue Γ_f sont linéairement indépendantes. 2. En fait, l'élimination du bruit d'état simplifiera la dérivation des résultats asymptotiques données dans la suite.

3.4.1 Problématique

Dans le cas des systèmes MIMO, la forme compagne du type (3.19) n'est plus aussi facile à obtenir. Cependant, nous allons montrer qu'il est toujours possible, sous certaines hypothèses, de décrire le système MIMO avec une matrice d'état A sous une forme canonique compagne.

Il peut arriver par exemple que tous les pôles du système soient observables à partir d'une seule sortie y_j. Cela signifie que $\Gamma_n(A, c_j^\top)$, avec $c_j^\top \in \mathbb{R}^{1\times n}$ la j-ème ligne de C, est de rang plein. Dans un tel cas, la séquence d'état du système pourrait être entièrement estimée en utilisant cette seule sortie y_j tout comme dans le cas des systèmes MISO (et ainsi, la matrice A peut être obtenue sous forme compagne). Mais cette condition est loin d'être toujours réalisée pour un système MIMO général, car toutes les autres sorties sont susceptibles d'introduire des dynamiques qui pourraient ne pas être visibles depuis y_j.

On peut trouver dans la littérature quelques tentatives d'identification structurée qui s'appuient sur une estimation de représentations d'état sous des formes canoniques variées. Ces structures de modèle résultent bien souvent d'une paramétrisation particulière des matrices d'état en se servant des invariants de Kronecker ou des indices d'observabilité et de commandabilité [1], [69]. Un inconvénient évident de ces types de schémas est que les invariants de Kronecker ne sont pas toujours connus *a priori* (puisque nous ne connaissons pas les matrices du système). L'idée suggérée ici permet de surmonter cette difficulté grâce à une transformation du système initial (MIMO) en un système intermédiaire assimilable à un MISO observable dont nous estimons ensuite une forme d'observabilité (3.20) et (3.19). Cependant, comme nous le verrons dans la suite, cette transformation requiert que la matrice de dynamique A soit similaire à une matrice de la forme $A_c = \bar{A}$ en (3.19), ce qui est le cas si et seulement si A est non dérogatoire [1][57], [83].

Considérons une sortie auxiliaire $y_a(t)$ construite comme une combinaison linéaire de l'ensemble des sorties du système

$$y_a(t) = \sum_{j=1}^{n_y} \gamma_j y_j(t), \tag{3.22}$$

où les $y_j(t)$ désignent les composantes du vecteur de sortie $y(t)$ et les γ_j sont des nombres réels non nuls. Si A est non dérogatoire, les γ_j peuvent être sélectionnés de façon à rendre toutes les dynamiques du système observables à partir de $y_a(t)$ seulement. En utilisant alors cette sortie combinée et les mesures de l'entrée, il est possible d'estimer les matrices A, B et des combinaisons linéaires des lignes de C et D. Au lieu de cela, nous remplaçons juste la première composante du vecteur de sortie [2] (c'est-à-dire la première sortie) par $y_a(t)$. Le vecteur de sortie $y(t)$ devient donc $\tilde{y}(t)$ définie par

$$\tilde{y}(t) = K(\gamma)y(t), \tag{3.23}$$

1. c'est-à-dire, une matrice dont le polynôme caractéristique est égal à son polynôme minimal. Notons au passage que A non dérogatoire est une condition nécessaire à l'observabilité d'un système MISO. 2. Ce choix est arbitraire. Nous aurions pu choisir une composante quelconque de $y(t)$.

avec

$$K(\gamma) = \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_{n_y} \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{3.24}$$

Soient $\tilde{C} = K(\gamma)C$, $\tilde{D} = K(\gamma)D$, $\tilde{\Gamma}_f = \Gamma_f(A, \tilde{C})$ et $\tilde{H}_f = H_f(A, B, \tilde{C}, \tilde{D})$ les nouvelles matrices. Avec cette transformation, l'équation de données (3.6) peut se réécrire

$$\tilde{Y}_{1,f,N} = \tilde{\Gamma}_f X_{1,N} + \tilde{H}_f U_{1,f,N} + \tilde{V}_{1,f,N}. \tag{3.25}$$

Rappelons que l'objectif principal de la transformation par $K(\gamma)$ est d'arriver à une sortie $y_a(t) \in \mathbb{R}$ à partir de laquelle tous les pôles du système (3.21) pourraient être observables. Cela revient à requérir que $\Gamma_n(A, \tilde{c}_1^\top)$, avec $\tilde{c}_1^\top = \gamma^\top C$, soit de rang plein. Si cela est vérifié, il deviendra alors possible d'obtenir directement, en posant $\bar{x}(t) \leftarrow \Gamma_n(A, \tilde{c}_1^\top)x$, la matrice A de (3.21) et $\tilde{c}_1^\top$ sous les formes (3.19) et (3.20). La question est, comment trouver des nombres $\gamma_1, \cdots, \gamma_{n_y}$ de façon à satisfaire cette condition, alors que $\Gamma_n(A, C)$ est inconnu. Une réponse est donnée par la proposition 3.3.
Avant de formuler cette proposition, quelques résultats préliminaires sont nécessaires.

Theorème 3.1 (de Barnett [13])**.** *Notons $a(z) = \det(zI_n - A_c)$, le polynôme caractéristique de $A_c = \bar{A}$ définie en (3.19). Considérons un ensemble de polynômes*

$$p_i(z) = b_{in} + b_{i,n-1}z + \cdots + b_{i2}z^{n-2} + b_{i1}z^{n-1}, \quad i = 1, \ldots, m,$$

avec m un entier. Alors $\deg\left\{\text{pgcd}\left(a(z), p_1(z), \cdots, p_m(z)\right)\right\} = n - \text{rang}(P)$, *où*

$$F = \begin{bmatrix} b_{1n} & \cdots & b_{11} \\ \vdots & \ldots & \vdots \\ b_{mn} & \cdots & b_{m1} \end{bmatrix} \quad \textit{et} \quad P = \begin{bmatrix} F \\ FA_c \\ \vdots \\ FA_c^{n-1} \end{bmatrix}.$$

Lemme 3.2. *Considérons deux matrices $A \in \mathbb{R}^{n\times n}$ et $C \in \mathbb{R}^{n_y \times n}$. Alors, il existe $\gamma = \begin{bmatrix} \gamma_1 & \cdots & \gamma_{n_y} \end{bmatrix}^\top \in \mathbb{R}^{n_y}$ vérifiant* $\text{rang}\left(\Gamma_n(A, \gamma^\top C)\right) = n$ *si et seulement si A est non dérogatoire et (A, C) est observable.*

Preuve. Supposons qu'il existe un $\gamma \in \mathbb{R}^{n_y}$ tel que $\text{rang}\left(\Gamma_n(A, \gamma^\top C)\right) = n$. Alors on peut vérifier que

$$\Gamma_n(A, \gamma^\top C)A\Gamma_n(A, \gamma^\top C)^{-1}$$

est une matrice compagne de la forme donnée par (3.19). Ainsi, étant similaire à une matrice compagne, A est non dérogatoire [57]. Pour montrer que (A, C) est observable, remarquons qu'on peut écrire

$$\Gamma_n(A, \gamma^\top C) = G\Gamma_n(A, C),$$

où

$$G = \begin{bmatrix} \gamma^\top & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \gamma^\top \end{bmatrix} \in \mathbb{R}^{n \times n n_y}.$$

Avec $\text{rang}\left(\Gamma_n(A, \gamma^\top C)\right) = \text{rang}\left(G\Gamma_n(A, C)\right) = n$, on a nécessairement $\text{rang}\left(\Gamma_n(A, C)\right) \geq n$ et donc $\text{rang}\left(\Gamma_n(A, C)\right) = n$, sinon le rang de $\Gamma_n(A, \gamma^\top C)$ serait strictement inférieur à n pour tout $\gamma \in \mathbb{R}^{n_y}$. Ainsi, (A, C) est observable.

Réciproquement, supposons que A soit non dérogatoire et que (A, C) soit observable. Nous devons montrer qu'il existe alors au moins un vecteur $\gamma \in \mathbb{R}^{n_y}$ vérifiant $\text{rang}\left(\Gamma_n(A, \gamma^\top C)\right) = n$.
A étant non dérogatoire, elle est similaire à une matrice compagne de la forme de $A_c = \bar{A}$ définie en (3.19) [57]. Cela implique qu'on peut trouver une matrice L, carrée et non singulière telle que $\Gamma_n(A, C)L^{-1} = \Gamma_n(A_c, C_c)$ avec $C_c = CL^{-1}$. Soit ${c'}_j^\top$ la j-ème ligne de C_c et notons $l_j(z)$ le polynôme de degré $n-1$ dont les coefficients sont donnés par ${c'}_j^\top$ comme suit : $l_j(z) = {c'}_j(n)z^{n-1} + {c'}_j(n-1)z^{n-2} + \cdots + {c'}_j(1)$, où ${c'}_j(i)$ le i-ème élément du vecteur ${c'}_j$. Alors, nous savons grâce au théorème de Barnett [13] (rappelé ci-dessus en théorème 3.1) que $\deg\left\{\text{pgcd}\left(l_1(z), \cdots, l_{n_y}(z), p_A(z)\right)\right\} = n - \text{rang}\left(\Gamma_n(A_c, C_c)\right)$, où $p_A(z) = \det(zI_n - A_c)$. Puisque $\text{rang}\left(\Gamma_n(A_c, C_c)\right) = n$, il s'ensuit que[1] $\text{pgcd}\left(l_1(z), \cdots, l_{n_y}(z), p_A(z)\right) = 1$. En d'autres termes, les polynômes $l_1(z), \cdots, l_{n_y}(z)$, $p_A(z)$ sont relativement premiers. Cela signifie que si nous désignons par $r_1, \cdots, r_n$, les racines de $p_A(z)$, alors pour toute racine r_j de $p_A(z)$, il existe $i \in \{1, \ldots, n_y\}$ tel que $l_i(r_j) \neq 0$. Par suite, toutes les lignes $\mathcal{L}_i^\top$ de la matrice

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_1^\top \\ \vdots \\ \mathcal{L}_n^\top \end{bmatrix} \doteq \begin{bmatrix} l_1(r_1) & \cdots & l_{n_y}(r_1) \\ \vdots & \cdots & \vdots \\ l_1(r_n) & \cdots & l_{n_y}(r_n) \end{bmatrix}$$

sont non nulles. Ici, $\mathcal{L}_i = \begin{bmatrix} l_1(r_i) & \cdots & l_{n_y}(r_i) \end{bmatrix}^\top$. En ayant de nouveau recours au théorème de Barnett, dire que $\text{rang}\left(\Gamma_n(A, \gamma^\top C)\right) = \text{rang}\left(\Gamma_n(A_c, \gamma^\top C_c)\right) = n$ revient à dire que $\gamma_1 l_1(z) + \cdots + \gamma_{n_y} l_{n_y}(z)$ et $p_A(z)$ n'ont aucune racine commune ou encore que, $\mathcal{L}_i^\top \gamma \neq 0$ pour tout $i = 1, \ldots, n$. L'existence d'un tel γ est garantie par le fait que toutes les lignes de $\mathcal{L}$ sont non nulles. En effet, si cela n'était pas le cas, il existerait pour tout $\gamma \in \mathbb{R}^{n_y}$ au moins un indice i tel que $\mathcal{L}_i^\top \gamma = 0$. En d'autres termes, on aurait $\mathbb{R}^{n_y} \subset (\mathcal{L}_1)^\perp \cup \cdots \cup (\mathcal{L}_n)^\perp$, où $(\mathcal{L}_i)^\perp$ est le sous-espace orthogonal à $\mathcal{L}_i$. Avec $\mathcal{L}_i \neq 0$ pour tout i, cela est impossible (cf. par exemple Lemme A.2 en annexe). Ainsi, il existe au moins un $\gamma \in \mathbb{R}^{n_y}$ tel que $\text{rang}\left(\Gamma_n(A, \gamma^\top C)\right) = n$. ■

Proposition 3.3. *Supposons la paire (A, C) du système (3.21) observable et la matrice A non dérogatoire. Soit $\gamma = \begin{bmatrix} \gamma_1 & \cdots & \gamma_{n_y} \end{bmatrix}^\top \in \mathbb{R}^{n_y}$ un vecteur généré de façon aléatoire selon une distribution uniforme. Alors, avec une probabilité de un, on a*

$$\text{rang}\left(\Gamma_n(A, \gamma^\top C)\right) = n,$$

où $\Gamma_n\left(A, \gamma^\top C\right)$ est la matrice d'observabilité relative à la sortie combinée $y_a(t)$. En d'autres termes, l'ensemble des dynamiques du système sont observables à partir seulement de la sortie auxiliaire $y_a(t)$ presque sûrement.

1. pgcd fait référence au plus grand commun diviseur considéré ici comme un polynôme monique.

Preuve. Soit $P(\gamma) = \det\left(\Gamma_n\left(A, \gamma^\top C\right)\right)$ un polynôme en γ défini comme le déterminant de $\Gamma_n\left(A, \gamma^\top C\right) \in \mathbb{R}^{n\times n}$. Considérons l'ensemble $\mathcal{S} = \{\gamma \in \mathbb{R}^{n_y} / P(\gamma) = 0\}$, de tous les γ tels que $\text{rang}(\Gamma_n(A, \gamma^\top C)) < n$. Notons Pr la mesure de probabilité associée à la distribution uniforme, définie sur une σ-algèbre $\mathcal{R}$ (contenant $\mathcal{S}$) de $\mathbb{R}^{n_y}$. Selon le Lemme 3.2, on sait qu'il existe au moins un γ^* vérifiant $\text{rang}\left(\Gamma_n(A, {\gamma^*}^\top C)\right) = n$ et donc, $P(\gamma^*) \neq 0$; ce qui implique que le polynôme P n'est pas identiquement nul. Alors, $\mathcal{S}$ est une variété algébrique propre dans l'espace de probabilité $(\mathbb{R}^{n_y}, \mathcal{R}, \text{Pr})$, c'est-à-dire que $\mathcal{S}$ est de dimension strictement inférieur à n_y. En théorie de la mesure [51], un sous-ensemble tel que $\mathcal{S}$ est connu pour être de mesure nulle. Ainsi, la propriété $\text{rang}\left(\Gamma_n(A, \gamma^\top C)\right) = n$ est vérifiée presque sûrement, c'est-à-dire avec une probabilité de un. ■

Nous avons vu dans l'analyse précédente que dans le cas des systèmes MIMO comme dans celui des MISO, une certaine représentation d'état canonique du système sous forme compagne pouvait être directement obtenue à l'issue du procédé d'identification. Dans cet objectif, nous avons fourni une matrice non singulière T qui permet d'opérer le changement de coordonnées nécessaire.

Soit donc $T = \Gamma_n(A, \tilde{c}_1^\top) \in \mathbb{R}^{n\times n}$ et $\mathcal{I}_i = \begin{bmatrix} e_i & e_{i+n_y} & \cdots & e_{i+(f-1)n_y} \end{bmatrix} \in \mathbb{R}^{fn_y \times f}$, où e_j est le vecteur de $\mathbb{R}^{fn_y}$ qui a 1 dans sa j-ème entrée et 0 partout ailleurs. Nous définissons aussi une matrice de permutation $S \in \mathbb{R}^{fn_y \times fn_y}$ par

$$S = \begin{bmatrix} \mathcal{I}_1 & \cdots & \mathcal{I}_{n_y} \end{bmatrix}^\top \in \mathbb{R}^{fn_y \times fn_y}. \tag{3.26}$$

Maintenant en multipliant la matrice $\tilde{\Gamma}_f$ de l'équation (3.25) à gauche par S, on obtient la partition suivante :

$$S\tilde{\Gamma}_f = \left[\begin{array}{c} \Gamma_n(A, \tilde{c}_1^\top) \\ \hline \Gamma_{f-n}(A, \tilde{c}_1^\top)A^n \\ \Gamma_f(A, \tilde{c}_2^\top) \\ \vdots \\ \Gamma_f(A, \tilde{c}_{n_y}^\top) \end{array}\right] \doteq \left[\begin{array}{c} I_n \\ \hline \mathscr{P}_f \end{array}\right] T, \tag{3.27}$$

où la partie inférieure de $S\tilde{\Gamma}_f$ a été écrite comme combinaison linéaire $\mathscr{P}_f T$ des lignes de la sous-matrice T, avec $\mathscr{P}_f \in \mathbb{R}^{(fn_y - n)\times n}$. Cela est possible par le fait que $\text{rang}\left(S\tilde{\Gamma}_f\right) = \text{rang}\,(T) = n$. Ainsi, comme dans la section 3.3, l'identification de $\tilde{\Gamma}_f$ ne requiert que l'estimation de $\mathscr{P}_f$. Cette matrice correspond au propagateur défini dans la section 3.3 [79, 84, 82].

Réalisation dans une base déterminée par S. Etant donné la transformation (3.23) du système (3.21), il existe en fait différentes manières de définir la matrice de permutation S en (3.26). En effet, pour obtenir une équation de la forme (3.27), la condition à remplir est simplement que les n premières lignes de $S\tilde{\Gamma}_f$ soient linéairement indépendantes. La matrice T (que nous notons T_s) sera alors constituée de ces n lignes linéairement indépendantes. Ainsi, le choix de S détermine une certaine base de l'état. Pour estimer l'espace des colonnes de la matrice d'observabilité étendue relativement à la base ainsi fixée, on peut procéder de la manière suivante. A partir de (3.27), notons que la matrice d'observabilité

étendue [1]

$$\tilde{\Gamma}_f = S^\top \left(S\tilde{\Gamma}_f \right) = S^\top \begin{bmatrix} I_n \\ \mathscr{P}_f \end{bmatrix} T_s. \tag{3.28}$$

En appliquant un changement de coordonnées $x_s(t) \leftarrow T_s x(t)$, les matrices $(A, \tilde{C})$ deviennent $(A_s, \tilde{C}_s) = (T_s A T_s^{-1}, \tilde{C} T_s^{-1})$ de sorte que $\tilde{\Gamma}_f \leftarrow \tilde{\Gamma}_f T_s^{-1}$. Par conséquent on peut faire abstraction de T_s dans l'équation (3.28). Ainsi, on peut immédiatement extraire A_s et $\tilde{C}_s$ une fois que $\mathscr{P}_f$ est connue. Un moyen classique d'effectuer cela consiste à exploiter la propriété de A-invariance de $\tilde{\Gamma}_f$ comme à l'équation (3.18).

Réalisation sous une forme compagne horizontale. Afin d'obtenir la matrice A sous la forme (3.19), considérons que la matrice de permutation S est définie comme en (3.26). Alors, en partant de l'équation (3.27), si nous posons $\Omega = \begin{bmatrix} I_n & \mathscr{P}_f^\top \end{bmatrix}^\top$, il vient

$$\Gamma_f\big(A, \tilde{C}(1,:)\big) = \Omega\,(1:f,:)\,T_o, \tag{3.29}$$

où $\Omega(1:f,:)$ désigne les f premières lignes de Ω. Une transformation similaire qui change $x(t)$ en $x_o(t) = T_o x(t)$ induit un changement des matrices

$$(A, \tilde{C}) \quad \text{en} \quad (A_o, \tilde{C}_o) = (T_o A T_o^{-1}, \tilde{C} T_o^{-1})$$

et donc

$$\Gamma_f\big(A_o, \tilde{C}_o(1,:)\big) = \Gamma_f\big(A, \tilde{C}(1,:)\big) T_o^{-1} = \Omega\,(1:f,:)\,.$$

Notons maintenant que $\Omega\,(2:n+1,:) = \Omega\,(1:n,:)\,A_o = A_o$. Par conséquent A_o et $\tilde{C}_o(1,:)$ peuvent prendre les formes (3.19)-(3.20) :

$$A_o = \Omega\,(2:n+1,:) \quad \text{et} \quad \tilde{C}_o(1,:) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{1\times n}. \tag{3.30}$$

De la même manière, en considérant la matrice globale $S\Gamma_f(A_o, \tilde{C}_o)$ comme à l'équation (3.27), on obtient $\tilde{C}_o(j,:) = \Omega\big((j-1)f+1,:\big)$, $j = 2, \cdots, n_y$. Finalement, la matrice C_o du système initial (3.21) se calcule de manière compacte par $C_o = K(\gamma)^{-1}\tilde{C}_o$.

Etant donné les matrices A et C dans une certaine base de l'espace d'état, les matrices B et D peuvent être estimées par régression linéaire (cf. par exemple [115] pour plus de détails).

Une caractéristique intéressante de la méthode d'identification structurée proposée est que, contrairement à la plupart des méthodes des sous-espaces, elle permet d'éviter le calcul de la pseudo-inverse de $\tilde{\Gamma}_f^{\uparrow}$ comme dans la relation (3.18). Cela représente une réduction significative du coût d'implémentation numérique et peut constituer un avantage confortable dans un contexte d'identification récursive.

Dans la suite, nous adopterons chaque fois que possible, des notations simplifiées comme $\tilde{Y} = \tilde{Y}_{t,f,N}$,

1. Rappelons que toute matrice de permutation S vérifie $S^\top S = SS^\top = I$.

$\tilde{\Gamma} = \tilde{\Gamma}_f$ et ainsi de suite. De cette façon, (3.25) peut se réécrire plus simplement comme

$$\tilde{Y} = \tilde{\Gamma} X + \tilde{H} U + \tilde{V}, \tag{3.31}$$

où $\tilde{V}$ est construite à partir de $\tilde{v}(t) = K(\gamma)v(t)$ selon la formule (3.6).

3.4.2 Estimation de la matrice d'observabilité étendue

Dans cette sous-section, nous considérons le problème de l'estimation de la matrice $\mathscr{P}_f$ définie en (3.27). En s'inspirant de la classe bien connue des schémas d'identification MOESP [122], la première étape dans l'estimation de $\mathscr{P}_f$ consiste à éliminer le terme $\tilde{H}U$ dans l'équation de données (3.31), en projetant orthogonalement l'ensemble de cette équation sur le complément orthogonal de l'espace des lignes de U. Pour ce faire, nous suivrons, pour des raisons de robustesse numérique, la méthode d'implémentation RQ [122]. Il en résulte la proposition suivante.

Theorème 3.2. *Supposons les hypothèses A1-A3 réalisées. Supposons que la matrice de dynamique A soit non dérogatoire et que la matrice de permutation S soit définie comme en (3.26). Considérons la factorisation RQ des matrices de données entrée-sortie.*

$$\begin{bmatrix} U \\ \tilde{Y} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \tag{3.32}$$

où $\tilde{Y}$ est défini comme en (3.5), à partir de $\tilde{y}(t) = K(\gamma)y(t)$, et γ est tel que $\gamma^\top\gamma = 1$ et $T = \Gamma_n(A, \gamma^\top C)$ est non singulière. Alors,

$$\lim_{N\to\infty} \left(\frac{1}{N} S R_{22} R_{22}^\top S^\top \right) = \begin{bmatrix} \Sigma_{\mathsf{z}} & \Sigma_{\mathsf{z}} \mathscr{P}_f^\top \\ \mathscr{P}_f \Sigma_{\mathsf{z}} & \mathscr{P}_f \Sigma_{\mathsf{z}} \mathscr{P}_f^\top \end{bmatrix} + \sigma_v^2 \tilde{R}_v, \tag{3.33}$$

où $\mathscr{P}_f \in \mathbb{R}^{(fn_y - n)\times n}$ est la matrice qui apparaît dans (3.27),

$$\Sigma_{\mathsf{z}} = \lim_{N\to\infty} \left(\frac{1}{N} \mathsf{Z} \Pi_U^\perp \mathsf{Z}^\top \right) \in \mathbb{R}^{n\times n}, \tag{3.34}$$

$\Pi_U^\perp = I_N - U^\top \left(UU^\top\right)^{-1} U$, $\mathsf{Z} = TX$, *et*

$$\tilde{R}_v = \begin{bmatrix} I_f & \gamma_2 I_f & \cdots & \gamma_{n_y} I_f \\ \gamma_2 I_f & I_f & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{n_y} I_f & 0 & \cdots & I_f \end{bmatrix}. \tag{3.35}$$

Preuve. Une preuve est donnée dans l'annexe A.2. ■

Maintenant, en considérant la formule (3.33), nous pouvons introduire la notation

$$\Sigma = \left[\begin{array}{c|c} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{array}\right] = \bar{\Sigma} + \sigma_v^2 \tilde{R}_v, \tag{3.36}$$

avec

$$\bar{\Sigma} = \left[\begin{array}{c|c} \Sigma_{\mathsf{z}} & \Sigma_{\mathsf{z}} \mathscr{P}_f^\top \\ \hline \mathscr{P}_f \Sigma_{\mathsf{z}} & \mathscr{P}_f \Sigma_{\mathsf{z}} \mathscr{P}_f^\top \end{array}\right].$$

Ainsi, $\mathscr{P}_f$ peut être estimé en minimisant le critère sous-optimal

$$\mathcal{J}(\mathscr{P}_f) = \|\Sigma_{21} - \mathscr{P}_f \Sigma_{11}\|_F^2, \tag{3.37}$$

dont la solution est donnée par

$$\hat{\mathscr{P}}_f = \Sigma_{21} \Sigma_{11}^{-1}. \tag{3.38}$$

Mais pour que cette solution soit consistante, il est indispensable que la matrice $\Sigma_{11} = \lim_{N\to\infty} \frac{1}{N} \mathsf{Z} \Pi_U^\perp \mathsf{Z}^\top$ soit non singulière. Selon les Propositions 3.1 et 3.2, il suffit pour cela que l'entrée soit suffisamment excitante d'ordre $f+n$. Cependant, il faut noter que l'estimée (3.38) de $\mathscr{P}_f$ pourrait être biaisée en présence de bruit même blanc. Une méthode bien classique pour surmonter asymptotiquement ce problème, est celle de la variable instrumentale [76].

Remarque 3.2.

- *On peut voir à partir de (3.38) que le calcul de $\mathscr{P}_f$ ne nécessite pas une connaissance entière de la matrice Σ du (3.36). En fait, la connaissance des f (ou les n si n est connu) premières lignes ou colonnes est suffisante.*
- *Il est évident aussi que la formule (3.30) par laquelle on détermine A et C, n'exploite pas la matrice $\mathscr{P}_f$ dans sa totalité. En réalité, il est seulement nécessaire de connaître quelques n_y lignes de $\mathscr{P}_f$.*
- *De ces deux observations, il apparaît que les matrices du système peuvent être extraites après élimination de quelques redondances, de l'équation de données*

$$\left[\begin{array}{c} \bar{\bar{Y}} \\ \hline \mathcal{Y}^{2:n_y} \end{array}\right] = \left[\begin{array}{c} \Gamma_f(A, \gamma^\top C) \\ \hline C(2:n_y,:) \end{array}\right] X + \left[\begin{array}{c|c} H_f(A, B, \gamma^\top C, \gamma^\top D) & \\ \hline D(2:n_y,:) & O_{(n_y-1)\times(f-1)n_u} \end{array}\right] \bar{\bar{U}},$$

où

$$\begin{aligned} \bar{\bar{U}} &= U_{f+1,f,N} \in \mathbb{R}^{f n_u \times N}, \\ \bar{\bar{Y}} &= \begin{bmatrix} y_{af}(f+1) & \cdots & y_{af}(f+N) \end{bmatrix} \in \mathbb{R}^{f\times N}, \\ \mathcal{Y}^{2:n_y} &= \begin{bmatrix} y_{2:n_y}(f+1) & \cdots & y_{2:n_y}(f+N) \end{bmatrix} \in \mathbb{R}^{(n_y-1)\times N}, \end{aligned}$$

avec $y_{af}(t) = \begin{bmatrix} y_a(t) & y_a(t+1) & \cdots & y_a(t+f-1) \end{bmatrix}^\top \in \mathbb{R}^f$, $y_a(t) \in \mathbb{R}$ *étant la sortie combinée*

définie par (3.22) et $y_{2:n_y}(t) = \begin{bmatrix} y_2(t) & \cdots & y_{n_y}(t) \end{bmatrix}^\top$ *indique les* $n_y - 1$ *dernières entrées du vecteur de sortie à l'instant* t*. Cela constitue une réduction remarquable des dimensions des matrices de données puisque contrairement à* $Y_{f+1,f,N}$ *défini par (3.5) (*fn_y *lignes),* $\bar{\bar{Y}}$ *contient seulement* $f + n_y - 1$ *lignes.*

3.4.3 Estimation de l'ordre

Afin de pouvoir estimer convenablement la matrice $\mathscr{P}_f$ à l'aide de la relation (3.38), on a besoin de déterminer d'abord l'ordre n du système (3.21). Généralement, du point de vue des schémas d'identification des sous-espaces, l'extraction de l'ordre s'opère presque toujours par une analyse des valeurs singulières [15] des matrices de données comme R_{22} en (3.32) par exemple. Dans cette sous-section, nous présentons une alternative à la procédure traditionnelle pour l'estimation de l'ordre.
Plus précisément, nous aimerions caractériser l'ordre sans avoir besoin de recourir à la très coûteuse DVS. L'objectif de cette démarche est d'arriver à un algorithme qui soit applicable en ligne. Bien que l'ordre soit inconnu, nous pouvons supposer qu'un majorant strict $r_{\max} = f > n$ de ce dernier est disponible *a priori* (hypothèse qui est à la base de tout algorithme d'identification des sous-espaces). Alors, l'idée consiste à exploiter la structure intéressante de la matrice Σ définie en (3.36).
Pour des raisons de clarté, nous traiterons d'abord le cas où les données sont déterministes puis celui où les données sont potentiellement sujettes à du bruit.

3.4.3.1 Cas déterministe

Pour procéder à l'estimation de l'ordre, nous considérons récursivement une sous-matrice de Σ définie en (3.36) de la forme $\Delta_r = \Sigma(1:r, 1:r)$ pour r allant de $r_{\min}$ vers $r_{\max}$ avec $r_{\min} < n < r_{\max}$. En référence à la formule (3.34), une condition nécessaire à l'obtention de l'ordre est $\text{rang}(X\Pi_U^\perp) = n$ ce qui, sous l'hypothèse que le système (3.21) est minimal, revient à supposer $\{u(t)\}$ est SE$(f+n)$ (cf. Propositions 3.1 et 3.2). Cette hypothèse implique en effet que la matrice de covariance Σ_{z} définie par (3.34) est définie-positive. Par conséquent, toute sous-matrice de Σ_{z} de la forme $\Delta_r = \Sigma_{\mathsf{z}}(1:r, 1:r)$ est aussi définie-positive. A la lumière de ces précisions, il est facile de remarquer que Δ_r est non singulière tant que $r \leq n$ mais devient singulière dès que $r > n$. Ainsi, l'ordre doit satisfaire à

$$n = \max\left\{r : \text{rang}\left(\Delta_r\right) = r\right\}.$$

Ces arguments justifient l'algorithme décrit ci-dessous pour l'identification de l'ordre.

Considérons $\Delta_r = \bar{\Sigma}(1:r, 1:r)$ (car en référence à l'équation (3.36), $\bar{\Sigma} = \Sigma$ en absence de bruit, c'est-à-dire lorsque $\sigma_v^2 = 0$) et écrivons

$$\Delta_{r+1} = \bar{\Sigma}(1:r+1, 1:r+1) = \begin{bmatrix} \Delta_r & w_{r+1} \\ w_{r+1}^\top & s_{r+1} \end{bmatrix},$$

avec $w_{r+1} = \bar{\Sigma}(:, r+1)$ et $s_{r+1} = \bar{\Sigma}(r+1, r+1)$. Par le lemme d'inversion de matrices blocs [70], si Δ_r

est non singulière, alors Δ_{r+1} est aussi non singulière si et seulement si

$$h_{r+1} = s_{r+1} - w_{r+1}^\top \Delta_r^{-1} w_{r+1} \tag{3.39}$$

est non nul. La raison de cette assertion est que Δ_{r+1}^{-1} est donnée par

$$\Delta_{r+1}^{-1} = h_{r+1}^{-1} \begin{bmatrix} h_{r+1}\Delta_r^{-1} + \varphi_{r+1}\varphi_{r+1}^\top & \varphi_{r+1} \\ \varphi_{r+1}^\top & 1 \end{bmatrix}, \tag{3.40}$$

où $\varphi_{r+1} = -\Delta_r^{-1} w_{r+1} \in \mathbb{R}^r$. Cela suggère que pour identifier l'ordre, une solution pourrait consister à calculer récursivement l'inverse de la sous-matrice Δ_r extraite de Σ en partant de $r = r_{\min}$ vers $r = r_{\max}$ jusqu'à ce que l'ordre soit détecté. Si $\Delta_{r_{\min}}^{-1}$ est connu, avec $r_{\min} \geq 1$, la valeur initiale de r peut être prise égale à $r = r_{\min}$, sinon on peut commencer à $r = 1$. On procède ensuite au calcul de h_{r+1} : si $h_{r+1} = 0$ pour un certain $r = r^*$, alors Δ_{r^*+1} est singulière d'où on conclut que $n = r^*$; par contre, si $h_{r+1} \neq 0$, on calcule Δ_{r+1}^{-1} puis h_{r+2} et ainsi de suite. La procédure s'arrête quand Δ_{r+1} devient singulière, c'est-à-dire, quand $h_{r+1} = 0$. Pour voir que $h_{n+1} = 0$, reportons nous à (3.36) pour constater que quand $r = n$, on a $w_{n+1} = \Delta_n \mathscr{P}(1,:)^\top$ et $s_{n+1} = \mathscr{P}(1,:)\Delta_n \mathscr{P}(1,:)^\top$. Alors, par définition de h_{r+1} en (3.39) il vient

$$\begin{aligned} h_{n+1} &= s_{n+1} - w_{n+1}^\top \Delta_n^{-1} w_{n+1} \\ &= \mathscr{P}_f(1,:)\Delta_n \mathscr{P}_f(1,:)^\top - \mathscr{P}_f(1,:)\Delta_n \Delta_n^{-1} \Delta_n \mathscr{P}_f(1,:)^\top = 0. \end{aligned}$$

A l'issue de notre procédure de recherche de l'ordre, on obtient en même temps que l'ordre, $\Sigma_{11}^{-1} = \Sigma_{\mathbf{z}}^{-1} = \Delta_n^{-1}$ et donc, $\mathscr{P}_f$ peut finalement être estimé grâce à la formule (3.38).

3.4.3.2 Cas stochastique

En présence de bruit dans les données, la variance σ_v^2 du bruit dans l'équation (3.36) n'est plus nulle. Par conséquent, contrairement au cas déterministe, h_{n+1} sera très probablement différent de zéro. Mais la méthode décrite précédemment peut toujours s'appliquer à condition d'introduire un seuillage adéquat. Naturellement, le seuil de détection de l'ordre va dépendre du niveau du bruit agissant sur le système. De plus, ce seuil doit être relié d'une certaine manière au système à identifier et donc il a besoin être adapté, particulièrement dans le cas des systèmes à commutations (cf. Chapitre 4).

La présence du bruit dans les mesures tend à amplifier les quantités h_r mais un « *saut* » devrait toujours être observable dans la séquence des valeurs de $h_{r_{\min}+1}, \cdots, h_n, h_{n+1}$ à moins que le bruit ne soit dominant par rapport au signal. Puisque $\Sigma_{\mathbf{z}}$ est définie positive, le théorème du complément de Schur [57, p. 472] (le paramètre h_{r+1} est en fait le complément de Schur de Δ_r dans Δ_{r+1}), permet de savoir que h_r est un scalaire strictement positif pour $r \leq n$. Par définition de h_{r+1} en (3.39), on a

$$h_{r+1} = s_{r+1} + \sigma_v^2 - w_{r+1}^\top \Delta_r^{-1} \left(I_r + \sigma_v^2 \Delta_r^{-1}\right)^{-1} w_{r+1}.$$

Si nous considérons que les valeurs propres de Δ_r sont nettement supérieures à la variance σ_v^2 du bruit, alors le rayon spectral de $\sigma_v^2\Delta_r^{-1}$ est inférieur à l'unité. Ainsi, en développant le terme entre parenthèses au premier ordre, on obtient l'approximation suivante

$$h_{r+1} \simeq \left(s_{r+1} - w_{r+1}^\top \Delta_r^{-1} w_{r+1}\right) + \sigma_v^2 \left(1 + w_{r+1}^\top \Delta_r^{-2} w_{r+1}\right),$$

qui est composé d'une première partie due au signal utile et d'une seconde partie liée au bruit indésirable. Comme dans le cas précédent, quand $r = n$, le premier terme s'annule si bien que $h_{n+1} \simeq \sigma_v^2\left(1 + \varphi_{n+1}^\top \varphi_{n+1}\right)$, où φ_r est défini comme dans l'équation (3.40). En raison de cette approximation, un seuil de test peut être sélectionné comme

$$\text{Thres}(r) = T_0 \left(1 + \varphi_{r+1}^\top \varphi_{r+1}\right), \tag{3.41}$$

où T_0 est une constante supposée légèrement supérieure à la variance σ_v^2 du bruit. L'ordre est alors déterminé par $n = \min\{r : h_{r+1} < \text{Thres}(r)\}$. Notons qu'après détection de l'ordre, on peut approximer *a posteriori* la variance du bruit de la façon suivante :

$$\hat{\sigma}_v^2 \approx \frac{h_{n+1}}{1 + \varphi_{n+1}^\top \varphi_{n+1}}.$$

3.4.4 Implémentation récursive

Dans les sous-sections précédentes, nous avons présenté un schéma complet hors ligne pour l'identification d'un modèle du type (3.21). Dans cette sous-section, nous allons développer une version récursive de ce schéma d'identification en vue d'une application future à l'identification de systèmes commutants (cf. Chapitre 4).

La version en ligne de notre algorithme s'appuie sur une adaptation récursive de la matrice Σ définie par (3.36). A chaque instant, Σ est d'abord mis à jour puis la procédure décrite ci-dessus est utilisée pour l'extraction simultanée de l'ordre et de la matrice d'observabilité étendue.

Supposons qu'une factorisation RQ de la matrice de données entrée-sortie (comme définie par la formule (3.32)) soit connue à l'instant $t = N + f - 1$. On aimerait alors adapter la partie triangulaire R de cette factorisation, chaque fois qu'une nouvelle colonne de mesures est ajoutée à la matrice $\begin{bmatrix} U^\top & \bar{Y}^\top \end{bmatrix}^\top$. Ainsi, à l'instant $t+1$, on peut écrire

$$\underbrace{\begin{bmatrix} \sqrt{\lambda}\begin{bmatrix} R_{11}(t) & 0 \\ R_{21}(t) & R_{22}(t) \end{bmatrix} & \begin{matrix} u_f(\bar{t}+1) \\ \bar{y}_f(\bar{t}+1) \end{matrix} \end{bmatrix}}_{\mathscr{R}} \begin{bmatrix} Q_1(t) & 0 \\ Q_2(t) & 0 \\ 0 & 1 \end{bmatrix},$$

où $\bar{t} = t - f + 1$ et $\lambda < 1$ est un facteur d'oubli. Pour retrouver la forme triangulaire de la partie R, nous adoptons la méthode des rotations de Givens [76], [49]. Cette méthode consiste à construire une séquence appropriée de matrices de rotations de Givens dont le produit noté ici $\mathbf{G}_v(t+1)$, est tel qu'en multipliant

$\mathscr{R}$ à droite, il annule les fn_u premiers éléments de la dernière colonne de $\mathscr{R}$. Ainsi, en appliquant cette méthode, on obtient

$$\begin{bmatrix} \sqrt{\lambda}\begin{bmatrix} R_{11}(t) & 0 \\ R_{21}(t) & R_{22}(t) \end{bmatrix} & \begin{matrix} u_f(\bar{t}+1) \\ \bar{y}_f(\bar{t}+1) \end{matrix} \end{bmatrix} \mathbf{G}_v(t+1) \\ = \begin{bmatrix} R_{11}(t+1) & 0 & 0 \\ R_{21}(t+1) & \sqrt{\lambda}R_{22}(t) & z_f(t+1) \end{bmatrix}. \tag{3.42}$$

Il apparaît alors que

$$R_{22}(t+1) = \begin{bmatrix} \sqrt{\lambda}R_{22}(t) & z_f(t+1) \end{bmatrix},$$

d'où

$$SR_{22}(t+1)R_{22}^{\top}(t+1)S^{\top} = \lambda SR_{22}(t)R_{22}^{\top}(t)S^{\top} + Sz_f(t+1)z_f^{\top}(t+1)S^{\top}. \tag{3.43}$$

De la formule (3.33), la matrice Σ définie en (3.36) s'apparente à une sorte de matrice de covariance. Dans un contexte récursif avec une pondération exponentiellement décroissante des données, $\Sigma(t+1)$ peut être obtenu en divisant la quantité

$$SR_{22}(t+1)R_{22}^{\top}(t+1)S^{\top} = \sum_{k=0}^{t+1} \lambda^{t+1-k} Sz_f(k)z_f^{\top}(k)S^{\top}$$

par la somme des poids, c'est-à-dire $\alpha(t+1) = 1 + \lambda + \cdots + \lambda^{t+1} = \frac{1-\lambda^{t+2}}{1-\lambda}$. Cependant, on peut noter que dans le présent cas, cette opération de division n'est pas nécessaire puisque ce qui importe pour l'estimation de $\mathscr{P}_f$, c'est la structure de la matrice Σ. Comme il n' y a aucun risque de divergence pour $\lambda < 1$, nous pouvons simplement poser dans la suite

$$\Sigma(t+1) = SR_{22}(t+1)R_{22}^{\top}(t+1)S^{\top}. \tag{3.44}$$

Comme nous l'avons déjà mentionné, il n'est pas nécessaire d'adapter entièrement la matrice $\Sigma(t)$. En fait, la mise à jour de ses $r_{\max}$ premières lignes ou colonnes suffira à l'identification du modèle du système. Après avoir mis à jour $\Sigma(t)$, il ne reste plus qu'à appliquer la procédure d'identification décrite dans les sous-sections 3.4.2 et 3.4.3. Pour ce faire, il convient (dans le cas où $r_{\min} > 1$) de calculer récursivement l'inverse $\Delta_{r_{\min}}^{-1}(t)$ de la sous-matrice $\Delta_{r_{\min}}(t)$ de $\Sigma(t)$, ce que nous faisons en appliquant le lemme d'inversion matricielle ou formule de Sherman-Morrison [83]. En effet en posant $\xi(t) = \begin{bmatrix} I_{r_{\min}} & 0 \\ 0 & 0 \end{bmatrix} Sz_f(t)$, on a

$$\Delta_{r_{\min}}(t+1) = \lambda\Delta_{r_{\min}}(t) + \xi(t)\xi^{\top}(t),$$

d'où

$$\Delta_{r_{\min}}^{-1}(t+1) = \lambda^{-1}\Delta_{r_{\min}}^{-1}(t) - \frac{\lambda^{-1}\Delta_{r_{\min}}^{-1}(t)\xi(t)\xi^{\top}(t)\Delta_{r_{\min}}^{-1}(t)}{\lambda + \xi^{\top}(t)\Delta_{r_{\min}}^{-1}(t)\xi(t)}. \tag{3.45}$$

Connaissant $\Delta_{r_{\min}}^{-1}$, on peut maintenant procéder à la recherche de l'ordre puis à l'estimation des paramètres comme résumé dans l'algorithme 3.1.

Algorithme 3.1 : Identification en ligne de sous-espaces

Initialisation : choisir λ, T_0, $r_{\max} = f$ et donner des valeurs à $\Sigma(0)$ et $\Delta_{r_{\min}}^{-1}(0)$
Pour $t = 1, \ldots, \infty$

1. Mettre à jour la factorisation RQ de la matrice de données comme dans la sous-section 3.4.4
2. Mettre à jour $\Sigma(t)$ par la formule (3.44)
3. Mettre à jour $\Delta_{r_{\min}}^{-1}(t)$ par la formule (3.45)
4. Estimer l'ordre
   - $r \leftarrow r_{\min}$
   - **Tant que** $h_{r+1} \geq \text{Thres}(r)$ et $r < r_{\max}$
     Calculer $\Delta_{r+1}^{-1}(t)$ en utilisant (3.40)
     $r \leftarrow r + 1$
     Calculer h_{r+1} et Thres(r)
     Fin Tant que
   - $n \leftarrow r$
5. Une fois l'ordre connu, calculer $\mathscr{P}_f(t)$ par la formule (3.38)
6. En déduire les matrices du système comme dans la sous-section 3.4.1

FinPour

3.5 Identification de modèles d'état canoniques

Comme déjà mentionné dans la partie précédente, une caractéristique des méthodes des sous-espaces est qu'elles fournissent une réalisation d'état du système considéré dans une base *arbitraire*. Le terme arbitraire qualifie ici le fait qu'on ne peut pas prévoir la base de représentation dans l'espace d'état (même si on peut la modifier) et que deux jeux différents de données aboutissent à des réalisations d'état dans des bases potentiellement différentes. Cela devient un inconvénient quand il s'agit par exemple d'estimer un modèle composite c'est-à-dire un modèle constitué de plusieurs sous-modèles linéaires. Car si les matrices des sous-modèles sont relatives à des bases différentes de l'espace d'état, le comportement entrée-sortie du système pourrait s'en trouver altéré.

Dans cette section, nous présentons une autre alternative aux méthodes des sous-espaces existantes. Celle-ci repose sur l'estimation directe de la relation entrée-sortie suivie d'une idée nouvelle d'obtention d'une réalisation d'état canonique pour une classe de systèmes MIMO. Cette approche est notamment motivée par de récentes études [109] selon lesquelles l'identification directe d'un transfert entrée-sortie (E-S) pourrait être plus attractive en termes de complexité numérique et de précision que la plupart des méthodes des sous-espaces (S-S).

Cela peut se justifier de la façon suivante : l'approche S-S possède beaucoup d'étapes parce qu'elle procède d'abord à l'estimation de structures surparamétrées (comme les matrices Γ_f et H_f construites

sur un horizon f dans le cas des méthodes MOESP ou souvent $f+p$ pour la classe des méthodes N4SID, avec $f, p > n$) avant d'en déduire les matrices du système par réduction de modèle (à travers une DVS) [98]. De plus, les performances de ces méthodes pourraient, comme cela a été mis en évidence dans [59], dépendre des paramètres f et p. Ainsi, l'approche E-S peut, dans bien des cas, être plus parcimonieuse et plus directe (en terme de nombre d'étapes) que les méthodes S-S [109].

Nous présentons ici une autre méthode somme toute assez semblable à la méthode discutée dans la section 3.4 de ce chapitre pour l'identification d'un système du type (3.1). Bien que notre objectif soit d'obtenir directement un modèle d'état à partir des mesures entrée-sortie, nous pouvons considérer une structure entrée-sortie dérivée des équations du système (3.1) à l'image de (3.6), mais à la différence que l'état n'apparaît plus explicitement.

3.5.1 Obtention d'un modèle entrée-sortie ARMAX

Pour des raisons de lisibilité, nous commençons par rappeler la forme du modèle (3.1) :

$$\left\{\begin{array}{rcl} x(t+1) & = & Ax(t)+Bu(t)+w(t) \\ y(t) & = & Cx(t)+Du(t)+v(t), \end{array}\right. \tag{3.46}$$

où $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^{n_u}$, $y(t) \in \mathbb{R}^{n_y}$ sont respectivement, les vecteurs d'état, d'entrée et de sortie. Les vecteurs $w(t) \in \mathbb{R}^n$ et $v(t) \in \mathbb{R}^{n_y}$ symbolisent comme précédemment les bruits de système et de mesure qui sont supposés blancs.

Au lieu d'exprimer la sortie sur une fenêtre f comme dans (3.4), considérons cette fois, pour l'estimation d'un modèle du type (3.1), une expression directe de la sortie $y(t)$ en fonction du passé du système. Pour $t > n$, on peut, par substitutions successives dans la première équation de (3.46), écrire

$$x(t) = A^n x(t-n) + \Delta_n u_n(t-n) + \mathscr{A}_n w_n(t-n),$$

où

$$\begin{array}{l} \Delta_n = \left[A^{n-1}B \quad \cdots \quad AB \quad B\right] \in \mathbb{R}^{n\times nn_u}, \\ \mathscr{A}_n = \left[A^{n-1} \quad \cdots \quad A \quad I_n\right] \in \mathbb{R}^{n\times n^2}. \end{array} \tag{3.47}$$

En portant l'expression de $x(t)$ dans la seconde équation de (3.46), on a

$$\begin{array}{rl} y(t) & = Cx(t)+Du(t)+v(t) \\ & = CA^n x(t-n) + C\Delta_n u_n(t-n) + C\mathscr{A}_n w_n(t-n) + Du(t) + v(t) \end{array} \tag{3.48}$$

Par ailleurs, le théorème de Cayley-Hamilton permet d'obtenir

$$A^n = -a_0 I_n - a_1 A - \cdots - a_{n-1}A^{n-1},$$

où les a_j, $j = 0, \cdots, n$ sont les coefficients du polynôme caractéristique de A, $p_A(z) = \det(zI - A) = a_0 + a_1 z + \cdots + a_{n-1}z^{n-1} + z^n$. Notons $a = \begin{bmatrix} a_0 & \cdots & a_{n-1} \end{bmatrix}^\top \in \mathbb{R}^n$ le vecteur de coefficients de $p_A(z)$. Ainsi,

$$\begin{aligned} CA^n x(t-n) &= -(a_0 C + a_1 CA + \cdots + a_{n-1}CA^{n-1})x(t-n) \\ &= -(a^\top \otimes I_{n_y})\Gamma_n x(t-n) \end{aligned} \tag{3.49}$$

où le symbole $\otimes$ fait référence au produit de Kronecker. D'autre part nous pouvons, d'une manière similaire à (3.4), écrire

$$\Gamma_n x(t-n) = y_n(t-n) - H_n u_n(t-n) - G_n w_n(t-n) - v_n(t-n), \tag{3.50}$$

En combinant maintenant (3.48), (3.49) et (3.50), il vient

$$\begin{aligned} y(t) = &-(a^\top \otimes I_{n_y})y_n(t-n) + \Big((a^\top \otimes I_{n_y})H_n + C\Delta_n\Big)u_n(t-n) + Du(t) \\ &+ \Big((a^\top \otimes I_{n_y})G_n + C\mathscr{A}_n\Big)w_n(t-n) + (a^\top \otimes I_{n_y})v_n(t-n) + v(t). \end{aligned} \tag{3.51}$$

Si dans l'expression de $y(t)$ ci-dessus, on note

$$\eta(t) = \Big((a^\top \otimes I_{n_y})G_n + C\mathscr{A}_n\Big)w_n(t-n) + (a^\top \otimes I_{n_y})v_n(t-n) + v(t), \tag{3.52}$$

la somme des termes relatifs au bruit, alors il vient :

$$y(t) = -(a^\top \otimes I_{n_y})y_n(t-n) + Fu_{n+1}(t-n) + \eta(t), \tag{3.53}$$

avec

$$F = \begin{bmatrix} (a^\top \otimes I_{n_y})H_n + C\Delta_n & D \end{bmatrix} \in \mathbb{R}^{n_y \times (n+1)n_u},$$

En terme de fonction de transfert rationnelle, F est la matrice formée des coefficients des différents polynômes numérateurs tandis le vecteur a encode les coefficients du dénominateur commun. Un avantage de l'écriture du modèle ARMAX (3.51) est qu'elle permet de mettre en évidence la structure du modèle du bruit (3.52). On peut voir que ce modèle de bruit est constitué d'une certaine recombinaison des paramètres de la partie déterministe du système, de sorte qu'il suffit de connaître les matrices de (3.1) pour disposer aussi du modèle stochastique (3.52).

En définissant maintenant

$$\Phi_n^y(t-n) = \begin{bmatrix} y(t-n) & \cdots & y(t-1) \end{bmatrix} \in \mathbb{R}^{n_y \times n},$$

l'on aboutit, en faisant usage de l'identité matricielle $\text{vec}(AXB) = (B^\top \otimes A)\text{vec}(X)$ [26], au modèle de régression

$$y(t) = \begin{bmatrix} -\Phi_n^y(t-n) & u_{n+1}^\top(t-n) \otimes I_{n_y} \end{bmatrix} \begin{bmatrix} a \\ \text{vec}(F) \end{bmatrix} + \eta(t), \tag{3.54}$$

où $\text{vec}(\cdot)$ est l'opérateur de vectorisation. Ainsi, a et F peuvent être obtenus par les méthodes classiques de régression linéaire (cf. [74]). Par exemple, dans le cas où le bruit $\{\eta(t)\}$ est blanc, ces paramètres peuvent être estimés par

$$\begin{bmatrix} \hat{a} \\ \text{vec}(\hat{F}_n) \end{bmatrix} = \underset{a,\text{vec}(F_n)}{\arg\min} \left\| \mathcal{Y} - \Phi_n \begin{bmatrix} a \\ \text{vec}(F_n) \end{bmatrix} \right\|_2^2, \tag{3.55}$$

avec

$$\mathcal{Y} = \begin{bmatrix} y(n+1) \\ \vdots \\ y(n+N) \end{bmatrix}, \qquad \Phi_n = \begin{bmatrix} -\Phi_n^y(1) & u_{n+1}^\top(1) \otimes I_{n_y} \\ \vdots & \vdots \\ -\Phi_n^y(N) & u_{n+1}^\top(N) \otimes I_{n_y} \end{bmatrix}.$$

Mais lorsque le bruit $\{\eta(t)\}$ est coloré, il est bien connu que la solution donnée par (3.55) est biaisée. Il convient alors d'introduire une variable instrumentale dans le critère à minimiser afin d'atténuer l'effet du bruit.

3.5.2 Obtention d'une réalisation d'état

Au regard de l'équation (3.54), on pourrait penser que le modèle minimal S-S (3.1) et le modèle E-S (3.54) qui en résulte, ont le même ordre n. Malheureusement, cela n'est pas vrai (cf. [109] pour un contrexemple) pour tous les types de systèmes MIMO. Ces situations dégénérées se produisent de façon générale quand la matrice de dynamique A est dérogatoire. Mais de façon générale on a $n_{ES} \leq n_{SS}$, où n_{ES} et n_{SS} désignent respectivement les ordres des modèles (3.54) et (3.1). Même dans le cas où $n_{ES} \neq n_{SS}$, l'ordre réel n_{ES} de (3.54) pourrait être déterminé à travers une condition de rang portant sur la matrice de données impliquée dans (3.1). De cette façon, le système (3.1) pourra, sans perte de généralité, être représenté par le modèle E-S (3.54).

Pour pouvoir passer facilement du modèle intermédiaire (3.54) au modèle d'état recherché (3.1), il nous faut connaître la relation qui lie n_{ES} à n_{SS}. Dans ce but, nous nous intéressons à l'établissement de conditions nécessaires et suffisantes pour que les modèles S-S (3.1) et E-S (3.54) (qui représentent le même système) aient le même ordre n. Nous obtenons alors le résultat suivant.

Theorème 3.3. *Supposons que le modèle (3.1) soit minimal, c'est-à-dire observable et atteignable. Alors l'ordre n_{ES} du modèle E-S (3.53) est égal à l'ordre n_{SS} du modèle S-S (3.1) si et seulement si la matrice A est non dérogatoire.*

Preuve. Une preuve de ce théorème est fournie dans l'annexe A.3. ■

Dans la suite, nous allons présenter une méthode qui permet de passer du modèle E-S (3.54) à un modèle d'état. Pour cela (pour une raison qui va apparaître plus clairement dans le paragraphe suivant), nous ferons l'hypothèse que A est non dérogatoire. Il en résulte, selon le lemme 3.2, qu'il existe au moins une combinaison $\gamma^\top y(t)$ des sorties à partir de laquelle on peut observer la totalité des dynamiques du système. En d'autres termes, il existe au moins un $\gamma \in \mathbb{R}^{n_y}$ tel que

$$\begin{bmatrix}(\gamma^\top C)^\top & (\gamma^\top CA)^\top & \cdots & (\gamma^\top CA^{f-1})^\top\end{bmatrix}^\top$$

soit de rang plein n pour tout $f \geq n$.

Proposition 3.4. *Supposons que le modèle (3.1) soit minimal et que la matrice A soit non dérogatoire. Alors, il existe une matrice $\Psi \in \mathbb{R}^{n\times nn_y}$ de la forme*

$$\Psi = \begin{bmatrix}\gamma^\top & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \gamma^\top\end{bmatrix} \in \mathbb{R}^{n\times nn_y}, \tag{3.56}$$

avec $\gamma \in \mathbb{R}^{n_y}$, telle que

$$\operatorname{rang}(\Psi\Gamma_n) = n, \tag{3.57}$$

où Γ_n est la matrice d'observabilité de (3.1).

Si Ψ vérifie la condition de rang (3.57), alors une réalisation minimale du système (3.1) peut être obtenue par :

$$A = \begin{bmatrix}0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & 0\\ 0 & 0 & \cdots & 1\\ -a_0 & -a_1 & \cdots & -a_{n-1}\end{bmatrix}, \tag{3.58}$$

$$D = F(:, nn_u + 1 : (n+1)n_u), \tag{3.59}$$

$$B = \Psi Q, \tag{3.60}$$

$$C = \operatorname{Row}_{n_y}(Q)\Omega_n^\top\left(\Omega_n\Omega_n^\top\right)^{-1}, \tag{3.61}$$

où

$$Q = \left(\mathcal{K} \otimes I_{n_y}\right)^{-1} \mathrm{Col}_{n_u}\left(F(:, 1 : nn_u) - a^\top \otimes D\right)$$
$$\Omega_n = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \in \mathbb{R}^{n \times nn_u},$$
$$\mathcal{K} = \begin{bmatrix} a_1 & \cdots & a_{n-1} & 1 \\ \vdots & \cdots & \vdots & \vdots \\ a_{n-1} & \cdots & 0 & 0 \\ 1 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n \times n},$$

$\mathrm{Row}_{n_y}(\cdot)$ *et* $\mathrm{Col}_{n_u}(\cdot)$ *sont des opérateurs de bloc-transposition du type*

$$\mathrm{Row}_{n_y}\left(\begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}\right) = \begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix},$$
$$\mathrm{Col}_{n_u}\left(\begin{bmatrix} Y_1 & \cdots & Y_n \end{bmatrix}\right) = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix},$$

où X_i *contient* n_y *lignes et* Y_i *contient* n_u *colonnes.*

Preuve. L'existence d'une matrice Ψ de la forme (3.56) vérifiant (3.57) est une conséquence directe du lemme 3.2.

Le reste de la preuve de la proposition consiste essentiellement à calculer une réalisation canonique du système (3.1) à partir de a et F. De la définition de F en (3.53), on a :

$$F(:, 1 : nn_u) = (a^\top \otimes I_{n_y})H_n + C\Delta_n,$$

avec Δ_n défini comme en (3.47) et H_n défini comme dans la section 3.2.2. Des développements algébriques simples montrent alors que

$$F(:, 1 : nn_u) - a^\top \otimes D =$$
$$\begin{bmatrix} (k_1^\top \otimes I_{n_y})\Gamma_n B, & \cdots, & (k_n^\top \otimes I_{n_y})\Gamma_n B \end{bmatrix},$$

où $k_i^\top$ désigne la i-ème ligne de $\mathcal{K}$. Il en résulte que les n premiers paramètres de Markov sont donnés par

$$\Gamma_n B = \left(\mathcal{K} \otimes I_{n_y}\right)^{-1} \mathrm{Col}_{n_u}\left(F(:, 1 : nn_u) - a^\top \otimes D\right) \tag{3.62}$$

car $(\mathcal{K} \otimes I_{n_y})$ est clairement inversible. Par ailleurs, il est bien connu que si (A, B, C, D) est une réalisation minimale du système (3.1), alors pour toute matrice carrée non-singulière $T \in \mathbb{R}^{n \times n}$, $(TAT^{-1}, TB, CT^{-1}, D)$ en est aussi une réalisation d'état. Posons donc $T = \Psi\Gamma_n$. Alors, TB est donnée par (3.60).

Pour trouver la matrice d'état, il nous suffit d'évaluer

$$\begin{aligned} TAT^{-1} &= \Psi\left(\Gamma_n A\right)\left(\Psi\Gamma_n\right)^{-1} \\ &= \Psi \begin{bmatrix} CA \\ \vdots \\ CA^n \end{bmatrix} (\Psi\Gamma_n)^{-1} \\ &= \Psi \begin{bmatrix} 0_{n_y\times n_y} & I_{n_y} & \cdots & 0_{n_y\times n_y} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{n_y\times n_y} & 0_{n_y\times n_y} & \cdots & I_{n_y} \\ -a_1 I_{n_y} & -a_2 I_{n_y} & \cdots & -a_n I_{n_y} \end{bmatrix} \Gamma_n(\Psi\Gamma_n)^{-1} \\ &= \Psi(A_c \otimes I_{n_y})\Gamma_n(\Psi\Gamma_n)^{-1} \end{aligned}$$

où A_c est la matrice définie par (3.58). Remarquons que la forme particulière (3.56) de Ψ permet d'avoir

$$\Psi(A_c \otimes I_{n_y}) = A_c\Psi. \tag{3.63}$$

Ainsi, $TAT^{-1} = A_c$. Disposant maintenant de TB et TAT^{-1}, la matrice CT^{-1} (Eq. (3.61)) se déduit facilement de (3.62).■

Il faut noter que la forme particulière (3.56) de Ψ ne tient qu'à la possibilité de pouvoir obtenir une propriété du type (3.63). Sans cette constrainte, Ψ aurait pu être choisie de forme quelconque.
Contrairement aux techniques traditionnelles [69] qui requièrent généralement une connaissance des indices d'observabilité et une étape intermédiaire de construction de modèles non minimaux suivie d'une application de routines de réduction de modèle, la proposition 3.4 fournit une méthode simple pour obtenir la représentation d'état d'un système MIMO à partir de ses équations aux différences.

3.6 Identification structurée des sous-espaces

Nous avons vu que l'application des méthodes discutées plus tôt dans ce chapitre pour l'identification de modèles d'état, requiert que la matrice de dynamique A du système considéré soit non dérogatoire. Cependant, cette hypothèse peut paraître assez restrictive. Afin de pallier cette faiblesse, nous présentons dans cette section une nouvelle méthode plus générique que les deux précédentes méthodes, pour l'identification de modèles d'état du type (3.1). Contrairement à la plupart des méthodes existantes, notre méthode ne requiert aucune DVS. Ainsi, son extension à l'identification récursive est très naturelle et ne nécessite pas de dévéloppement supplémentaire contrairement aux méthodes récursives des sous-espaces qui sont connues dans la littérature [76, 82, 89]. De plus, elle permet d'avoir un contrôle simple de la base dans laquelle les matrices du système seront obtenues.

Pour commencer, supposons l'ordre n du système (3.1) connu et rappelons l'équation de données (3.7)

$$Y = \Gamma_f X + H_f U + E, \tag{3.64}$$

où nous supposons $f \geq n+1$. Considérons une matrice quelconque $\Lambda_f \in \mathbb{R}^{n \times f n_y}$ vérifiant [1]

$$\text{rang}(\Lambda_f \Gamma_f) = n. \tag{3.65}$$

Si on multiplie l'équation (3.64) à gauche par Λ_f, il vient

$$\Lambda_f Y = \Lambda_f \Gamma_f X + \Lambda_f H_f U + \Lambda_f E. \tag{3.66}$$

Comme $T = \Lambda_f \Gamma_f$ est une matrice carrée inversible, on réalise un changement de base d'état en posant $X \leftarrow \bar{X} = TX$. L'équation précédente donne alors la séquence d'état dans la nouvelle base, soit

$$\bar{X} = \Lambda_f Y - \Lambda_f H_f U - \Lambda_f E. \tag{3.67}$$

Suite à ce changement de coordonnées, les matrices (A, B, C, D) deviennent

$$(\bar{A}, \bar{B}, \bar{C}, \bar{D}) = (TAT^{-1}, TB, CT^{-1}, D)$$

et Γ_f devient $\bar{\Gamma}_f = \Gamma_f T^{-1}$ mais H_f reste inchangée puisqu'elle ne dépend pas de la base de coordonnées. Pour des raisons de clarté, nous traiterons d'abord le cas où les données ne sont pas affectées par du bruit, c'est-à-dire celui où E est identiquement nulle. Puis, nous montrerons que la méthode peut s'appliquer aussi à des données corrompues par du bruit.

3.6.1 Cas déterministe : première méthode

Commençons par noter qu'en absence de bruit la relation (3.67) équivaut à

$$\bar{X} = \Lambda_f Y - \Lambda_f H_f U. \tag{3.68}$$

De cette relation, il apparaît clairement que le choix de Λ_f détermine complètement la base de l'état puisqu'elle définit directement l'état comme une combinaison linéaire des données entrée-sortie.
Pour obtenir $\bar{\Gamma}_f$, on peut reporter l'expression (3.68) de l'état dans (3.64). On obtient

$$\begin{aligned} Y &= \Gamma_f T^{-1} T X + H_f U \\ &= \bar{\Gamma}_f \bar{X} + H_f U \\ &= \bar{\Gamma}_f \Lambda_f Y + (I_{f n_y} - \bar{\Gamma}_f \Lambda_f) H_f U \\ &= \begin{bmatrix} \bar{\Gamma}_f & (I_{f n_y} - \bar{\Gamma}_f \Lambda_f) H_f \end{bmatrix} \begin{bmatrix} \Lambda_f Y \\ U \end{bmatrix}. \end{aligned} \tag{3.69}$$

Cela montre que le problème d'identification des sous-espaces peut être transformé en un problème de moindres carrés ordinaires, grâce à une élimination de l'état qui est inconnu. Selon les Propositions 3.1

1. La détermination d'une telle matrice Λ_f est reportée à la sous-section 3.6.4.

et 3.2, quand le processus d'entrée est suffisamment excitant d'ordre au moins $n+f$, et le modèle (3.1) est minimal, on a

$$\text{rang}\left(\begin{bmatrix} Y \\ U \end{bmatrix}\right) = n + fn_u.$$

En menant un raisonnement similaire à celui utilisé dans la preuve de la proposition 3.2 (voir Eq. (3.11)), on peut facilement établir les relations

$$\text{rang}\left(\begin{bmatrix} \Lambda_f Y \\ U \end{bmatrix}\right) = n + fn_u \quad \text{et} \quad \text{rang}(\Lambda_f Y \Pi_U^{\perp}) = n,$$

ce qui implique qu'on peut extraire directement et de manière consistante la matrice de paramètres $\begin{bmatrix} \bar{\Gamma}_f & (I_{fn_y} - \bar{\Gamma}_f \Lambda_f) H_f \end{bmatrix}$ de l'équation (3.69). Cependant, remarquons qu'il n'est pas nécessairement intéressant d'estimer le terme $(I_{fn_y} - \bar{\Gamma}_f \Lambda_f) H_f$ à partir de (3.69) car la matrice $(I_{fn_y} - \bar{\Gamma}_f \Lambda_f)$ n'étant pas de rang plein (cf. Remarque 3.4), cela ne permet pas de calculer H_f. Ainsi, nous allons commencer par éliminer ce terme de (3.69) en multipliant toute l'équation à droite par $\Pi_U^{\perp}$ avant de calculer $\bar{\Gamma}_f$. On obtient

$$\begin{aligned} \bar{\Gamma}_f &= Y \Pi_U^{\perp} (\Lambda_f Y \Pi_U^{\perp})^{\dagger}, \\ &= \Sigma_{yu} \Lambda_f^{\top} \left(\Lambda_f \Sigma_{yu} \Lambda_f^{\top} \right)^{-1}, \end{aligned} \tag{3.70}$$

où $\Sigma_{yu} = \frac{1}{N} Y \Pi_U^{\perp} Y^{\top}$. Ici, Σ_{yu} est une matrice de corrélation tandis que $\Pi_U^{\perp}$ définie en (3.9) est une matrice de projection sur les directions orthogonales aux lignes de U. Le symbole $\dagger$ fait référence à l'inverse de Moore-Penrose. En pratique, la projection $\Pi_U^{\perp}$ peut être implémentée en recourant à des méthodes numériques réputées robustes comme la décomposition QR [122], [72]. Une fois que $\bar{\Gamma}_f$ est disponible, l'extraction des matrices $\bar{A}$ et $\bar{C}$, s'opère très simplement par la proprieté d'A-invariance de $\bar{\Gamma}_f$ comme en (3.18). Quant aux matrices $\bar{B}$ et $\bar{D}$ correspondantes, elles peuvent être estimées par exemple par régression linéaire [72], [76].

3.6.2 Cas déterministe : deuxième méthode

L'objectif ici est de récupérer directement les matrices du système en évitant l'étape intermédiaire qui consiste à calculer d'abord la matrice d'observabilité étendue $\bar{\Gamma}_f$. Cette démarche pourrait avoir un certain intérêt dans le cadre d'une implémentation récursive par exemple. A notre connaissance, il n'existe aucun algorithme récursive des sous-espaces qui permette de mettre à jour directement la matrice de dynamique A.

Pour ce faire, remarquons que les équations (3.1) du système, écrites sur un horizon $[1,\ N]$ donnent

$$\begin{bmatrix} \bar{X}^1 \\ \mathcal{Y} \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{B} \\ \bar{C} & \bar{D} \end{bmatrix} \begin{bmatrix} \bar{X}^o \\ \mathcal{U} \end{bmatrix}, \tag{3.71}$$

avec $\bar{X}^1 = \bar{X}(:, 2:N)$, $\bar{X}^o = \bar{X}(:, 1:N-1)$ et

$$\begin{aligned}\mathcal{Y} &= \begin{bmatrix} y(f+1) & \cdots & y(f+N-1) \end{bmatrix} \in \mathbb{R}^{n_y \times (N-1)},\\ \mathcal{U} &= \begin{bmatrix} u(f+1) & \cdots & u(f+N-1) \end{bmatrix} \in \mathbb{R}^{n_u \times (N-1)}.\end{aligned}$$

Reportons maintenant l'expression (3.67) de la séquence d'état dans (3.71). Cela produit

$$\begin{aligned}\begin{bmatrix} \Lambda_f Y^1 \\ \mathcal{Y} \end{bmatrix} &= \begin{bmatrix} \bar{A} & \bar{B} \\ \bar{C} & \bar{D} \end{bmatrix} \begin{bmatrix} \Lambda_f Y^o \\ \mathcal{U} \end{bmatrix}\\ &\quad - \begin{bmatrix} \bar{A} \\ \bar{C} \end{bmatrix} \Lambda_f H_f U^o + \begin{bmatrix} I_n \\ 0 \end{bmatrix} \Lambda_f H_f U^1,\end{aligned} \tag{3.72}$$

où

$$\begin{aligned} Y^1 &= Y(:, 2:N), \quad Y^o = Y(:, 1:N-1),\\ U^1 &= U(:, 2:N), \quad U^o = U(:, 1:N-1).\end{aligned}$$

Les seules inconnues dans le système d'équations (3.72) sont les matrices $\bar{A}, \bar{B}, \bar{C}, \bar{D}$ et H_f. Celles-ci pourraient être directement calculées si

$$\begin{bmatrix} (\Lambda_f Y^o)^\top & \mathcal{U}^\top & (U^o)^\top & (U^1)^\top \end{bmatrix}^\top \quad \text{et} \quad \begin{bmatrix} (\Lambda_f Y^o)^\top & \mathcal{U}^\top & (U^o)^\top \end{bmatrix}^\top$$

étaient toutes deux de rang plein ligne. Malheureusement, à cause des redondances évidentes dans les lignes de ces matrices, cela n'est pas le cas. Néanmoins, il est possible d'extraire directement $\bar{A}$ et $\bar{C}$. Pour cela, nous réécrivons (3.72) comme

$$\begin{aligned}\Lambda_f Y^1 &= \bar{A} \Lambda_f Y^o + \bar{B} \mathcal{U} - \bar{A} \Lambda_f \bar{H}_f U^o + \Lambda_f \bar{H}_f U^1,\\ \mathcal{Y} &= \bar{C} \Lambda_f Y^o + D \mathcal{U} - \bar{C} \Lambda_f \bar{H}_f U^o.\end{aligned} \tag{3.73}$$

Remarquons que

$$\begin{cases} \mathcal{U} = \begin{bmatrix} I_{n_u} & \mathrm{O}_{nu, (f-1)n_u} \end{bmatrix} U^o \\ U^1(1:(f-1)n_u, :) = \begin{bmatrix} \mathrm{O}_{(f-1)n_u, n_u} & I_{(f-1)n_u} \end{bmatrix} U^o, \end{cases}$$

d'où

$$\begin{cases} \mathcal{U} \Pi_{U^o}^\perp = 0 \\ U^1 \Pi_{U^o}^\perp = \left[\begin{array}{c} 0_{(f-1)n_u, N-1} \\ \hline -- \end{array}\right]. \end{cases}$$

Par conséquent, en multipliant la relation (3.73) à droite par la matrice de projection $\Pi_{U^o}^{\perp}$, il vient

$$\begin{aligned}\Lambda_f Y^1 \Pi_{U^o}^{\perp} &= \bar{A}\Lambda_f Y^o \Pi_{U^o}^{\perp} \\ &\quad + \Lambda_f^{(2)} \bar{D} U^1\big((f-1)n_u+1 : fn_u, :\big)\Pi_{U^o}^{\perp} \\ \mathcal{Y}\Pi_{U^o}^{\perp} &= \bar{C}\Lambda_f Y^o \Pi_{U^o}^{\perp},\end{aligned} \tag{3.74}$$

$\Lambda_f^{(2)} = \Lambda_f\big(:, (f-1)n_y : fn_y\big)$ désignant la matrice bloc constituée des n_y dernières colonnes de Λ_f. De la relation (3.74), on peut déterminer directement $\bar{A}$ et $\bar{C}$. Après cela, les matrices $\bar{B}$ et $\bar{D}$ peuvent être calculées une fois de plus par régression linéaire [76].

Remarque 3.3. *En absence de bruit, (3.67) peut se reécrire*

$$\bar{x}(t) = \Lambda_f\big(y_f(t) - H_f u_f(t)\big)$$

pour tout $t \geq 1$. Ainsi, (3.67) peut être interprété comme un observateur d'état à mémoire finie qui, contrairement à l'idée usuelle dans la conception des observateurs (exprimer l'état en fonction du passé), fournit l'état présent comme combinaison linéaire des entrées-sorties présentes et futures du système. Ainsi, pour un système autonome, l'état $\bar{x}(t) = \Lambda_f y_f(t)$ se réduit seulement à une certaine combinaison linéaire des sorties.

Remarque 3.4. *Pour toute matrice L, désignons par* $\mathrm{im}(L)$ *et* $\mathrm{null}(L)$ *respectivement l'espace des colonnes et l'espace noyau de L. Si $\Lambda_f \in \mathbb{R}^{n\times fn_y}$ vérifie la propriété (3.65), alors on a :*

1. $\mathrm{im}(\Gamma_f) = \mathrm{im}(\bar{\Gamma}_f) = \mathrm{null}(I_{fn_y} - \bar{\Gamma}_f\Lambda_f) = \mathrm{null}\left(I_{fn_y} - \Gamma_f(\Lambda_f\Gamma_f)^{-1}\Lambda_f\right)$.
2. $\mathrm{rang}(I_{fn_y} - \bar{\Gamma}_f\Lambda_f) = fn_y - n$.
3. $\mathrm{im}(I_{fn_y} - \bar{\Gamma}_f\Lambda_f) = \mathrm{null}(\Lambda_f)$.
4. *$I_{fn_y} - \bar{\Gamma}_f\Lambda_f$ est une matrice de projection c'est-à-dire,* $\left(I_{fn_y} - \bar{\Gamma}_f\Lambda_f\right)^2 = I_{fn_y} - \bar{\Gamma}_f\Lambda_f$. *Elle projette l'espace ambiant $\mathbb{R}^{fn_y}$ sur* $\mathrm{null}(\Lambda_f)$ *parallèlement à* $\mathrm{im}(\Gamma_f)$.
5. *L'espace vectoriel $\mathbb{R}^{fn_y}$ peut se décomposer comme*

$$\mathbb{R}^{fn_y} = \mathrm{im}(I_{fn_y} - \bar{\Gamma}_f\Lambda_f) \oplus \mathrm{null}(I_{fn_y} - \bar{\Gamma}_f\Lambda_f) = \mathrm{null}(\Lambda_f) \oplus \mathrm{im}(\Gamma_f),$$

où le symbole $\oplus$ fait référence à la somme directe de sous-espaces.

En nous basant sur cette remarque, nous pouvons interpréter l'équation (3.69) comme la projection de l'équation de données (3.64) sur $\mathrm{null}(\Lambda_f)$ parallèlement à l'espace $\mathrm{im}(\Gamma_f)$ des colonnes de Γ_f. Ainsi, il existe une connexion entre notre méthode et les algorithmes d'identification classiques des sous-espaces comme MOESP [122] et N4SID [115]. Cependant, une différence fondamentale est la suivante. Les méthodes classiques projettent l'équation (3.64) sur un sous-espace formé de données mesurées ($\mathrm{im}_{\mathrm{row}}(U_{f+1,f,N})^{\perp}$ pour MOESP et $\mathrm{im}_{\mathrm{row}}(\begin{bmatrix} U_{1,f,N}^{\top} & Y_{1,f,N}^{\top}\end{bmatrix}^{\top})$ pour N4SID, $\mathrm{im}_{\mathrm{row}}(\cdot)$ désignant l'espace

des lignes de $(\cdot)$) et donc variant selon la réalisation entrée-sortie utilisée. De plus ces méthodes nécessitent une DVS. Dans le cas de notre méthode, l'équation (3.64) est plutôt projetée sur le sous-espace null(Λ_f) qui est entièrement défini par Λ_f car null(Λ_f) est indépendant de la réalisation de données utilisée pour l'identification. Aussi, comme nous l'avons déjà indiqué, le choix de la matrice Λ_f détermine la base de représentation de l'état.

3.6.3 Cas stochastique

Considérons maintenant le cas plus réaliste où les mesures disponibles sont sujettes à des perturbations. Alors, la combinaison de (3.6) et (3.67) donne

$$Y = \bar{\Gamma}_f \Lambda_f Y + (I_{fn_y} - \bar{\Gamma}_f \Lambda_f) H_f U + \bar{E}, \tag{3.75}$$

où $\bar{E} = (I_{fn_y} - \bar{\Gamma}_f \Lambda_f)E$. On peut remarquer que la transformation (3.67) a une incidence directe sur les propriétés statistiques de la séquence du bruit $e_f(t)$. Par exemple, si $e_f(t)$ est un bruit blanc Gaussien avec pour matrice de covariance $R_e = \mathbb{E}(e_f(t)e_f(t)^\top) = \sigma_e^2 I_{fn_y}$, alors $\bar{e}_f(t) = (I_{fn_y} - \bar{\Gamma}_f \Lambda_f)e_f(t)$ est un bruit coloré de matrice de covariance

$$R_{\bar{e}} = \mathbb{E}(\bar{e}_f(t)\bar{e}_f(t)^\top) = \sigma_e^2 (I_{fn_y} - \bar{\Gamma}_f \Lambda_f)(I_{fn_y} - \bar{\Gamma}_f \Lambda_f)^\top.$$

Par conséquent, une estimation immédiate de la matrice d'observabilité étendue par les moindres carrés à partir de l'équation (3.75) est biaisée. Pour atténuer cet effet du bruit, nous adoptons la méthode bien connue de la variable instrumentale [27]. Il s'agit de construire une matrice d'instruments que nous noterons $\mathrm{Z} \in \mathbb{R}^{n_z \times N}$, $n_z \geq n$, (formée par exemple des entrées et sorties passées [29]), pour décorréler les données du bruit sans altérer l'information qu'elles portent.

Pour ce faire, nous éliminons comme précédemment, le terme $(I_{fn_y} - \bar{\Gamma}_f \Lambda_f)H_f$ dans l'équation (3.75) en post-multipliant celle-ci par $\Pi_U^\perp$. Il vient

$$Y\Pi_U^\perp = \bar{\Gamma}_f \Lambda_f Y \Pi_U^\perp + \bar{E}\Pi_U^\perp. \tag{3.76}$$

En multipliant la relation (3.76) à droite par la matrice d'instruments [1] $\mathrm{Z}^\top$ et en la divisant par N, on obtient par suite d'une combinaison avec la définition (3.14) de $\Pi_U^\perp$,

$$\begin{aligned} \frac{1}{N} Y\Pi_U^\perp \mathrm{Z}^\top =& \bar{\Gamma}_f \frac{1}{N} \Lambda_f Y \Pi_U^\perp \mathrm{Z}^\top + (I_{fn_y} - \bar{\Gamma}_f \Lambda_f)\frac{1}{N} E\mathrm{Z}^\top \\ & - (I_{fn_y} - \bar{\Gamma}_f \Lambda_f)\frac{1}{N} EU^\top \big(\frac{1}{N} UU^\top\big)^{-1} \frac{1}{N} U\mathrm{Z}^\top. \end{aligned} \tag{3.77}$$

Avec les hypothèses que la séquence $\{u(t)\}$ est ergodique et indépendante de $\{w(t)\}$ et $\{e(t)\}$, on peut

1. On pourrait aussi appliquer la méthode de la variable instrumentale aux méthodes des sections 3.4 et 3.5.

voir que $\lim_{N\to\infty} \frac{1}{N} EU^\top = 0$ dans (3.77). Ainsi, Eq. (3.77) se réduit asymptotiquement à

$$\lim_{N\to\infty} \left(\frac{1}{N} Y\Pi_U^\perp \mathrm{Z}^\top\right) \quad = \quad \bar{\Gamma}_f \lim_{N\to\infty} \left(\frac{1}{N}\Lambda_f Y\Pi_U^\perp \mathrm{Z}^\top\right) \; + \; (I_{fn_y} \; - \; \bar{\Gamma}_f\Lambda_f) \lim_{N\to\infty} \left(\frac{1}{N} E\mathrm{Z}^\top\right). \quad (3.78)$$

Grâce à l'équation précédente, il apparaît que pour obtenir une estimée de $\bar{\Gamma}_f$ qui soit consistante et décorrelée de l'effet du bruit, la matrice d'instruments Z doit être conçue de façon à réaliser les deux conditions suivantes

$$\begin{cases} \lim\limits_{N\to\infty} \left(\frac{1}{N} E\mathrm{Z}^\top\right) = 0, \\ \mathrm{rang}\left(\lim\limits_{N\to\infty} \frac{1}{N}\Lambda_f Y\Pi_U^\perp \mathrm{Z}^\top\right) = n. \end{cases} \quad (3.79)$$

Lorsque Z satisfait aux conditions (3.79), il est facile de vérifier que la relation asymptotique suivante

$$\lim_{N\to\infty} \frac{1}{N} Y\Pi_U^\perp \mathrm{Z}^\top = \bar{\Gamma}_f \lim_{N\to\infty} \left(\frac{1}{N}\Lambda_f Y\Pi_U^\perp \mathrm{Z}^\top\right). \quad (3.80)$$

De cette relation on peut extraire $\bar{A}$ et $\bar{C}$ puis estimer à partir de ces matrices, $\bar{B}$ et $\bar{D}$ par régression linéaire comme indiqué dans la partie 3.6.1.
En retournant maintenant à la question de savoir comment choisir efficacement la matrice d'instruments Z, nous suivrons juste [29] où il a été montré que la concatenation des entrées et sorties passées,

$$\mathrm{Z} = \begin{bmatrix} U_{1,f,N} \\ Y_{1,f,N} \end{bmatrix} \quad (3.81)$$

constitue un choix valide d'instruments.

3.6.4 Sur le choix de la matrice de pondération Λ_f

La méthode d'identification décrite depuis le début de la section 3.6 repose sur la possibilité de trouver une matrice $\Lambda_f \in \mathbb{R}^{n\times fn_y}$ de forme quelconque telle que la condition (3.65) soit satisfaite. Dès lors, on peut naturellement se demander si une telle matrice existe toujours et surtout, comment la sélectionner alors que Γ_f est inconnue. La réponse à la question de l'existence est évidente sous l'hypothèse que le système considéré est observable car dans ce cas, $\Lambda_f = \Gamma_f^\top$ (par exemple), vérifie (3.65). Mais, comme Γ_f n'est pas connu, nous ne pouvons pas choisir Λ_f égal à $\Gamma_f^\top$. Nous allons donc nous intéresser un peu plus à la façon dont Λ_f pourrait être sélectionné. Bien que Γ_f ne soit pas connu, nous allons montrer qu'il n'est en fait pas difficile de trouver une matrice Λ_f qui satisfait (3.65).

En se référant aux Propositions 3.1 et 3.2, il est facile de voir que si $\{u(t)\}$ est SE$(n+f)$, alors $\mathrm{rang}(\Lambda_f\Gamma_f) = n$ si et seulement si $\mathrm{rang}(\Lambda_f Y\Pi_U^\perp) = n$. En vertu de cette remarque, Λ_f peut être sélectionné juste de manière à vérifier $\mathrm{rang}(\Lambda_f Y\Pi_U^\perp) = n$ puisque contrairement à Γ_f, $Y\Pi_U^\perp$ est connu. Mais cette solution pourrait nécessiter une DVS. Ainsi, une alternative consiste à générer Λ_f de façon aléatoire comme suggéré par la proposition suivante.

Proposition 3.5. *Supposons que le système (3.1) soit observable et considérons une matrice* $\Lambda_f \in$

$\mathbb{R}^{n\times fn_y}$ générée de façon aléatoire suivant une distribution uniforme. Alors, la condition $\text{rang}(\Lambda_f\Gamma_f) = n$ *est vérifiée avec une probabilité de un.*

Preuve. La preuve est similaire à celle de la proposition 3.3. ■

Ainsi, il suffit de générer Λ_f selon une distribution uniforme (par exemple) pour que les estimées obtenues précédemment correspondent aux matrices du système réel avec une probabilité de un.

Remarque 3.5 (Sur l'estimation de l'ordre)**.** *Dans le cas où l'ordre n du système (3.1) est inconnu, il serait intéressant de pouvoir l'estimer sans devoir utiliser une DVS. Notons que la procédure proposée à la section 3.4 pour l'identification de l'ordre peut se généraliser à la méthode de la section 3.6. Pour cela, il suffit de sélectionner*[1] *Λ_f dans $\mathbb{R}^{fn_y\times fn_y}$ telle que pour tout $r \leq n$, $\text{rang}(\Lambda_f^{1:r}\Gamma_f) = r$, avec $\Lambda_f^{1:r} = \Lambda_f(1:r,:)$. Pour illustrer la faisabilité de cette idée, faisons abstraction du bruit dans (3.7). Alors de $Y = \Gamma_f X + H_f U$, on peut écrire*

$$\begin{bmatrix}\Lambda_f^1\\ \Lambda_f^2\end{bmatrix} Y\Pi_U^\perp = \begin{bmatrix}\Lambda_f^1\\ \Lambda_f^2\end{bmatrix}\Gamma_f X\Pi_U^\perp,$$

où $\Lambda_f^1 = \Lambda_f(1:n,:)$ et $\Lambda_f^2 = \Lambda_f(n+1:fn_y,:)$. En élevant le terme de gauche au carré on peut, de la même manière que dans (3.36), poser

$$\Sigma = \begin{bmatrix}\Lambda_f^1\Sigma_{yu}(\Lambda_f^1)^\top & \Lambda_f^1\Sigma_{yu}(\Lambda_f^2)^\top\\ \Lambda_f^2\Sigma_{yu}(\Lambda_f^1)^\top & \Lambda_f^2\Sigma_{yu}(\Lambda_f^2)^\top\end{bmatrix},$$

avec $\Sigma_{yu} = Y\Pi_U^\perp Y^\top$. Ainsi, il est possible d'appliquer ici la même procédure que dans la sous-section 3.4.3 pour la recherche de l'ordre.

L'alternative traditionnelle, elle, nécessite un calcul des valeurs singulières :

$$n = \underset{r=1,\ldots,f-1}{\arg\min}\ \frac{\sigma_r^2\big(Y\Pi_U^\perp\big)}{\sum_{j=1}^{r-1}\sigma_j^2\big(Y\Pi_U^\perp\big)} + \kappa(r), \tag{3.82}$$

où $\sigma_1(M) \geq \sigma_2(M) \geq \cdots$, sont les valeurs singulières de M, données dans un ordre non décroissant et $\kappa(r)$ est une fonction croissante de r qui symbolise ici la complexité du modèle à déterminer. Un problème cependant est qu'il est fastidieux de choisir $\kappa(r)$ de façon satisfaisante.

Remarque 3.6 (Comparaison des trois algorithmes)**.** *A quelques différences d'implémentation près, les méthodes des sections 3.4 et 3.5 peuvent être regardées comme des cas particuliers de celle discutée dans*

1. Λ_f peut encore être générée de façon aléatoire.

la section 3.6 et correspondent en fait à un choix de la matrice Λ_f sous la forme

$$\Lambda_f = \left[\begin{array}{ccc|c} \gamma^\top & \cdots & 0 & \\ \vdots & \ddots & \vdots & O_{n,(f-n)n_y} \\ 0 & \cdots & \gamma^\top & \end{array}\right] \in \mathbb{R}^{n\times fn_y}.$$

Sur un système observable, nous avons vu que pour qu'un tel choix de Λ_f vérifie la condition (3.65), il faut et il suffit que A soit non dérogatoire.

Remarque 3.7 (Non unicité de la relation entrée-sortie)**.** *Soit $M \in \mathbb{R}^{n\times nn_y}$ une matrice telle que $M\Gamma_n$ soit carrée et inversible. En ignorant dans (3.1) les termes décrivant l'effet du bruit, on obtient pout $t > n$:*

$$x(t-n) = (M\Gamma_n)^{-1}\big(My_n(t-n) - MH_nu_n(t-n)\big).$$

Par conséquent,

$$\begin{aligned} x(t) = {} & A^n(M\Gamma_n)^{-1}My_n(t-n) \\ & + \big(\Delta_n - A^n(M\Gamma_n)^{-1}MH_n\big)u_n(t-n), \end{aligned}$$

et donc en faisant usage du théorème de Cayley-Hamilton, on obtient après calcul,

$$\begin{aligned} y(t) = {} & -(a^\top \otimes I_{n_y})\Gamma_n(M\Gamma_n)^{-1}My_n(t-n) \\ & + \Big(C\Delta_n + (a^\top \otimes I_{n_y})\Gamma_n(M\Gamma_n)^{-1}MH_n\Big)\,u_n(t-n) \\ & + Du(t), \end{aligned}$$

pour tout $t > n$. Dans cette expression, $\Gamma_n(M\Gamma_n)^{-1}M$ n'est pas génériquement égale à l'identité I_{nn_y} lorsque $n_y > 1$ car $\operatorname{rang}\big(\Gamma_n(M\Gamma_n)^{-1}M\big) = n < nn_y$*. Or, pour un Γ_n donné, de rang plein colonne (observable), il existe clairement une infinité de matrices M vérifiant la propriété* $\operatorname{rang}(M\Gamma_n) = n$*. Ainsi, d'une façon quelque peu surprenante, il existe une infinité de modèles MIMO ($n_y > 1$) entrée-sortie minimaux qui réalisent le comportement entrée-sortie du système (3.1).*

3.7 Exemples numériques

Afin d'évaluer les algorithmes présentés dans ce chapitre, nous considérons ici un exemple de système linéaire d'ordre trois avec deux entrées et deux sorties. Les matrices du système considéré sont données par :

$$A = \begin{bmatrix} -0.0398 & -0.4909 & -0.6115 \\ 0.4784 & 0.4808 & -0.4169 \\ -0.6214 & 0.4020 & -0.2488 \end{bmatrix}, \; B = \begin{bmatrix} 0.7433 & 0 \\ 0 & -1.3193 \\ -0.8510 & -0.0181 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.0028 & 2.3968 & 0 \\ -0.0857 & 0.0264 & -0.6553 \end{bmatrix}, \quad D = \begin{bmatrix} -1.7384 & 0.0425 \\ 0.7127 & 0 \end{bmatrix}.$$

L'entrée d'excitation est choisie comme un bruit blanc Gaussien, centré et de variance unité. Les bruits d'état $w(t)$ et de sortie $e(t)$ sont tous deux simulés dans des rapports raisonnables de 25 dB respectivement de l'état et de la sortie. L'ordre n du système est supposé connu. [1] Le paramètre f est pris égal à 5 tandis que la matrice Λ_f et le vecteur γ sont générés de façon aléatoire. Nous procédons alors à une simulation de Monte-Carlo de 1000 tirages sur chacune de nos trois méthodes discutées, avec différentes réalisations des bruits et de l'entrée d'excitation. Chacune de ces 1000 simulations porte sur un échantillon de 1000 points. Dans un objectif de comparaison, nous effectuons dans les mêmes conditions, le même type de test sur la méthode MOESP [122]. Nous utilisons pour cette dernière méthode, la Toolbox d'identification des sous-espaces développé dans [54].
Dans un premier temps, aucun traitement spécial n'est appliqué au bruit affectant les données. Les résultats alors obtenus sont présentés ci-dessous à titre indicatif.
La légende des figures est la suivante.

Propagateur MIMO	:	Méthode de la section 3.4
Pondération E-S	:	Méthode de la section 3.5
Pondération S-S	:	Méthode de la section 3.6
MOESP	:	Méthode développée dans [122], [54].

Sur la figure 3.1, nous présentons les valeurs singulières des matrices de transfert du système réel superposées à celles des estimées obtenues respectivement par les quatre méthodes mentionnées ci-dessus. La figure 3.2 consiste en un zoom de la figure 3.1 sur l'intervalle de fréquence [0, 300 rad/s]. Il apparaît sur cette dernière figure qu'en absence de variable instrumentale, les estimées obtenues par la méthode MOESP présentent une variance légèrement plus forte que celle des estimées fournies par les trois autres algorithmes. Cette tendance est confirmée par les figures 3.5 et 3.6.

Afin de donner aussi une idée de la précision avec laquelle la matrice de transfert entrée-sortie est estimée par les différentes méthodes, nous montrons à titre d'exemple sur les figures 3.3 et 3.4, le gain du transfert entre la première sortie et la première entrée du système. Sur cette figure, il apparaît que toutes les quatre méthodes étudiées présentent des variances à peu près similaires.
Pour apprécier les performances de nos méthodes dans un sens statistique, nous représentons sur la figure 3.5 les histogrammes des erreurs relatives $\left\|M - \hat{M}\right\| / \|M\|$ entre les paramètres de Markov (qui sont des invariants de la base d'état) du modèle réel et les paramètres de Markov des modèles estimés. Selon

1. Nous testerons notre méthode d'estimation de l'ordre (cf. Section 3.4.) au Chapitre 4 sur un exemple de système linéaire commutant

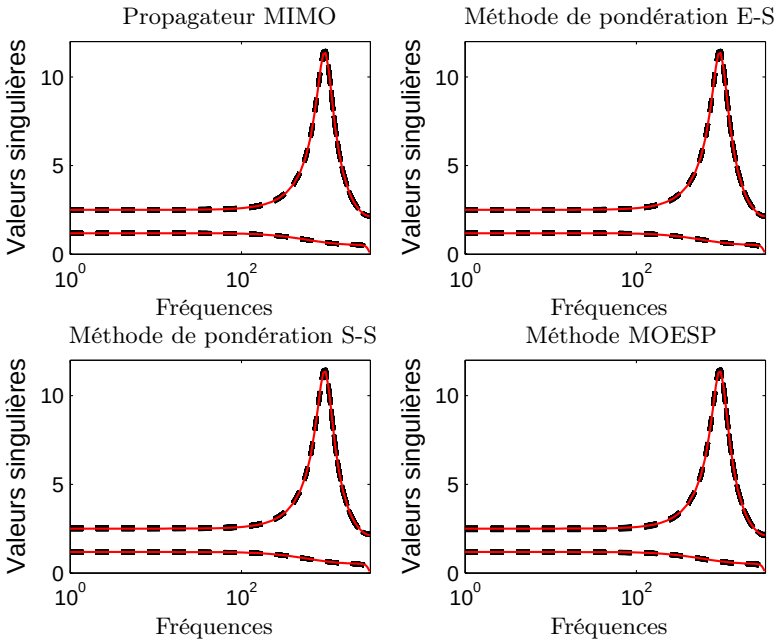


FIGURE 3.1 – Représentation (en fonction de la fréquence) des deux valeurs singulières de la matrice de transfert $G(e^{jw}) = C(e^{jw}I_n - A)^{-1}B + D$. Il s'agit en fait d'une superposition des 1000 transferts estimés (en pointillés noirs) et du transfert réel (en trait plein rouge).

ce résultat, nos trois algorithmes donnent une erreur relative inférieure à 0.02 sur l'ensemble des 1000 simulations effectuées. Une fois encore l'estimation avec l'algorithme MOESP résulte en une variance plus grande.

La figure 3.6 superpose dans le plan complexe, les pôles (un pôle réel et deux pôles complexes conjugués) du système réel à ceux de son modèle estimé sur l'ensemble de toutes les 1000 simulations. On peut noter que lorsqu'aucune Variable Instrumentale (VI) n'est utilisée pour le traitement du bruit, les différents estimateurs discutés sont biaisés du fait de la présence du bruit. Cette constatation n'est pas surprenante puisque nous l'avons théoriquement mise en évidence plus tôt. Le bruit considéré dans les données n'étant pas strictement limité à un bruit blanc de sortie comme dans [120], l'algorithme MOESP semble aussi légèrement biaisé. Mais une impression générale est que, sur notre exemple, les pôles complexes semblent assez bien estimés par rapport au pôle réel.

Dans une seconde expérience, nous appliquons la méthode de la VI pour réduire l'effet du bruit dans les données. La VI est construite ici avec les entrées et les sorties passées. Nous procédons une fois encore à une simulation de Monte-Carlo de 1000 tirages sur chacune des quatre méthodes mentionnées, avec différentes réalisations des bruits et de l'entrée d'excitation. La figure 3.7 représente les pôles des modèles estimés. On peut voir que l'utilisation de la VI permet de réduire, sinon de supprimer presque complètement le biais mis en évidence à la figure 3.6. Cependant, dans le cas de la méthode E-S et S-S, la variance du pôle réel est plus importante. Elle est même plus forte qu'en absence de variable instrumentale comme le montre la figure 3.6.

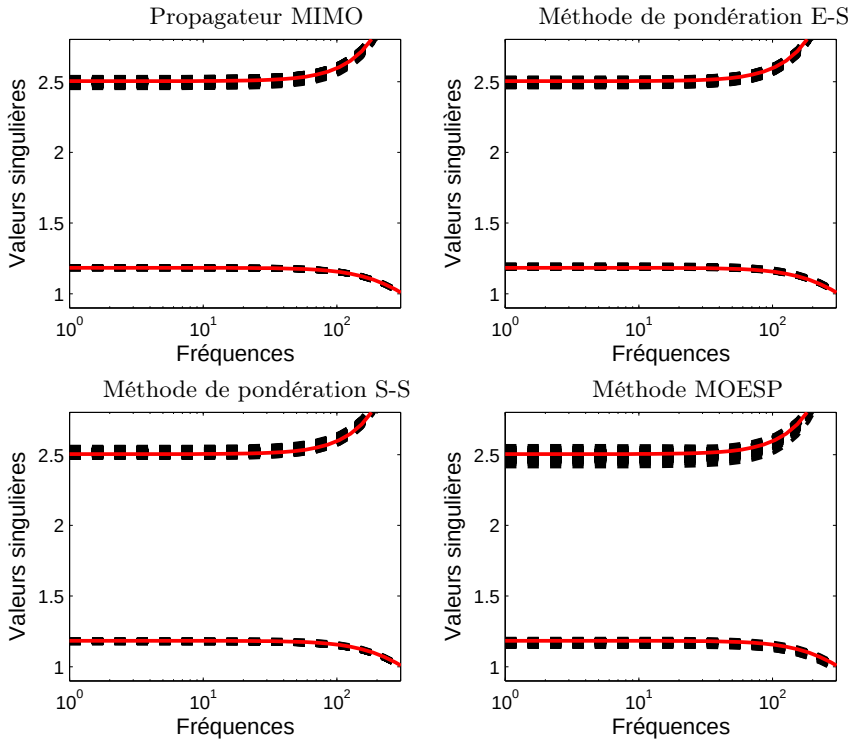


FIGURE 3.2 – Zoom de la figure 3.1 sur l'intervalle de fréquences [0, 300 rad/s]. Représentation (en fonction de la fréquence) des deux valeurs singulières de la matrice de transfert $G(e^{jw}) = C(e^{jw}I_n - A)^{-1}B + D$.

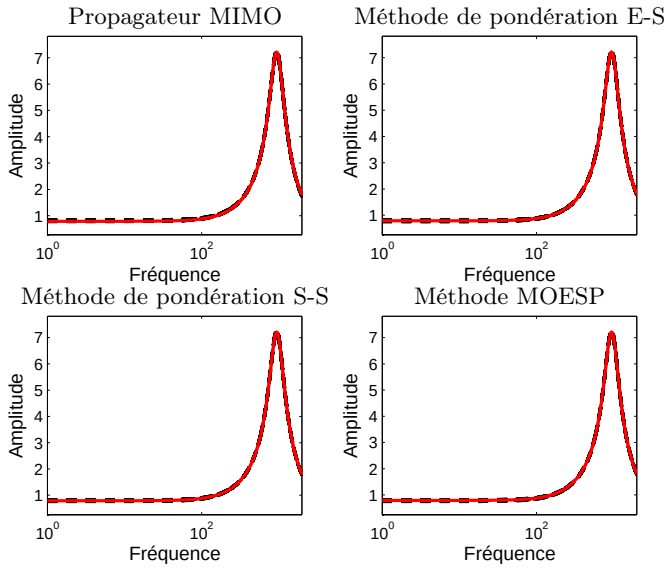


FIGURE 3.3 – Amplitudes de la $(1,1)$-ème entrée de la fonction transfert. Transfert réel (trait plein rouge) et transferts estimés sur une simulation de Monte-Carlo de 100 jeux (pointillés noirs).

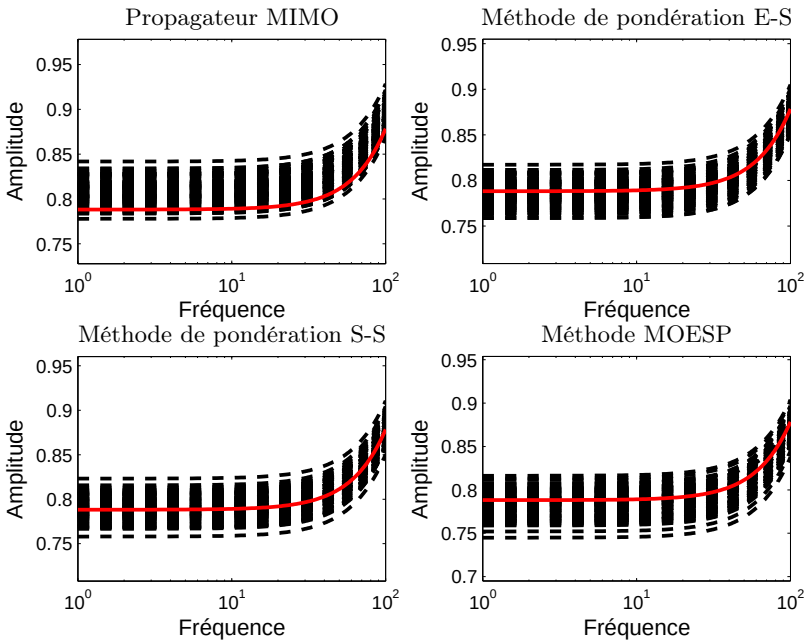


FIGURE 3.4 – Amplitudes de la $(1,1)$-ème entrée de la fonction transfert. Transfert réel (trait plein rouge) et transferts estimés sur une simulation de Monte-Carlo de 100 jeux (pointillés noirs).

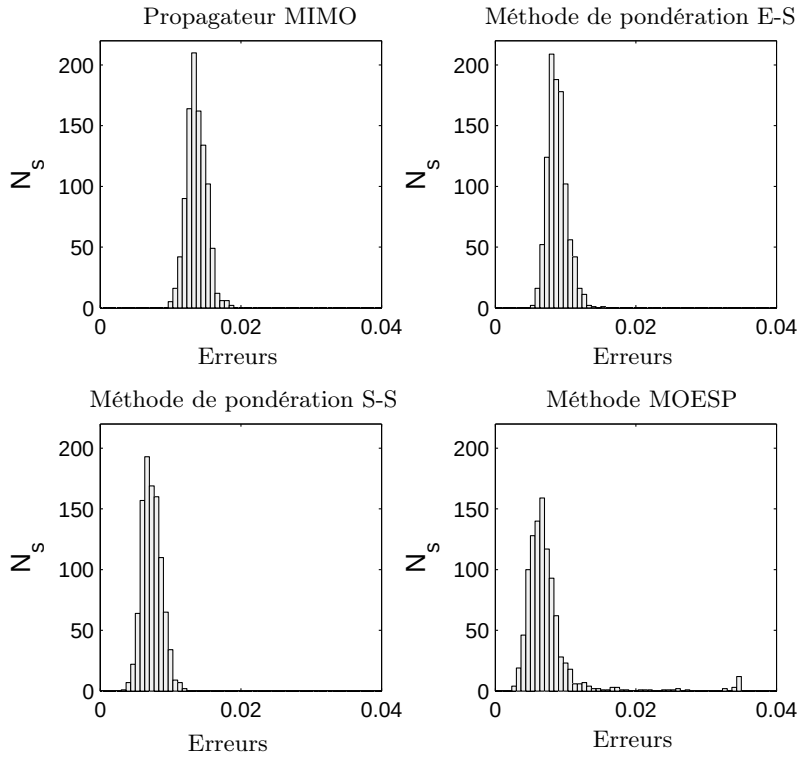


FIGURE 3.5 – Histogrammes des erreurs relatives entre les Paramètres de Markov du système réel d'une part et les paramètres de Markov du modèle estimé d'autre part.

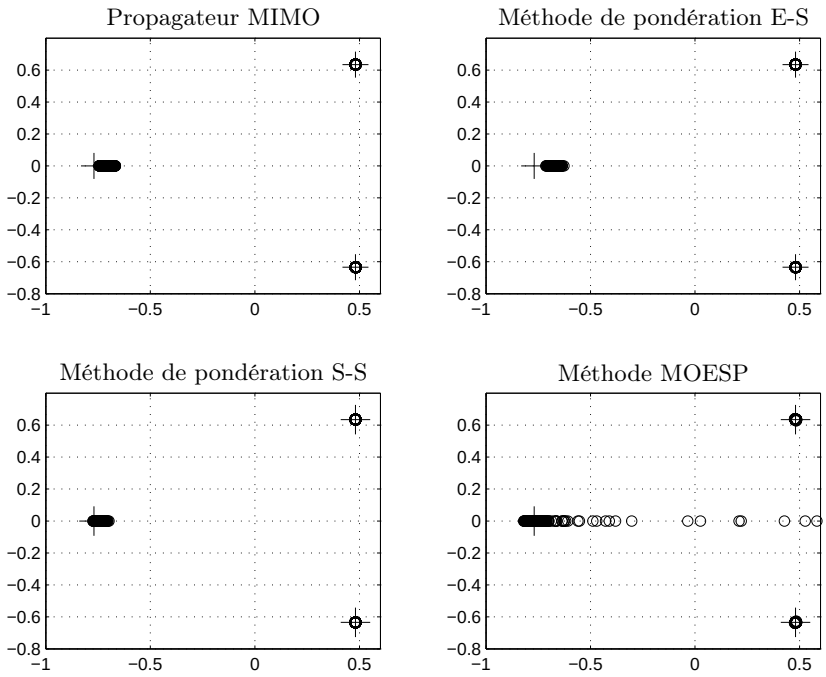


FIGURE 3.6 – Pôles du système réel (grandes croix) et pôles du modèle estimé (cercles).

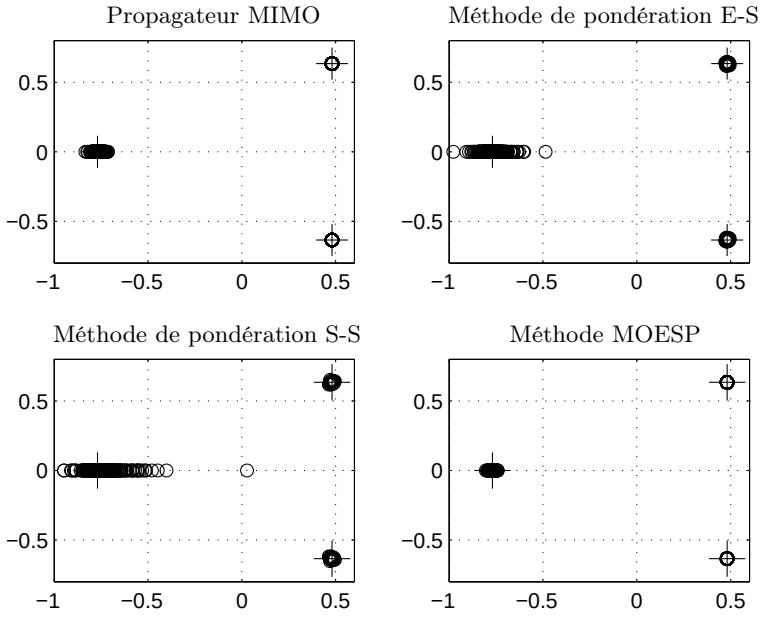


FIGURE 3.7 – Pôles du système réel (grandes croix) et pôles du modèle estimé (cercles). Ces estimées sont obtenues avec traitement des données par une variable instrumentale construite à partir des données passées (cf. Eq. (3.81)).

Tous les résultats de simulation présentés illustrent clairement le potentiel de nos méthodes comme de sérieuses alternatives aux méthodes classiques des sous-espaces pour l'identification directe de modèles d'état.

3.8 Conclusion

Nous avons proposé dans ce chapitre quelques nouvelles méthodes d'identification de modèles d'état pour des systèmes MIMO linéaires, avec comme objectif envisagé une application ultime aux systèmes linéaires à commutations. Les méthodes appartiennent toutes à une catégorie de techniques que nous convenons de désigner par *méthodes d'identification structurée* de modèles d'états. La caractéristique principale de ces méthodes est qu'elles sont basées sur une fixation *a priori* de la base de la réalisation d'état à déterminer. Cela est effectué par l'introduction de matrices de pondération spécifiques qui sont astreintes à vérifier certaines conditions de rang. Nous arrivons ainsi à éliminer l'inconnu que représente la séquence d'état dans un modèle d'état et donc à transformer le problème d'identification des sous-espaces en un problème de moindres carrés ordinaires.

Une caractéristique importante de nos méthodes est qu'elles ne nécessitent pas l'utilisation du très coûteux algorithme de Décomposition en Valeurs Singulières contrairement à la plupart des méthodes des sous-espaces existantes. De plus, elles présentent l'avantage remarquable d'affranchir les matrices d'état identifiées de l'aléa de la base d'état. On peut désormais, indépendamment du jeu de données entrée-sortie, fixer la base d'état et en avoir une maîtrise complète. Généralement, dans le cadre de l'identification récursive, ou de l'estimation de systèmes composites (avec plusieurs sous-modèles linéaires), il est nécessaire, voire indispensable que les matrices des différents sous-modèles soient obtenues dans la même base. Nos algorithmes s'avèrent, comme nous le verrons dans le chapitre 4, être d'un grand intérêt dans ces types de situations.

4

Identification de modèles d'état commutants

Sommaire

4.1 Introduction

Ce chapitre est dédié à l'identification de systèmes linéaires commutants dont les sous-systèmes sont décrits par des modèles d'état. Ce problème peut être génériquement compris comme celui de l'estimation, à partir de mesures entrée-sortie, des ordres, des paramètres et du nombre de sous-modèles composant le système à commutations. La principale difficulté du problème se situe dans le fait que l'état discret, l'état continu et les paramètres des différents sous-modèles sont très couplés et sont malheureusement tous inconnus. De ce fait, l'identification de modèles d'états commutants est un problème très délicat qui n'a pas encore reçu beaucoup d'attention dans la littérature. En effet, comme nous l'avons montré dans le chapitre 2, la plupart des contributions existantes sur le sujet de l'identification de systèmes hybrides s'appliquent aux classes de modèles PWARX ou SARX, avec en principe, des commutations arbitraires, c'est-à-dire qu'aucun temps de séjour minimum n'est explicitement requis entre deux changements consécutifs de modes [46], [67], [90].

Cependant, il peut être souvent pratique d'identifier directement des modèles d'état commutants. Cela est particulièrement sensé dans le cas des systèmes MIMO parce que les modèles d'état permettent de représenter ces derniers systèmes d'une façon très compacte. Les notions telles que l'observabilité ou l'atteignabilité sont bien définies et habituellement étudiées sur des réalisations d'état. De plus, beaucoup de méthodes relatives à la commande, la conception d'observateurs, la détection de défauts et diverses autres méthodes d'analyse de systèmes multivariables, s'appuient sur des modèles d'état. Ainsi, l'identification de systèmes hybrides linéaires directement sous forme d'état peut être d'un certain intérêt. Les travaux relatifs à ce sujet opèrent tous sous l'hypothèse que deux instants de commutation consécutifs sont séparés par un certain temps de séjour minimum, suffisamment long pour permettre l'application des méthodes d'identification des sous-espaces entre ces deux instants [9], [22], [58], [93], [119].

En rupture avec cette traditionnelle simplification, nous commençons ici (Section 4.3) par considérer le problème de l'identification de systèmes linéaires commutants sous un angle très général c'est-à-dire, sans hypothèse de temps de séjour. Tout en mettant en évidence l'ambiguïté et la complexité du modèle d'état pour la description de systèmes commutants, nous proposons une méthode assez générale pour l'estimation de ce type de modèle. Malheureusement, sans hypothèse concernant la fréquence des commutations, cette méthode s'avère numériquement très complexe : le nombre de sous-modèles est exponentiellement grand sur une fenêtre de données sur laquelle l'état continu peut être éliminé.

C'est pourquoi nous nous intéressons ensuite (Section 4.4) au cas particulier de systèmes commutants avec temps de séjour. En nous basant alors sur les techniques développées au chapitre 3, nous proposons une méthode d'identification qui réalise en ligne les multiples tâches d'estimation, de détection et de décision. Dans cette dernière procédure, les ordres des sous-modèles peuvent être inconnus et peuvent même différer d'un sous-modèle à un autre. De plus, il n'est pas nécessaire d'avoir une connaissance *a priori* du nombre de sous-modèles.

4.2 Réalisation de systèmes à commutations

4.2.1 Première formulation du problème d'identification

Dans ce chapitre, nous considérons principalement l'identification de Systèmes Linéaires Commutants (SLC) représentés par un modèle d'état à temps discret. Les sections 4.2 et 4.3 sont basées sur un modèle [1] du type

$$\begin{cases} x(t+1) &= A_{\lambda_t}x(t) + B_{\lambda_t}u(t) \\ y(t) &= C_{\lambda_t}x(t) + D_{\lambda_t}u(t), \end{cases} \tag{4.1}$$

où $\lambda_t \in \mathcal{S} = \{1, \ldots, s\}$ fait référence à l'état discret du système, s est le nombre d'états discrets ou de sous-modèles du système hybride global et n est l'ordre du système, supposé être le même pour tous les sous-modèles. Les vecteurs $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^{n_u}$, $y(t) \in \mathbb{R}^{n_y}$ sont respectivement l'état, l'entrée et la sortie du système. Les matrices $A_{\lambda_t}, B_{\lambda_t}, C_{\lambda_t}, D_{\lambda_t}$ sont les matrices de paramètres du sous-modèle indexé par λ_t. Il convient de préciser que dans l'étude qui va suivre, il n'est fait aucune hypothèse concernant le mécanisme de commutation. Ainsi, la loi de commutation peut être fonction du temps ou de l'état continu du système ; elle peut aussi être événementielle, exogène, déterministe ou stochastique. Dans le but de simplifier les équations du système, nous faisons abstraction, dans le modèle (4.1), des bruits éventuels qui peuvent affecter le système.

Problème 4.1. *Etant donné une collection finie d'observations entrée-sortie* $\{u(t), y(t)\}_{t=1}^{N}$ *générées par un modèle linéaire commutant du type (4.1), estimer le nombre* s *de sous-modèles, leur ordre* n *ainsi que leurs matrices respectives* $\{A_j, B_j, C_j, D_j\}_{j=1}^{s}$.

De toute évidence, l'inférence du modèle (4.1) à partir des observations entrée-sortie est un problème assez complexe. Il est donc naturel de commencer ici par se demander sous quelles conditions un tel problème peut être résolu. Ainsi, avant de procéder à l'identification concrète du modèle (4.1), il est important d'étudier quelques propriétés fondamentales de ce dernier. Par exemple, il est primordial de savoir sous quelles conditions le modèle (4.1) peut être identifié de façon unique à partir de ses mesures entrée-sortie. Aussi dans le cas où il est possible d'inférer un modèle, quel sens peut-on donner en pratique à ce modèle ?

Le problème qui consiste à rechercher l'ensemble des modèles du type (4.1) qui traduisent le comportement entrée-sortie du même système est classiquement connu comme le problème de la réalisation de systèmes linéaires commutants [94]. Contrairement à la théorie de la réalisation des systèmes linéaires qui est maintenant très solidement développée [69], la réalisation des systèmes hybrides reste encore un problème très ouvert. Il existe néanmoins quelques travaux qui introduisent le sujet [94, 128, 103, 91] et auxquels nous nous référons tout le long du chapitre.

1. Bien que l'équation (4.1) ne soit qu'une représentation mathématique du système considéré, elle est considérée ici comme la génératrice réelle des données. A ce titre, elle sera appelée de façon interchangeable, modèle ou système.

4.2.2 Comportement entrée-sortie d'un modèle d'état commutant

Dans cette sous-section nous étudions la réalisabilité du comportement entrée-sortie associé au modèle d'état (4.1). Etant donné les entrées et les sorties générées par un modèle de la forme (4.1), nous nous intéressons à l'existence puis à la forme et au nombre d'autres modèles potentiels qui pourraient expliquer ces mêmes données. En d'autres termes nous devons tenter de répondre brièvement aux questions suivantes. Existe-t-il d'autres modèles que (4.1) qui représentent les données $\{u(t), y(t)\}_{t=1}^{N}$? S'il en existe, alors quelles formes peuvent avoir ces modèles et quelles relations les lient éventuellement au modèle (4.1) qui est le générateur effectif des données ?

Pour procéder à cette analyse, nous posons, pour tout état discret $\lambda_t \in \mathcal{S}$,

$$M_{\lambda_t} = \left[\begin{array}{c|c} A_{\lambda_t} & B_{\lambda_t} \\ \hline C_{\lambda_t} & D_{\lambda_t} \end{array}\right].$$

Alors, le modèle (4.1) peut être réécrit sous la forme

$$\begin{bmatrix} x(t+1) \\ y(t) \end{bmatrix} = M_{\lambda_t} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}. \tag{4.2}$$

Désignons maintenant par $\mathscr{E}$, la trajectoire entrée-sortie [132] du système (4.1) définie par

$$\mathscr{E} = \left\{ \big(u(t), y(t)\big) \in \mathbb{R}^{n_u} \times \mathbb{R}^{n_y}, \exists\big(\{\lambda_t\}, \{x(t)\}\big) \subset \mathcal{S} \times \mathbb{R}^n : \begin{bmatrix} x(t+1) \\ y(t) \end{bmatrix} = M_{\lambda_t} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \right\},$$

c'est-à-dire, $\mathscr{E}$ est l'ensemble des couples $\big(u(t), y(t)\big)$, $t \in \mathbb{N}$, tel qu'il existe une séquence d'états discrets $\{\lambda_t : t \in \mathbb{N}\}$ et d'états continus $\{x(t) : t \in \mathbb{N}\}$ satisfaisant à l'équation (4.2). Si on se réfère à l'équation de récurrence du modèle (4.1), alors l'ensemble $\mathscr{E}$ correspond simplement à la donnée d'un état continu initial $x(0)$, et d'une séquence d'états discrets $\{\lambda_t : t \in \mathbb{N}\}$. Comme dans le cas d'un système linéaire, étant donné la trajectoire entrée-sortie $\mathscr{E}$ d'un système du type (4.1), il existe une infinité de modèles qui peuvent reproduire le même comportement. En effet, on peut écrire à partir de (4.1),

$$\begin{aligned} y(t) =& C_{\lambda_t} A_{\lambda_{t-1}} \dots A_{\lambda_0} x(0) + C_{\lambda_t} A_{\lambda_{t-1}} \dots A_{\lambda_1} B_{\lambda_0} u(0) \\ &+ C_{\lambda_t} A_{\lambda_{t-1}} \dots A_{\lambda_2} B_{\lambda_1} u(1) + \dots + C_{\lambda_t} A_{\lambda_{t-1}} B_{\lambda_{t-2}} u(t-2) \\ &+ C_{\lambda_t} B_{\lambda_{t-1}} u(t-1) + D_{\lambda_t} u(t). \end{aligned} \tag{4.3}$$

Si dans cette équation, on remplace selon le schéma

$$\left[\begin{array}{c|c} A_{\lambda_t} & B_{\lambda_t} \\ \hline C_{\lambda_t} & D_{\lambda_t} \end{array}\right] \leftarrow \left[\begin{array}{c|c} T_{t+1} A_{\lambda_t} T_t^{-1} & T_{t+1} B_{\lambda_t} \\ \hline C_{\lambda_t} T_t^{-1} & D_{\lambda_t} \end{array}\right], \tag{4.4}$$

$A_{\lambda_t}, B_{\lambda_t}, C_{\lambda_t}$ respectivement par $T_{t+1} A_{\lambda_t} T_t^{-1}$, $T_{t+1} B_{\lambda_t}$, $C_{\lambda_t} T_t^{-1}$, où $\{T_k\}$ est une séquence de matrices non singulières, et si de plus on substitue $T_0 x(0)$ à $x(0)$, alors la fonction qui exprime la sortie en fonction

de l'entrée est inchangée. Au lieu d'une séquence infinie de matrices $\{T_t\}$ indexée par le temps, il est possible de choisir une séquence de matrices non singulières $\{T_{\lambda_t}\}$ indexées plutôt par l'état discret. Ainsi, étant donné la trajectoire entrée-sortie $\mathscr{E}$, une dimension n de l'espace d'état, un nombre s d'états discrets et une séquence $\{\lambda_t\} \subset \mathcal{S}$, il existe une infinité de modèles $\left(A_{\lambda_t}, B_{\lambda_t}, C_{\lambda_t}, D_{\lambda_t}\right)$ qui réalisent $\mathscr{E}$ pour un $x(0)$ convenablement choisi (comme indiqué ci-dessus).
De même, à moins d'imposer des conditions particulières sur le nombre s et la séquence $\{\lambda_t\}$ des états discrets, il est concevable qu'avec un nombre d'états discrets plus important, on puisse produire la même séquence de sortie. Une analyse intéressante a été justement présentée dans [128] concernant l'identifiabilité des systèmes linéaires commutants. Notre objectif principal étant ici l'identification de systèmes commutants, nous nous contenterons juste de résumer cette analyse comme suit :
Etant donné une trajectoire entrée-sortie $\mathscr{E}$ effectivement générée par un modèle du type (4.1), comportant s états discrets et un ordre n, il est possible de trouver une infinité de modèles [1] $\{A_j, B_j, C_j, D_j\}_{j=1}^{\bar{s}}$ d'ordre $\bar{n}$, avec d'une part, $\bar{n} \leq n$ et $\bar{s} \geq s$ ou d'autre part, $\bar{n} \geq n$ et $\bar{s} \leq s$, qui réalisent la même trajectoire $\mathscr{E}$. Cela signifie qu'en général, on ne peut pas, à partir de l'observation de la seule trajectoire $\mathscr{E}$ retrouver le « vrai » modèle (c'est-à-dire l'ordre réel n, le nombre réel d'états discrets s, la séquence réelle des états discrets, les matrices réelles) qui en est le générateur [128].

Dans le but d'élucider ces remarques, considérons l'exemple simple d'un seul modèle linéaire (A, B, C, D) d'ordre n, et supposons $D = 0$ pour des raisons de simplicité. Si on excite ce modèle avec un signal d'entrée quelconque $\{u(t) : t \in \mathbb{N}\}$ en partant d'un état initial $x(0)$, alors l'équation entrée-sortie de ce modèle est donnée par

$$y(t) = CA^t x(0) + CA^{t-1}Bu(1) + \cdots + CABu(t-2) + CBu(t-1). \tag{4.5}$$

Considérons un certain état discret $\lambda : \mathbb{N} \to \{1, \ldots, \bar{s}\}$, $t \longmapsto \lambda_t$, avec $\bar{s}$ un entier quelconque, potentiellement infini. Soit $\{T_{\lambda_t}\}$ une séquence quelconque de matrices non singulières indexées par λ. Alors on peut écrire l'ensemble des relations suivantes

$$\begin{array}{rcl}
CA^t x(0) & = & (CT_{\lambda_t}^{-1}) \cdot (T_{\lambda_t} A T_{\lambda_{t-1}}^{-1}) \cdots (T_{\lambda_1} A T_{\lambda_0}^{-1}) \cdot (T_{\lambda_0} x(0)) \\
CA^{t-1}Bu(1) & = & (CT_{\lambda_t}^{-1}) \cdot (T_{\lambda_t} A T_{\lambda_{t-1}}^{-1}) \cdots (T_{\lambda_2} A T_{\lambda_1}^{-1}) \cdot (T_{\lambda_1} B) u(1) \\
\vdots & = & \vdots \quad \cdots \quad \vdots \quad \cdots \\
CABu(t-2) & = & (CT_{\lambda_t}^{-1}) \cdot (T_{\lambda_t} A T_{\lambda_{t-1}}^{-1}) \cdot (T_{\lambda_{t-1}} B) u(t-2) \\
CBu(t-1) & = & (CT_{\lambda_t}^{-1}) \cdot (T_{\lambda_t} B) u(t-1).
\end{array} \tag{4.6}$$

En posant maintenant $A_{\lambda_t} = T_{\lambda_{t+1}} A T_{\lambda_t}^{-1}$, $B_{\lambda_t} = T_{\lambda_{t+1}} B$ et $C_{\lambda_t} = CT_{\lambda_t}^{-1}$, nous pouvons voir que le modèle commutant $\{A_{\lambda_t}, B_{\lambda_t}, C_{\lambda_t}, D_{\lambda_t}\}_{\lambda_t \in \{1,\ldots,\bar{s}\}}$ réalise le comportement (4.5). Clairement, le nombre $\bar{s}$ peut être choisi de façon arbitraire. Ainsi, il existe une infinité de modèles (selon le choix de $\{T_{\lambda_t}\}$) commutant entre un nombre arbitraire $\bar{s}$ de sous-modèles linéaires qui réalisent le même comportement entrée-sortie que le modèle linéaire (4.5). Remarquons de plus que, étant donné $\bar{s}$, la séquence d'états discrets (c'est-à-dire la succession des commutations) peut être choisie de façon arbitraire.

1. Nous illustrons cela tout au long de ce chapitre.

D'une manière similaire, on peut choisir l'ordre de ces sous-modèles de façon arbitraire. Pour le voir, il faut se rappeler que le modèle linéaire (4.5) peut être décrit avec un autre modèle linéaire d'ordre $\bar{n} > n$ arbitrairement grand. En opérant une transformation similaire à (4.6) sur ce dernier modèle, on peut alors montrer que le comportement entrée-sortie est réalisable avec un nombre $\bar{s}$ de sous-modèles d'ordre $\bar{n}$ tous choisis de façon arbitraires.

En définitive, il existe trois principaux facteurs d'indétermination ou d'ambiguïté du modèle commutant à partir des données :

1. la séquence et le nombre d'états discrets,
2. l'ordre du modèle ou la dimension de l'espace d'état,
3. la multiplicité des bases de coordonnées possibles.

Dans le cas des modèles linéaires (un seul sous-modèle), l'ambiguïté sur l'ordre n peut être éliminée en imposant au modèle à identifier la contrainte d'être minimal, c'est-à-dire observable et atteignable [69]. De cette façon, il ne reste plus que le problème de la multiplicité des bases de représentation de l'état dont le procédé d'identification peut facilement s'accommoder en spécifiant arbitrairement une base de coordonnées [10].
Dans le cas des modèles commutants, il faudrait, pour lever (réduire) l'indétermination du modèle (4.1), imposer des contraintes (des bornes supérieures par exemple [128]) à la fois sur le nombre de sous-modèles et sur l'ordre du modèle. Cela reviendrait comme dans le cas des modèles linéaires, à imposer une sorte de contrainte de minimalité au modèle à estimer. Malheureusement, la notion de modèle minimal pour les modèles hybrides n'est à notre connaissance, pas encore clairement formalisée dans la littérature existante [95]. Néanmoins, nous savons que d'une manière classique, une contrainte inspirée par la théorie de systèmes linéaires et contribuant à la réduction de l'ordre du modèle est la notion d'observabilité.

4.2.3 Observabilité d'un modèle d'état commutant

L'observabilité d'un modèle commutant fait généralement référence à la propriété intrinsèque que posséderait ce modèle à permettre d'inférer uniquement son état $(\lambda_t, x(t))$ à partir des données entrée-sortie observées sur un certain horizon de temps. Selon l'horizon d'observation considéré ou selon que l'on souhaite observer l'état discret λ_t et/ou l'état continu $x(t)$, plusieurs notions d'observabilité sont définies dans la littérature. L'objectif ici n'est pas d'étudier en détail le problème de l'observabilité des systèmes commutants. Le lecteur intéressé par cet aspect pourra se référer par exemple à [128, 6, 44]. Notre démarche vise plutôt à introduire quelques propriétés de base du modèle (4.1), nécessaires pour envisager son estimation.
Pour commencer, considérons une séquence d'états discrets (ou un chemin [6]) $\bar{\lambda}_{t,f} = \lambda_t \cdots \lambda_{t+f-1} \in \mathcal{S}^f = \mathcal{S} \times \cdots \times \mathcal{S}$, avec f un entier quelconque. Nous définissons le long de la séquence $\bar{\lambda}_{t,f}$, la matrice d'observabilité $\Gamma(\bar{\lambda}_{t,f})$ et la matrice $H(\bar{\lambda}_{t,f})$ (contenant les paramètres de Markov généralisés du système hybride) :

$$\Gamma(\bar{\lambda}_{t,f}) = \begin{bmatrix} C_{\lambda_t} \\ C_{\lambda_{t+1}}A_{\lambda_t} \\ C_{\lambda_{t+2}}A_{\lambda_{t+1}}A_{\lambda_t} \\ \vdots \\ C_{\lambda_{t+f-1}}A_{\lambda_{t+f-2}}\cdots A_{\lambda_t} \end{bmatrix}, \tag{4.7}$$

$$H(\bar{\lambda}_{t,f}) = \begin{bmatrix} D_{\lambda_t} & 0 & 0 & \dots & 0 \\ C_{\lambda_{t+1}}B_{\lambda_t} & D_{\lambda_{t+1}} & 0 & \dots & 0 \\ C_{\lambda_{t+2}}A_{\lambda_{t+1}}B_{\lambda_t} & C_{\lambda_{t+2}}B_{\lambda_{t+1}} & D_{\lambda_{t+2}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{\lambda_{t+f-1}}A_{\lambda_{t+f-2}}\dots A_{\lambda_{t+1}}B_{\lambda_t} & C_{\lambda_{t+f-1}}A_{\lambda_{t+f-2}}\dots A_{\lambda_{t+2}}B_{\lambda_{t+1}} & \dots & \dots & D_{\lambda_{t+f-1}} \end{bmatrix}. \tag{4.8}$$

Selon les valeurs prises par la séquence $\bar{\lambda}_{t,f} = \lambda_t \cdots \lambda_{t+f-1}$, les matrices définies en (4.7)-(4.8) peuvent contenir des paramètres de un ou plusieurs sous-modèles différents. Dans le premier cas, nous dirons que ces matrices sont « pures » et dans le deuxième cas qu'elles sont « mixtes ». Si nous définissons maintenant les vecteurs

$$\begin{aligned} y_f(t) &= \begin{bmatrix} y(t)^\top & \cdots & y(t+f-1)^\top \end{bmatrix}^\top \in \mathbb{R}^{fn_y}, \\ u_f(t) &= \begin{bmatrix} u(t)^\top & \cdots & u(t+f-1)^\top \end{bmatrix}^\top \in \mathbb{R}^{fn_u}, \end{aligned} \tag{4.9}$$

alors, pour tout $t \geq 0$, il est facile d'obtenir à partir des équations (4.1) du système,

$$y_f(t) = \Gamma\left(\bar{\lambda}_{t,f}\right) x(t) + H\left(\bar{\lambda}_{t,f}\right) u_f(t). \tag{4.10}$$

L'observabilité du système (4.1) (dans le sens qui nous intéresse ici) fait référence à la possibilité de déterminer l'état continu $x(t)$ de façon unique en fonction des vecteurs $u_f(t)$ et $y_f(t)$ pour tout $t \geq 0$, où f est un certain entier fixé. Au vu de l'équation (4.10), une telle propriété dépend exclusivement des propriétés de rang de la matrice $\Gamma\left(\bar{\lambda}_{t,f}\right)$. Plus précisément, nous pouvons formuler la définition suivante [6].

Définition 4.1. *Le système linéaire à commutations (4.1) est dit observable par chemin s'il existe un entier m tel que pour tout instant t,* $\text{rang}(\Gamma(\bar{\lambda}_{t,m})) = n$. *Le plus petit des entiers m vérifiant cette propriété est appelé indice d'observabilité par chemin.*

Dans tout le reste du chapitre, nous noterons $\nu = \min\left\{m : \text{rang}(\Gamma(\bar{\lambda}_{t,m})) = n\ \forall t\right\}$ l'indice d'observabilité par chemin et supposerons que $\nu \leq M_\nu$, où M_ν est un nombre fini qui n'est pas très grand. Cette hypothèse, comme nous le verrons dans la suite, nous permettra de limiter théoriquement la complexité des méthodes développées.
De la définition 4.1, nous pouvons obtenir le résultat suivant.

Theorème 4.1 (Observabilité de systèmes MISO commutants)**.** *Supposons que le système (4.1) soit un MISO, c'est-à-dire $n_y = 1$. Alors (4.1) est observable par chemin avec un indice d'observabilité $\nu = n$ si*

et seulement s'il existe une séquence de matrices non singulières $\{T_t\}$ *dans* $\mathbb{R}^{n\times n}$ *telle que pour* t,

$$T_{t+1}A_{\lambda_t}T_t^{-1} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a_t^0 & -a_t^1 & \cdots & -a_t^{n-1} \end{bmatrix} \tag{4.11}$$

$$C_{\lambda_t}T_t^{-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}. \tag{4.12}$$

Preuve. Nécessité : Supposons que le système (4.1) soit observable par chemin et que $n_y = 1$. Pour plus de clarté, désignons C_{λ_t} par $c_{\lambda_t}^\top$ pour insister sur le fait que C_{λ_t} est un vecteur ligne. Alors, notons d'une part que

$$\begin{aligned}\Gamma(\bar{\lambda}_{t,n+1})\Gamma(\bar{\lambda}_{t,n})^{-1} &= \begin{bmatrix} c_{\lambda_t}^\top \\ c_{\lambda_{t+1}}^\top A_{\lambda_t} \\ c_{\lambda_{t+2}}^\top A_{\lambda_{t+1}}A_{\lambda_t} \\ \vdots \\ c_{\lambda_{t+n-1}}^\top A_{\lambda_{t+n-2}}\cdots A_{\lambda_t} \\ c_{\lambda_{t+n}}^\top A_{\lambda_{t+n-1}}\cdots A_{\lambda_t} \end{bmatrix}\Gamma(\bar{\lambda}_{t,n})^{-1} \\ &= \begin{bmatrix} \Gamma(\bar{\lambda}_{t,n}) \\ c_{\lambda_{t+n}}^\top A_{\lambda_{t+n-1}}\cdots A_{\lambda_t} \end{bmatrix}\Gamma(\bar{\lambda}_{t,n})^{-1}, \\ &= \begin{bmatrix} I_n \\ c_{\lambda_{t+n}}^\top A_{\lambda_{t+n-1}}\cdots A_{\lambda_t}\Gamma(\bar{\lambda}_{t,n})^{-1} \end{bmatrix}\end{aligned}$$

et d'autre part que

$$\Gamma(\bar{\lambda}_{t,n+1})\Gamma(\bar{\lambda}_{t,n})^{-1} = \begin{bmatrix} c_{\lambda_t}^\top\Gamma(\bar{\lambda}_{t,n})^{-1} \\ \Gamma(\bar{\lambda}_{t+1,n})A_{\lambda_t}\Gamma(\bar{\lambda}_{t,n})^{-1} \end{bmatrix}.$$

Ainsi en posant $T_t = \Gamma(\bar{\lambda}_{t,n})$ et $T_{t+1} = \Gamma(\bar{\lambda}_{t+1,n})$, il apparaît clairement, en égalant les deux expressions précédentes, que $c_{\lambda_t}^\top T_t^{-1}$ et $T_{t+1}A_{\lambda_t}T_t^{-1}$ prennent les formes (4.12) et (4.11).

Suffisance : Supposons maintenant qu'il existe une suite $\{T_t\}$ de matrices non singulières telles que $T_{t+1}A_{\lambda_t}T_t^{-1}$ et $c_{\lambda_t}^\top T_t^{-1}$ s'écrivent sous la forme (4.11) et (4.12). Alors il est facile de vérifier que pour tout t, on a $\Gamma(\bar{\lambda}_{t,n}) = T_t$ et donc $\text{rang}\left(\Gamma(\bar{\lambda}_{t,n})\right) = n$ pour tout t, d'où on conclut que le système est observable par chemin. ■

Cette proposition, à condition que l'on ajuste l'indexation des matrices A et C, peut s'appliquer aussi aux systèmes à temps variant (LTV) ou aux systèmes à paramètres variants (LPV) [117]. Dans un objectif de vérification algébrique de l'observabilité d'un système du type (4.1) telle que caractérisée par le Théorème 4.1, il n'est pas nécessaire de prouver l'existence d'une séquence infinie de matrices $\{T_t\}$ (indexée par le temps) qui vérifie les conditions (4.11) et (4.12) du Théorème 4.1. Puisque λ_t vit dans un ensemble fini $\mathcal{S}$ de cardinal s, il suffit de trouver un nombre s de matrices non singulières $\{T_{\lambda_t}\}$ (indexées par l'état discret) telles que $\bar{c}_{\lambda_t}^\top = c_{\lambda_t}^\top T_{\lambda_t}^{-1}$ et $\bar{A}_{\lambda_t} = T_{\lambda_{t+1}}A_{\lambda_t}T_{\lambda_t}^{-1}$ puissent prendre les formes (4.11) et (4.12).

Afin de pouvoir généraliser le Théorème 4.1 à des systèmes MIMO, supposons qu'il existe un ensemble $\{\gamma_j\}_{j=1}^{s} \subset \mathbb{R}^{n_y}$ de vecteurs tels que pour toute séquence $\bar{\lambda}_{t,n}$, $\text{rang}(\Psi(\bar{\lambda}_{t,n})) = \text{rang}(\Gamma(\bar{\lambda}_{t,n}))$, où

$$\Psi(\bar{\lambda}_{t,n}) = \begin{bmatrix} \bar{c}_{\lambda_t}^\top \\ \bar{c}_{\lambda_{t+1}}^\top A_{\lambda_t} \\ \vdots \\ \bar{c}_{\lambda_{t+n-1}}^\top A_{\lambda_{t+n-2}} \cdots A_{\lambda_t} \end{bmatrix} = \begin{bmatrix} \gamma_{\lambda_t}^\top & & \\ & \ddots & \\ & & \gamma_{\lambda_{t+n-1}}^\top \end{bmatrix} \Gamma(\bar{\lambda}_{t,n}) \in \mathbb{R}^{n \times n},$$

avec $\bar{c}_{\lambda_t}^\top = \gamma_{\lambda_t}^\top C_{\lambda_t}$. Pour un système MIMO satisfaisant à cette hypothèse, l'observabilité par chemin est vérifiée avec un indice d'observabilité $\nu = n$ si le système MISO commutant $\left\{A_j, \bar{c}_j^\top\right\}_{j=1}^{s}$ est observable par chemin, c'est-à-dire si et seulement si A_{λ_t} et $\bar{c}_{\lambda_t}^\top$ peuvent se mettre sous les formes (4.11) et (4.12) pour tout λ_t.

En général l'observabilité individuelle des sous-modèles constitutifs du modèle commutant n'implique pas l'observabilité du système hybride global [16]. Cependant, lorsqu'un temps de séjour minimum supérieur à $2n$ est respecté, on a le résultat suivant.

Theorème 4.2. *Considérons un système MIMO commutant représenté par un modèle du type (4.1). Supposons que*

- *chaque sous-modèle est observable c'est-à-dire que,* $\text{rang}\left(\Gamma_n(A_j, C_j)\right) = n$ *pour tout* $j \in \mathcal{S}$*, avec* $\Gamma_n(A_j, C_j)$ *défini comme au Chapitre 3.*
- $\text{rang}(A_j) = n$ *pour tout* $j \in \mathcal{S}$*,*
- *les instants de commutation sont séparés par un temps de séjour* $\tau_{dwell} \geq 2n$*.*

Alors, le système (4.1) est observable par chemin avec un indice d'observabilité $\nu \leq 2n$ *au sens de la définition 4.1.*

Preuve. Considérons deux états discrets quelconques $i, j \in \mathcal{S}$. Puisque $\tau_{\text{dwell}} \geq 2n$ la matrice $\Gamma(\bar{\lambda}_{t,2n})$ prend, pour tout t, l'une des formes suivantes.

$$\begin{bmatrix} C_i \\ C_iA_i \\ \vdots \\ C_iA_i^{2n-2} \\ C_iA_i^{2n-1} \end{bmatrix} \to \begin{bmatrix} C_i \\ C_iA_i \\ \vdots \\ C_iA_i^{2n-2} \\ \boldsymbol{C_jA_i^{2n-1}} \end{bmatrix} \to \begin{bmatrix} C_i \\ C_iA_i \\ \vdots \\ \boldsymbol{C_jA_i^{2n-2}} \\ \boldsymbol{C_jA_jA_i^{2n-2}} \end{bmatrix} \to \cdots \to \begin{bmatrix} C_i \\ \boldsymbol{C_jA_i} \\ \vdots \\ \boldsymbol{C_jA_j^{2n-3}A_i} \\ \boldsymbol{C_jA_j^{2n-2}A_i} \end{bmatrix} \to \begin{bmatrix} \boldsymbol{C_j} \\ \boldsymbol{C_jA_j} \\ \vdots \\ \boldsymbol{C_jA_j^{2n-2}} \\ \boldsymbol{C_jA_j^{2n-1}} \end{bmatrix}.$$

Remarquons alors que les matrices d'observabilité données ci-dessus contiennent toujours au moins une sous-matrice de la forme $\Gamma_n(A_i, C_i)$, $\Gamma_n(A_j, C_j)$, ou $\Gamma_n(A_j, C_j)A_j^pA_i^q$, avec p et q des entiers. Alors, puisque pour tout j, $\text{rang}(A_j) = n$, les matrices ci-dessus sont toutes de rang plein colonne pour tous i et j, et donc $\text{rang}\left(\Gamma(\bar{\lambda}_{t,2n})\right) = n$

pour tout t. ■

En résumé, nous avons vu que le modèle (4.1) n'est pas défini de façon unique par une réalisation donnée des ses entrées et de ses sorties. Cela signifie que ni l'ordre n, ni le nombre de sous-modèles s, ni les matrices $\{A_j, B_j, C_j, D_j\}_{j=1}^{s}$ ne peuvent être inférés de façon unique à partir d'échantillons d'entrées-sorties. Puisque plusieurs solutions sont possibles pour le problème 4.1, il est nécessaire d'imposer quelques contraintes sur la solution recherchée. Une de ces contraintes est l'observabilité qui a pour effet de réduire la dimension de l'espace d'état du modèle (donc partiellement sa complexité) à estimer sans pour autant garantir l'unicité du modèle. Nous avons ainsi introduit la notion d'observabilité complète et présenté quelques conditions suffisantes à la réalisation de cette contrainte d'observabilité. Dans la suite, nous nous intéressons au problème général d'identification de modèles commutants.

4.3 Commutations sans temps de séjour

De l'étude préliminaire réalisée dans les sections précédentes, nous allons, dans cette sous-section, procéder à l'identification du modèle (4.1) à partir d'un nombre fini d'observations. Nous étudions d'abord le cas général où aucune hypothèse de temps de séjour n'est formulée concernant l'occurrence des commutations.
Si l'état continu était connu, le problème d'identification se réduirait à l'extraction des paramètres et de l'état discret de la relation (4.2). Cela pourrait être réalisé en utilisant par exemple l'algorithme Generalized Principal Component Analysis (GPCA) [129]. Malheureusement, dans bon nombre de situations pratiques, l'état $x(t)$ n'est pas accessible à la mesure et devra donc être estimé grâce aux observations entrée-sortie. En conséquence, une première étape consiste à éliminer l'état continu $x(t)$ dans les équations du système. L'expression obtenue pour l'état nous permet de discuter dans les sous-sections 4.3.1, 4.3.2, 4.3.3 de différentes structures intermédiaires de modèles entrée-sortie pouvant être estimées en vue de recouvrer ensuite un modèle d'état.

4.3.1 Du modèle d'état au modèle entrée-sortie

Considérons le système linéaire commutant (4.1) et voyons comment nous pouvons en dériver une relation entrée-sortie. Dans cet objectif, soit $f \geq \nu$, où ν est l'indice d'observabilité par chemin du système et soit $\bar{\lambda}_{t,f} \in \mathcal{S}^f$ une séquence d'états discrets. Définissons les matrices

$$\mathscr{A}(\bar{\lambda}_{t,f}) = A_{\lambda_{t+f-1}} A_{\lambda_{t+f-2}} \cdots A_{\lambda_t},$$
$$\Delta(\bar{\lambda}_{t,f}) = \begin{bmatrix} \mathscr{A}(\bar{\lambda}_{t+1,f-1}) B_{\lambda_t} & \mathscr{A}(\bar{\lambda}_{t+2,f-2}) B_{\lambda_{t+1}} & \cdots & \mathscr{A}(\bar{\lambda}_{t+f-1,1}) B_{\lambda_{t+f-2}} & B_{\lambda_{t+f-1}} \end{bmatrix}.$$

Afin de pouvoir exprimer l'état continu du système sur chaque fenêtre temporelle du type $[t-f,\, t]$, $t \geq f$, considérons l'hypothèse suivante.

Hypothèse 4.1. *Le système commutant (4.1) est observable par chemin, c'est-à-dire qu'il existe $\nu \leq M_\nu$ tel que*

$$\text{rang}(\Gamma(\bar{j})) = n \quad \forall\, \bar{j} \in \mathscr{D}_\nu \subset \mathcal{S}^\nu,$$

où $\mathscr{D}_\nu = \left\{\bar{\lambda}_{t,\nu} \in \mathcal{S}^\nu / t = 0, \ldots, N-\nu\right\}$ est l'ensemble des séquences d'états discrets $\bar{\lambda}_{t,\nu}$ de longueur ν qui sont réalisables par le système.

En supposant que cette hypothèse est réalisée, on a $\text{rang}(\Gamma(\bar{\lambda}_{t,f}) = n$ pour tout $f \geq \nu$ et pour tout $\bar{\lambda}_{t,f} \in \mathscr{D}_f$. A partir des équations du système (4.1) on peut obtenir par substitutions successives

$$x(t) = \mathscr{A}\left(\bar{\lambda}_{t-f,f}\right) x(t-f) + \Delta\left(\bar{\lambda}_{t-f,f}\right) u_f(t-f) \tag{4.13}$$

et

$$y_f(t-f) = \Gamma\left(\bar{\lambda}_{t-f,f}\right) x(t-f) + H\left(\bar{\lambda}_{t-f,f}\right) u_f(t-f). \tag{4.14}$$

Alors, sous l'hypothèse que le système considéré est observable par chemin (Hypothèse 4.1), on peut tirer $x(t-f)$ de l'équation (4.14). En reportant l'expression ainsi obtenue dans (4.13), l'état continu du système à l'instant t peut être exprimé comme

$$x(t) = \Omega\left(\bar{\lambda}_{t-f,f}\right) z_f(t), \tag{4.15}$$

avec [1]

$$z_f(t) = \begin{bmatrix} y_f(t-f)^\top & u_f(t-f)^\top \end{bmatrix}^\top \in \mathbb{R}^{f(n_y+n_u)},$$
$$\Omega\left(\bar{\lambda}_{t-f,f}\right) = \left[\mathscr{A}\left(\bar{\lambda}_{t-f,f}\right)\Gamma\left(\bar{\lambda}_{t-f,f}\right)^\dagger, \quad \Delta\left(\bar{\lambda}_{t-f,f}\right) - \mathscr{A}\left(\bar{\lambda}_{t-f,f}\right)\Gamma\left(\bar{\lambda}_{t-f,f}\right)^\dagger H\left(\bar{\lambda}_{t-f,f}\right)\right],$$

$\Omega\left(\bar{\lambda}_{t-f,f}\right) \in \mathbb{R}^{n \times f(n_y+n_u)}$. Ainsi, l'état continu du système hybride peut s'exprimer comme une certaine combinaison (incluant les séquences de commutations) des données relatives au passé du système sur une fenêtre temporelle finie. Pour des raisons de minimalité de l'état $x(t)$, nous pouvons aussi supposer que $\Delta\left(\bar{\lambda}_{t-f,f}\right)$ est de rang plein ligne pour tout t, c'est-à-dire que le système est complètement atteignable. Il est facile de voir que cela implique que $\text{rang}\left(\Omega\left(\bar{\lambda}_{t-f,f}\right)\right) = n$, pour tout $\bar{\lambda}_{t-f,f} \in \mathscr{D}_f$, avec $f \geq \nu$.
Si on remplace maintenant l'expression (4.15) de l'état continu dans la seconde équation de (4.1), on obtient

$$\begin{aligned} y(t) &= C_{\lambda_t}\Omega\left(\bar{\lambda}_{t-f,f}\right) z_f(t) + D_{\lambda_t} u(t), \\ &= L\left(\bar{\lambda}_{t-f,f+1}\right)\begin{bmatrix} z_f(t) \\ u(t) \end{bmatrix}, \end{aligned} \tag{4.16}$$

où $\bar{\lambda}_{t-f,f+1} = \bar{\lambda}_{t-f,f}\lambda_t \in \mathscr{D}_{f+1} = \mathscr{D}_f \times \mathscr{D}_1$ et $L\left(\bar{\lambda}_{t-f,f+1}\right) = \begin{bmatrix} C_{\lambda_t}\Omega\left(\bar{\lambda}_{t-f,f}\right) & D_{\lambda_t} \end{bmatrix}$. Si nous notons $|\mathscr{D}_{f+1}| \leq s^{f+1}$ le cardinal de $\mathscr{D}_{f+1}$, et définissons un nouvel état discret $\rho : \mathbb{Z} \to \{1, \ldots, |\mathscr{D}_{f+1}|\}$,

1. $M^\dagger$ désignant la pseudo-inverse de Moore-Penrose de M qui s'exprime ici par $M^\dagger = \left(M^\top M\right)^{-1} M^\top$.

$t \longmapsto \rho_t$, alors l'équation (4.16) peut encore se lire de la façon suivante : pour tout $t > f$, il existe $\rho_t \in \{1, \ldots, |\mathscr{D}_{f+1}|\}$ tel que

$$y(t) = L_{\rho_t} \begin{bmatrix} z_f(t) \\ u(t) \end{bmatrix}, \tag{4.17}$$

où $L_{\rho_t} = L\big(\bar{\lambda}_{t-f,f+1}\big) \in \mathbb{R}^{n_y \times (fn_y + (f+1)n_u)}$ est utilisé avec un petit abus de notation. De l'équation (4.16), on peut formuler

Proposition 4.1. *Si un système commutant du type (4.1) avec un ordre n et un nombre de sous-modèles s est observable par chemin avec un indice d'observabilité ν, alors il peut être converti en un système ARX commutant entre au plus $s^{\nu+1}$ sous-modèles distincts.*

La proposition 4.1 montre que, contrairement au cas des systèmes linéaires, il n'y a en général pas d'équivalence directe entre les modèles d'état et les modèles entrée-sortie ARX de systèmes à commutations et que la conversion d'une forme de modèle en une autre peut entraîner une variation du nombres d'états discrets. Mais il peut arriver dans quelques cas particuliers qu'il y ait autant de sous-modèles dans le modèle (4.17) que dans (4.1) [103], [39].
Nous donnons ci-dessous deux exemples de systèmes SLC pour lesquels les modèles (4.1) et (4.17) ont le même nombre d'états discrets.

Exemple 4.1 (Système MISO)**.** *Pour un système MISO tel que pour tout λ_t, il existe une matrice non singulière T_o vérifiant*

$$\begin{aligned}
T_o A_{\lambda_t} T_o^{-1} &= \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a^0_{\lambda_t} & -a^1_{\lambda_t} & \cdots & -a^{n-1}_{\lambda_t} \end{bmatrix}, \\
C_{\lambda_t} T_o^{-1} &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \\
T_o B_{\lambda_t} &= Cte, \\
D_{\lambda_t} &= Cte,
\end{aligned}$$

la matrice $L\big(\bar{\lambda}_{t-n,n+1}\big)$ ne dépend que de λ_{t-n}. Pour le voir, remplaçons simplement A_{λ_t}, B_{λ_t}, C_{λ_t} respectivement par $T_o A_{\lambda_t} T_o^{-1}$, $T_o B_{\lambda_t} = Cte = B$ et $C_{\lambda_t} T_o^{-1} = e_1^\top$ définies par les expressions ci-dessus, où e_j est le vecteur de $\mathbb{R}^n$ possédant 1 à la $j$ème entrée et 0 partout ailleurs. Calculons ensuite $L\big(\bar{\lambda}_{t-n,n+1}\big)$, grâce à son expression (4.16). Dans ce but, remarquons que pour tout $\bar{\lambda}_{t,n}$, $\Gamma(\bar{\lambda}_{t,n}) = I_n$ et $H(\bar{\lambda}_{t,n}) = Cte$. On a donc $\nu = n$. De plus on a

$$\begin{aligned}
C_{\lambda_t} \mathscr{A}(\bar{\lambda}_{t-n,n}) &= \begin{bmatrix} -a^0_{\lambda_{t-n}} & -a^1_{\lambda_{t-n}} & \cdots & -a^{n-1}_{\lambda_{t-n}} \end{bmatrix} \\
C_{\lambda_t} \Delta(\bar{\lambda}_{t-n,n}) &= \begin{bmatrix} e_n^\top B & e_{n-1}^\top B & \cdots & e_1^\top B \end{bmatrix} = Cte.
\end{aligned}$$

En portant ces valeurs dans l'expression de $L\big(\bar{\lambda}_{t-n,n+1}\big)$*, on peut voir que* $L\big(\bar{\lambda}_{t-n,n+1}\big)$ *ne dépend que de* λ_{t-n} *et donc le modèle (4.17) possède le même nombre d'états discrets que le modèle initial (4.1).*

Exemple 4.2. *Si pour tout* $j \in \mathcal{S}$, $\text{rang}(C_j) = n$*, alors* $\nu = 1$*. De la relation* $y(t) = C_{\lambda_t}x(t) + D_{\lambda_t}u(t)$*, on peut alors exprimer* $x(t)$ *comme* $x(t) = C^{\dagger}_{\lambda_t}\big(y(t) - D_{\lambda_t}u(t)\big)$ *et finalement obtenir un modèle entrée-sortie du type (4.17) comme* $y(t) = C_{\lambda_t}C^{\dagger}_{\lambda_t}y(t) + (I_{n_y} - C_{\lambda_t}C^{\dagger}_{\lambda_t})D_{\lambda_t}u(t)$*. Ainsi ce modèle possède le même nombre d'états discrets que le modèle (4.1).*

Dans le but d'obtenir un modèle d'état du type (4.1), nous pouvons, en posant $f = \nu$ (supposé connu), envisager par exemple l'identification du modèle hybride entrée-sortie (4.17) qui est composé de $|\mathscr{D}_{f+1}| \leq s^{\nu+1}$ sous-modèles. Une fois ce dernier modèle connu, nous pouvons le reconvertir en un modèle d'état non minimal. Cela peut se faire en procédant comme dans [132], c'est-à-dire en définissant simplement un état continu de dimension $\nu(n_u + n_y)$

$$x(t) = \begin{bmatrix} y_\nu(t-\nu) \\ u_\nu(t-\nu) \end{bmatrix} \in \mathbb{R}^{\nu(n_u+n_y)}.$$

Alors, une réalisation $(\mathcal{A}_{\rho_t}, \mathcal{B}_{\rho_t}, \mathcal{C}_{\rho_t}, \mathcal{D}_{\rho_t})$ d'état du système (4.1) peut être obtenue comme

$$\mathcal{A}_{\rho_t} = \left[\begin{array}{c|c} \begin{bmatrix} \mathrm{O}_{(\nu-1)n_y,n_y} & I_{(\nu-1)n_y} \end{bmatrix} & \mathrm{O}_{\nu n_u} \\ L^y_{\rho_t} & L^u_{\rho_t} \\ \mathrm{O}_{(\nu-1)n_u,\nu n_y} & \begin{bmatrix} \mathrm{O}_{(\nu-1)n_u,n_u} & I_{(\nu-1)n_u} \end{bmatrix} \\ \mathrm{O}_{n_u,\nu n_y} & \mathrm{O}_{n_u,\nu n_u} \end{array}\right],$$

$$\mathcal{B}_{\rho_t} = \begin{bmatrix} \mathrm{O}_{(\nu-1)n_y,n_u} \\ L^o_{\rho_t} \\ \mathrm{O}_{(\nu-1)n_u,n_u} \\ I_{n_u} \end{bmatrix},$$

$$\mathcal{C}_{\rho_t} = \begin{bmatrix} L^y_{\rho_t} & L^u_{\rho_t} \end{bmatrix},$$

$$\mathcal{D}_{\rho_t} = L^o_{\rho_t},$$

où $L_{\rho_t} = \begin{bmatrix} L^y_{\rho_t} & L^u_{\rho_t} & L^o_{\rho_t} \end{bmatrix}$ est une partition de L_{ρ_t} telle que $L^y_{\rho_t}$, $L^u_{\rho_t}$, $L^o_{\rho_t}$ contiennent respectivement νn_y, νn_u et n_u colonnes de L_{ρ_t}. Notons que $L^o_{\rho_t} = D_{\lambda_t}$. Cependant, l'ordre de cette réalisation et le nombre d'états discrets valent respectivement $\nu(n_u + n_y) \geq n$ et $|\mathscr{D}_{f+1}| \geq s$ et sont supérieures à l'ordre n et au nombre s d'états discrets du modèle qui a généré les données. Ce cas illustre bien la discussion de la section 4.2. D'une façon similaire au cas des modèles linéaires, on pourrait maintenant envisager une procédure de réduction de modèle afin de minimiser les dimensions de la réalisation $(\mathcal{A}_{\rho_t}, \mathcal{B}_{\rho_t}, \mathcal{C}_{\rho_t}, \mathcal{D}_{\rho_t})$ ci-dessus. Mais une telle procédure nécessiterait au préalable, l'établissement de résultats d'équivalence entrée-sortie entre différents modèles d'état en théorie de la réalisation de systèmes commutants. Ce problème n'est pas traité dans le cadre de ce manuscrit mais pourra faire l'objet de futures recherches.

4.3.2 Application directe des méthodes des sous-espaces

Dans cette sous-section, nous étudions dans le cas général, le problème de l'identification d'un système commutant du type (4.1). Bien que cette étude n'aboutit pas, sous les hypothèses génériques considérées, à un algorithme d'une grande portée pratique, elle nous permettra notamment de mettre en évidence la complexité d'un tel problème.

Nous allons tenter d'appliquer la méthodologie générale des méthodes des sous-espaces. Dans cet objectif, écrivons l'équation étendue du système sur un intervalle de la forme $[t,\, t+f-1]$:

$$y_f(t) = \Gamma\left(\bar{\lambda}_{t,f}\right) x(t) + H\left(\bar{\lambda}_{t,f}\right) u_f(t). \tag{4.18}$$

En portant ensuite l'expression (4.15) de $x(t)$ dans cette dernière équation, il vient pour tout $t > f$

$$\begin{aligned} y_f(t) &= \begin{bmatrix}\Gamma\left(\bar{\lambda}_{t,f}\right)\Omega\left(\bar{\lambda}_{t-f,f}\right) & H\left(\bar{\lambda}_{t,f}\right)\end{bmatrix}\begin{bmatrix} z_f(t) \\ u_f(t)\end{bmatrix} \\ &= F(\bar{\lambda}_{t-f,2f})\begin{bmatrix} z_f(t) \\ u_f(t)\end{bmatrix}, \end{aligned} \tag{4.19}$$

avec $F(\bar{\lambda}_{t-f,2f}) = \begin{bmatrix}\Gamma\left(\bar{\lambda}_{t,f}\right)\Omega\left(\bar{\lambda}_{t-f,f}\right) & H\left(\bar{\lambda}_{t,f}\right)\end{bmatrix} \in \mathbb{R}^{fn_y \times f(n_y+2n_u)}$. Il est important de remarquer la similarité de l'équation (4.19) avec celle qu'on pourrait obtenir à partir par exemple d'un système linéaire à temps variant ou un système à paramètres variants. Cependant, une différence fondamentale est que le nombre de matrices de paramètres du type $F(\bar{\lambda}_{t-f,2f})$, $\bar{\lambda}_{t-f,2f} \in \mathscr{D}_{2f}$, dans (4.19) est fini tandis que dans les deux autres cas mentionnés, ce nombre pourrait être infini. Notons $\bar{s} = |\mathscr{D}_{2f}| \leq s^{2f}$ le cardinal de $\mathscr{D}_{2f} = \mathscr{D}_f \times \mathscr{D}_f$. Du fait de l'équipotence[1] entre $\mathscr{D}_{2f}$ et $\{1,\ldots,\bar{s}\}$, on peut utiliser plutôt $\{1,\ldots,\bar{s}\}$ comme ensemble d'indexation des matrices $F(\bar{\lambda}_{t-f,2f})$. Cela revient en fait à définir un nouvel état discret par l'application qui, à $t \in \mathbb{Z}$, associe $\sigma_t \in \{1,\ldots,\bar{s}\}$. Avec un petit abus de notations, nous notons $F_{\sigma_t} = F(\bar{\lambda}_{t-f,2f})$. Alors, l'équation (4.19) peut se lire de la façon suivante : pour tout $t > f$, $\exists \sigma_t \in \{1,\ldots,\bar{s}\}$, tel que

$$y_f(t) = F_{\sigma_t}\begin{bmatrix} z_f(t) \\ u_f(t)\end{bmatrix}. \tag{4.20}$$

De cette relation, il est théoriquement possible d'estimer les paramètres F_{σ_t} et d'en déduire ensuite les matrices du système (4.1) en se basant sur le concept des méthodes des sous-espaces [115]. Mais l'intérêt pratique d'une telle méthode est franchement limité par quelques difficultés :

- Le nombre potentiel $\bar{s} \leq s^{2f}$ de sous-modèles du système (4.20) est très important.
- En fonction des commutations acceptées par le système original (4.1), tous ces sous-modèles ne sont pas nécessairement assez visités. Ils ne pourraient donc pas être estimés de manière consistante.

Il faut aussi remarquer que, contrairement aux modèles entrée-sortie commutants, l'ordre des commutations est important dans le cas des modèles d'état parce que le nombre $\bar{s}$ est fonction des séquences $\left\{\bar{\lambda}_{t-f,2f}\right\}, t > f$ formées des $2f$ états discrets consécutifs possibles. En conclusion, le modèle (4.20), à

1. Ces deux ensembles finis ont le même cardinal.

moins que le modèle original (4.1) (dont il est dérivé) soit de petites dimensions, est généralement trop complexe (beaucoup de sous-modèles dont certains sont insignifiants) pour être estimé directement. Toutefois, en fonction de la façon dont les commutations sont acceptées par le système, il peut arriver que le modèle (4.20) ait un nombre d'états discrets bien réduit. En exemple, considérons comme à la figure 4.1, un modèle commutant entre trois sous-modèles. Les commutations sont orientées c'est-à-dire que, partant de l'état 1, le système ne peut rester que dans l'état 1 ou aller dans l'état 2 etc. Ainsi, le nombre d'états discrets du modèle (4.20) correspondant à un tel système est au plus de $s \times 2^{2f-1} = 3 \times 2^{2f-1}$ au lieu de s^{2f} (dans un cas général), où s est le nombre d'états discrets du système (4.1).

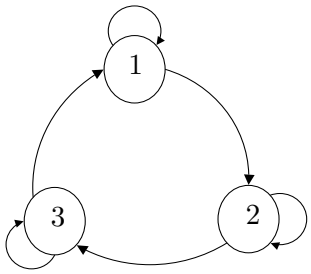


FIGURE 4.1 – Commutations particulières entre trois sous-modèles

Remarque 4.1. *Si on suppose que les commutations qui commandent le passage du système d'un état discret à un autre, ne surviennent que très rarement (temps de séjour assez grand), et si on dispose d'une fenêtre d'observation suffisamment large, alors on peut négliger l'apparition de sous-modèles mixtes dans (4.20). De cette manière, (4.20) peut pratiquement être regardé comme ayant le même nombre s de sous-modèles que le modèle d'état original (4.1). Cette approximation a été utilisée par exemple dans [58] où les données relatives aux sous-modèles mixtes sont traitées comme des « points aberrants » qu'il convient de détecter et d'exclure des données d'identification.*

4.3.3 Application de la méthode de la pondération

Dans l'objectif de limiter relativement l'explosion du nombre de sous-modèles à estimer, une alternative aux deux méthodes précédentes pourrait être d'appliquer la méthode décrite dans le chapitre 3 (Section 3.6). Pour ce faire, considérons l'équation (4.18) et supposons l'ordre n du système connu et le système (4.1) observable par chemin avec un indice d'observabilité ν. Puisque pour tout $f \geq \nu$ et pour tout t, $\text{rang}(\Gamma(\bar{\lambda}_{t,f})) = n$, nous pouvons trouver une matrice $\Lambda(\bar{\lambda}_{t,f}) \in \mathbb{R}^{n \times f n_y}$ telle que $T(\bar{\lambda}_{t,f}) = \Lambda(\bar{\lambda}_{t,f})\Gamma(\bar{\lambda}_{t,f}) \in \mathbb{R}^{n \times n}$ soit une matrice carrée non singulière. Nous pouvons donc utiliser la matrice $T(\bar{\lambda}_{t,f})$ pour réaliser une transformation d'état. En multipliant l'équation (4.18) à gauche par $\Lambda(\bar{\lambda}_{t,f})$ et en définissant un nouvel état continu

$$\bar{x}(t) = T(\bar{\lambda}_{t,f})x(t), \tag{4.21}$$

on obtient

$$\bar{x}(t) = \Lambda(\bar{\lambda}_{t,f})y_f(t) - \Lambda(\bar{\lambda}_{t,f})H(\bar{\lambda}_{t,f})u_f(t). \tag{4.22}$$

Alors, l'équation (4.18) peut être réécrite comme

$$\begin{aligned} y_f(t) &= \Gamma(\bar{\lambda}_{t,f})T(\bar{\lambda}_{t,f})^{-1}T(\bar{\lambda}_{t,f})x(t) + H(\bar{\lambda}_{t,f})u_f(t) \\ &= \bar{\Gamma}(\bar{\lambda}_{t,f})\bar{x}(t) + H(\bar{\lambda}_{t,f})u_f(t) \\ &= \bar{\Gamma}(\bar{\lambda}_{t,f})\Big(\Lambda(\bar{\lambda}_{t,f})y_f(t) - \Lambda(\bar{\lambda}_{t,f})H(\bar{\lambda}_{t,f})u_f(t)\Big) + H(\bar{\lambda}_{t,f})u_f(t) \\ &= \underbrace{\begin{bmatrix}\bar{\Gamma}(\bar{\lambda}_{t,f}) & \left(I_{fn_y} - \bar{\Gamma}(\bar{\lambda}_{t,f})\Lambda(\bar{\lambda}_{t,f})\right)H(\bar{\lambda}_{t,f})\end{bmatrix}}_{\mathcal{F}_{q_t}} \begin{bmatrix}\Lambda(\bar{\lambda}_{t,f})y_f(t) \\ u_f(t)\end{bmatrix} \\ &= \mathcal{F}_{q_t}\begin{bmatrix}\Lambda(\bar{\lambda}_{t,f})y_f(t) \\ u_f(t)\end{bmatrix}, \end{aligned} \tag{4.23}$$

où $\bar{\Gamma}(\bar{\lambda}_{t,f}) = \Gamma(\bar{\lambda}_{t,f})T(\bar{\lambda}_{t,f})^{-1}$, $\mathcal{F}_{q_t} \in \mathbb{R}^{fn_y \times (n+fn_u)}$ et $q_t \in \{1, \ldots, |\mathscr{D}_f|\}$ est défini de façon similaire à ρ_t dans (4.17) et σ_t dans (4.20). De l'équation (4.23), il est possible d'estimer les matrices $\mathcal{F}_{q_t}$.

Pour ce faire, on a besoin de connaître des matrices $\Lambda(\bar{\lambda}_{t,f})$, $t = 1, \ldots, N-f+1$, satisfaisant à la condition rang $\big(\Lambda(\bar{\lambda}_{t,f})\Gamma(\bar{\lambda}_{t,f})\big) = n$. Mais comme $\bar{\lambda}_{t,f}$ est inconnu, nous ne pouvons pas déterminer des matrices qui en dépendent à moins de générer ces matrices pour chaque instant t. Mais cela aurait pour inconvénient de multiplier le nombre de matrices $\bar{\Gamma}(\bar{\lambda}_{t,f})$ possibles et donc le nombre de matrices $\mathcal{F}_{q_t}$ à estimer puisque q_t serait alors dans $\{1, \ldots, N-f+1\}$. Pour éviter ce problème, nous considérons une matrice constante $\Lambda(\bar{\lambda}_{t,f}) = \Lambda\ \forall t$, c'est-à-dire indépendante de $\bar{\lambda}_{t,f}$ et qui vérifie

$$\text{rang}\left(\Lambda\Gamma(\bar{\lambda}_{t,f})\right) = n\ \forall t. \tag{4.24}$$

En effet on a :

Lemme 4.1. *Soit $\{A_1, \cdots, A_s\}$, un ensemble fini de matrices de $\mathbb{R}^{m\times n}$, avec $m \geq n$ et* rang$(A_i) = n$ *pour tout $i \in \{1, \ldots, s\}$. Alors il existe toujours au moins une matrice $L \in \mathbb{R}^{n\times m}$ telle que*

$$\text{rang}(LA_1) = \cdots = \text{rang}(LA_s) = n. \tag{4.25}$$

Preuve. Pour tout $j \in \{1, \ldots, s\}$, $p_j(\lambda) = \det(LA_j)$, avec $\lambda = \text{vec}(L) \in \mathbb{R}^{nm}$, définit un polynôme p_j par rapport au vecteur λ des éléments de la matrice L. Dire qu'il existe une matrice L vérifiant (4.25) revient à dire qu'il existe $\lambda \in \mathbb{R}^{nm}$ tel que $p_1(\lambda) \neq 0, \cdots, p_s(\lambda) \neq 0$, c'est-à-dire, $\prod_{j=1}^{s} p_j(\lambda) \neq 0$. Or, avec rang$(A_j^\top A_j) = n$, on a $p_j(\text{vec}(A_j^\top)) \neq 0$. Ainsi, pour tout $j \in \{1, \ldots, s\}$, le polynôme p_j n'est pas identiquement nul. Par suite, le polynôme produit $\prod_{j=1}^{s} p_j \neq 0$ n'est pas identiquement nul, comme produit de polynômes non nuls dans l'anneau $\mathbb{R}[X_1, \cdots, X_{mn}]$ des polynômes à mn variables et à coefficients dans $\mathbb{R}$. En conclusion, il existe au moins un $\lambda \in \mathbb{R}^{nm}$ satisfaisant $\prod_{j=1}^{s} p_j(\lambda) \neq 0$, et donc il existe aussi une matrice L vérifiant (4.25), ce qui conclut la preuve. ■

Grâce à ce lemme, puisque le nombre de matrices $\Gamma(\bar{\lambda}_{t,f})$ est fini et que rang $\big(\Gamma(\bar{\lambda}_{t,f})\big) = n$ pour tout t, on

peut montrer facilement (cf. Proposition 3.3 du Chapitre 3) qu'en générant Λ de façon aléatoire suivant une distribution uniforme par exemple, les conditions de rang requises (4.24) sont remplies presque sûrement. Disposant maintenant de Λ, l'équation (4.23) ne représente plus qu'un modèle entrée-sortie commutant dont on peut identifier les matrices $\mathcal{F}_{q_t}$.

4.3.4 Identification des modèles auxiliaires

De tous les modèles auxiliaires (4.17), (4.20) et (4.23) discutés, il apparaît que le modèle (4.23) est celui qui, théoriquement, présente le moins d'états discrets. Le tableau 4.1 donne à titre indicatif, la complexité en nombre de paramètres et d'états discrets des différents modèles discutés dans les sous-sections 4.3.1, 4.3.2 et 4.3.3.

Matrice de paramètres	Modèle	Nombre de paramètres	Nombre d'états discrets
L_{ρ_t}	Eq. (4.17)	$fn_y^2 + (f+1)n_u n_y$	s^{f+1}
F_{σ_t}	Eq. (4.20)	$f^2 n_y^2 + 2f^2 n_u n_y$	s^{2f}
$\mathcal{F}_{q_t}$	Eq. (4.23)	$fnn_y + f^2 n_u n_y$	s^f

TABLE 4.1 – Comparaison indicative de la complexité des différents modèles entrée-sortie commutant dérivés de (4.1).

Ainsi, bien que la stratégie d'identification proposée soit sensiblement la même pour tous ces modèles, nous nous focalisons plus particulièrement sur le modèle (4.23) que nous réécrivons sous la forme

$$y_f(t) = \mathcal{F}^y_{q_t} \Lambda y_f(t) + \mathcal{F}^u_{q_t} u_f(t), \tag{4.26}$$

où $\mathcal{F}^y_{q_t}$ et $\mathcal{F}^u_{q_t}$ sont des matrices formées respectivement des n premières colonnes et des fn_u dernières colonnes de $\mathcal{F}_{q_t}$:

$$\mathcal{F}^y_{q_t} = \bar{\Gamma}(\bar{\lambda}_{t,f}) \in \mathbb{R}^{fn_y \times n} \quad \text{et} \quad \mathcal{F}^u_{q_t} = \left(I_{fn_y} - \bar{\Gamma}(\bar{\lambda}_{t,f})\Lambda\right) H(\bar{\lambda}_{t,f}) \in \mathbb{R}^{fn_y \times fn_u}.$$

Notons pour commencer qu'en général, les $|\mathscr{D}_f|$ sous-modèles de (4.26) ne sont pas nécessairement tous suffisamment visités par les données. Ces sous-modèles qui sont pour la plupart, engendrés par les transitions entre les états discrets lors de commutations, ne pourront être estimés de manière consistante par un algorithme d'identification. De ce fait, nous devons supposer que les données relatives à chacun des $|\mathscr{D}_f|$ sous-modèles sont en nombre suffisant. Pour que le problème de l'inférence de l'état discret q_t du modèle (4.26) à partir des données soit bien posé, il est aussi nécessaire de faire l'hypothèse suivante.

Hypothèse 4.2. *Pour tout indice temporel t et pour tout couple $(i,j) \in \mathcal{S}^2$ avec $i \neq j$, $\left(\mathcal{F}_i - \mathcal{F}_j\right)\varphi(t) \neq 0$, avec $\varphi(t)$ défini par*

$$\varphi(t) = \begin{bmatrix} (\Lambda y_f(t))^\top & u_f(t)^\top \end{bmatrix}^\top \in \mathbb{R}^{n+fn_u}. \tag{4.27}$$

En d'autres termes, tout couple $\big(\varphi(t), y_f(t)\big)$ *d'observations satisfait à un unique sous-modèle de (4.26).*

Maintenant, supposons connu un majorant $s_o \geq |\mathscr{D}_f|$ du nombre d'états discrets du modèle (4.26). Nous utilisons alors l'algorithme suivant, qui procède alternativement entre l'assignation des données à un état discret et l'identification des matrices de paramètres correspondant à ces états discrets :

1. Initialiser de façon aléatoire les matrices de paramètres $\hat{\mathcal{F}}_j$, $j = 1, \ldots, s_o$. De même, fixer Λ par exemple de façon aléatoire. Nous notons $\hat{\mathcal{F}}_j(0)$ les valeurs affectées à ces matrices au cours de la phase d'initialisation.
2. Pour toute paire $\big(u_f(t), y_f(t)\big)$ d'observations,
   – Estimer l'état discret par

$$\hat{q}_t = \underset{j=1,\ldots,s_o}{\arg\min} \frac{1}{\delta_j(t-1)} \left\| y_f(t) - \hat{\mathcal{F}}_j(t-1)\varphi(t) \right\|_2, \tag{4.28}$$

   avec

$$\delta_j(t-1) = \sqrt{fn_y + \mathrm{trace}(\hat{\mathcal{F}}_j(t-1)^\top \hat{\mathcal{F}}_j(t-1))}. \tag{4.29}$$

   En fait, $\delta_j(t-1)$ est la norme de Frobenius de la matrice $\begin{bmatrix} I_{fn_y} & \hat{\mathcal{F}}_j(t-1) \end{bmatrix}$ et donc le division par $\delta_j(t-1)$ dans le critère (4.28) consiste en une normalisation.
   – Adapter [1] en conséquence la matrice de paramètres $\hat{\mathcal{F}}_{\hat{q}_t}(t-1)$.
3. Aller à l'étape 2 jusqu'à ce que toutes les données soient traitées.
4. Répéter si nécessaire la procédure sur l'ensemble des données en se servant des estimées finales obtenues pour initialiser l'étape suivante.

   Si N désigne le nombre total d'observations $\big(u(t), y(t)\big)$ disponibles, alors $\hat{\mathcal{F}}_j(N_o)$, avec $N_o = N - f + 1$, sont les estimées finales obtenues à l'issue de la procédure récursive décrite par les étapes 2 et 3. En se servant de ces estimées on peut reconstruire l'état discret de la façon suivante

$$\hat{q}_t = \underset{j=1,\ldots,s_o}{\arg\min} \frac{1}{\delta_j(N_o)} \left\| y_f(t) - \hat{\mathcal{F}}_j(N_o)\varphi(t) \right\|_2,$$

   pour tout $t = 1, \ldots, N_o$. De cette nouvelle reconstruction de l'état discret, nous pouvons décider d'une répétition des procédures 2 et 3 selon que $\mathscr{C}$ soit supérieur ou non à un certain seuil, où $\mathscr{C}$ est par exemple le critère défini par

$$\mathscr{C}\big(\hat{\mathcal{F}}_{j=1}^{s_o}\big) = \sum_{t=1}^{N_o} \frac{1}{\delta_{\hat{q}_t}(N_o)} \left\| y_f(t) - \hat{\mathcal{F}}_{\hat{q}_t}(N_o)\varphi(t) \right\|_2.$$

Sous réserve d'une bonne initialisation de l'algorithme, les estimées $\hat{\mathcal{F}}_{\hat{q}_t}$ se stabilisent au bout d'un certain nombre d'itérations. Dans ce cas, on pourrait obtenir le nombre de sous-modèles consistants en éliminant dans les s_o sous-modèles initialement présumés, ceux auxquels très peu de données ont été affectées.

1. par les moindres carrés récursifs par exemple [75].

Notons cependant que l'initialisation est une étape cruciale dans la convergence d'une telle procédure. Pour assurer cette étape, on pourrait utiliser par exemple l'algorithme GPCA [129]. Mais cette méthode pourrait souffrir dans certains cas, d'un problème sévère de dimensionalité puisque le nombre $|\mathscr{D}_f|$ d'états discrets du modèle est potentiellement très élevé. On pourra alors se contenter d'une initialisation aléatoire. Dans ce dernier cas, le résultat de l'algorithme n'est plus déterministe si bien que plusieurs essais peuvent être souvent nécessaires à l'obtention d'estimées de bonne qualité.

4.3.5 Extraction des matrices du système

Etant donné les matrices $\mathcal{F}_{q_t}$ du modèle (4.23), nous nous intéressons maintenant à l'extraction d'une réalisation d'état du système commutant. Notons qu'on peut directement obtenir la matrice d'observabilité $\bar{\Gamma}(\bar{\lambda}_{t,f})$ de la façon suivante

$$\bar{\Gamma}(\bar{\lambda}_{t,f}) = \mathcal{F}^y_{q_t} = \mathcal{F}_{q_t}(:, 1:n).$$

Par contre, on ne peut pas calculer directement $H(\bar{\lambda}_{t,f})$ à partir de $\mathcal{F}^u_{q_t} = \mathcal{F}_{q_t}(:, n+1 : n+fn_u)$ car, comme nous l'avons déjà mis en évidence dans le Chapitre 3, $I_{fn_y} - \bar{\Gamma}(\bar{\lambda}_{t,f})\Lambda$ n'est pas inversible. En effet, pour toute matrice Λ vérifiant (4.24), on a

$$\text{rang}\left(I_{fn_y} - \bar{\Gamma}(\bar{\lambda}_{t,f})\Lambda\right) = fn_y - n.$$

En vue d'extraire les matrices du système, remarquons que la transformation (4.21) de l'état s'accompagne d'une modification des matrices du modèle (4.1) ainsi que de son nombre d'états discrets. Plus précisément, cette transformation induit une modification du modèle (4.1) $\{A_{\lambda_t}, B_{\lambda_t}, C_{\lambda_t}, \ D_{\lambda_t}\}_{\lambda_t \in \mathcal{S}}$ en un autre modèle commutant $\left\{\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{C}_{p_t}, \bar{D}_{p_t}\right\}_{p_t \in \{1,\ldots,|\mathscr{D}_{f+1}|\}}$ tel que

$$\left[\begin{array}{c|c} \bar{A}_{p_t} & \bar{B}_{p_t} \\ \hline \bar{C}_{p_t} & \bar{D}_{p_t} \end{array}\right] = \left[\begin{array}{c|c} T(\bar{\lambda}_{t+1,f}) A_{\lambda_t} T(\bar{\lambda}_{t,f})^{-1} & T(\bar{\lambda}_{t+1,f}) B_{\lambda_t} \\ \hline C_{\lambda_t} T(\bar{\lambda}_{t,f})^{-1} & D_{\lambda_t} \end{array}\right], \tag{4.30}$$

avec $T(\bar{\lambda}_{t,f}) = \Lambda\Gamma(\bar{\lambda}_{t,f})$ et $p_t \in \{1, \ldots, |\mathscr{D}_{f+1}|\}$. Ici, nous avons défini l'état discret p_t en exploitant la correspondance bijective entre l'ensemble fini $\left\{\bar{\lambda}_{t,f+1} : t = 1, \ldots, N-f\right\}$ des séquences d'états discrets impliquées dans $T(\bar{\lambda}_{t+1,f}) A_{\lambda_t} T(\bar{\lambda}_{t,f})^{-1}$, et l'ensemble $\{1, \ldots, |\mathscr{D}_{f+1}|\}$.

Remarque 4.2. *Dans les équations (4.30), l'état discret p_t prend ses valeurs dans l'ensemble fini $\{1, \ldots, |\mathscr{D}_{f+1}|\}$, où $\mathscr{D}_{f+1}$ est l'ensemble des séquences d'états discrets de longueur $f+1$ qui sont réalisables tandis que l'état discret q_t du système (4.23) prend ses valeurs plutôt dans $\{1, \ldots, |\mathscr{D}_f|\}$.*
En réalité, selon l'équation (4.30), les matrices $\bar{C}_{p_t}, \bar{D}_{p_t}$ peuvent être indexées par q_t qui correspond directement à la séquence $\bar{\lambda}_{t,f}$. La matrice $\bar{D}_{p_t}$ peut même être indexée par l'état discret $\lambda_t \in \mathcal{S}$ car elle prend au plus s valeurs distinctes ($\bar{D}_{p_t} = D_{\lambda_t}$). Par contre les matrices $\bar{A}_{p_t}$ et $\bar{B}_{p_t}$ (qui dépendent de q_t et q_{t+1}) peuvent prendre $|\mathscr{D}_{f+1}| \geq |\mathscr{D}_f|$ valeurs différentes.

En prenant $f = \nu + 1$, les matrices du modèle (4.30) peuvent être obtenues grâce à la proposition 4.2.

Pour pouvoir formuler cette proposition de façon commode, nous introduisons la notation

$$\mathcal{K}_{q_t} = I_{fn_y} - \bar{\Gamma}(\bar{\lambda}_{t,f})\Lambda = I_{fn_y} - \mathcal{F}^y_{q_t}\Lambda,$$

et considérons les partitions suivantes

$$\begin{aligned}
\mathcal{F}^y_{q_t} &= \begin{bmatrix}(\mathcal{F}^{y,1}_{q_t})^\top & \cdots & (\mathcal{F}^{y,f}_{q_t})^\top\end{bmatrix}^\top \in \mathbb{R}^{fn_y \times n},\\
\mathcal{F}^{y,\alpha:\beta}_{q_t} &= \begin{bmatrix}(\mathcal{F}^{y,\alpha}_{q_t})^\top & \cdots & (\mathcal{F}^{y,\beta}_{q_t})^\top\end{bmatrix}^\top \in \mathbb{R}^{(\beta-\alpha+1)n_y \times n},\\
\mathcal{K}_{q_t} &= \begin{bmatrix}\mathcal{K}^1_{q_t} & \cdots & \mathcal{K}^f_{q_t}\end{bmatrix} \in \mathbb{R}^{fn_y \times fn_y},\\
\mathcal{K}^{\alpha:\beta}_{q_t} &= \begin{bmatrix}\mathcal{K}^\alpha_{q_t} & \cdots & \mathcal{K}^\beta_{q_t}\end{bmatrix} \in \mathbb{R}^{fn_y \times (\beta-\alpha+1)n_y},\\
\mathcal{F}^u_{q_t} &= \begin{bmatrix}\mathcal{F}^{u,1}_{q_t} & \cdots & \mathcal{F}^{u,f}_{q_t}\end{bmatrix} \in \mathbb{R}^{fn_y \times fn_u},
\end{aligned}$$

où $\mathcal{F}^{y,k}_{q_t} \in \mathbb{R}^{n_y \times n}$, $\mathcal{K}^k_{q_t} \in \mathbb{R}^{fn_y \times n_y}$, $\mathcal{F}^{u,k}_{q_t} \in \mathbb{R}^{fn_y \times n_u}$, $k = 1, \ldots, f$, et $1 \leq \alpha \leq \beta \leq f$.
A partir des matrices $\{\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{C}_{p_t}, \bar{D}_{p_t}\}$ définies en (4.30), si l'on construit les matrices $\bar{\Gamma}(\bar{p}_{t,f})$ et $\bar{H}(\bar{p}_{t,f})$ selon les définitions (4.7)-(4.8), il apparaît, selon les structures de $\mathcal{F}^u_{q_t}$ et $\mathcal{F}^y_{q_t}$, que les relations suivantes sont vérifiées pour tout $t \geq f$.

$$\begin{aligned}
\bar{C}_{p_t} &= \mathcal{F}^{y,1}_{q_t},\\
\mathcal{F}^{y,1:f-1}_{q_{t+1}}\bar{A}_{p_t} &= \mathcal{F}^{y,2:f}_{q_t},\\
\mathcal{K}^f_{q_{t-f+1}}\bar{D}_{p_t} &= \mathcal{F}^{u,f}_{q_{t-f+1}},\\
\mathcal{K}^{2:f}_{q_t}\mathcal{F}^{y,1:f-1}_{q_{t+1}}\bar{B}_{p_t} &= \mathcal{F}^{u,1}_{q_t} - \mathcal{K}^1_{q_t}\bar{D}_{p_t}.
\end{aligned} \tag{4.31}$$

Dans le but d'exploiter ces relations pour la détermination des matrices $\{\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{C}_{p_t}, \bar{D}_{p_t}\}$, il est nécessaire de savoir quels états discrets q_{t+1} on peut atteindre en partant d'un état discret $q_t = j \in \{1, \ldots, |\mathscr{D}_f|\}$. En d'autres termes, étant donné $q_t = j$, l'état q_{t+1} peut-il prendre une valeur quelconque dans $\{1, \ldots, |\mathscr{D}_f|\}$? En fait cela n'est pas le cas. L'assignation d'une valeur à q_t impose automatiquement une contrainte sur l'ensemble des valeurs autorisées pour q_{t+1}. Pour le voir, reportons nous à la définition de l'état discret q_t dans (4.23) pour remarquer l'existence d'une bijection $\pi : \mathscr{D}_f \longrightarrow \{1, \ldots, |\mathscr{D}_f|\}$, $\bar{\lambda}_{t,f} \longmapsto q_t$. En conséquence, on a

$$\begin{aligned}
q_t &= \pi(\bar{\lambda}_{t,f}) = \pi(\lambda_t \underbrace{\lambda_{t+1} \cdots \lambda_{t+f-1}}_{\bar{\lambda}_{t+1,f-1}})\\
q_{t+1} &= \pi(\bar{\lambda}_{t+1,f}) = \pi(\underbrace{\lambda_{t+1} \cdots \lambda_{t+f-1}}_{\bar{\lambda}_{t+1,f-1}} \lambda_{t+f}).
\end{aligned}$$

Il apparaît donc que si on fixe q_t, alors q_{t+1} peut prendre au plus s valeurs distinctes correspondant au choix de λ_{t+f} dans $\mathcal{S}$. Ainsi, au vu des relations (4.31), p_t vit dans un ensemble fini dont la cardinal vaut $|\mathscr{D}_f| \cdot s$.

Proposition 4.2. *Etant donné les matrices $\mathcal{F}_j$, $j = 1, \ldots, s$, et l'état discret q_t du modèle (4.23), on a pour tout $t \geq f-1$*

$$\begin{cases} \bar{A}_{p_t} = \left(\mathcal{F}_{q_{t+1}}^{y,1:f-1}\right)^{\dagger} \mathcal{F}_{q_t}^{y,2:f}, \\ \bar{C}_{p_t} = \mathcal{F}_{q_t}^{y,1}, \\ \begin{bmatrix} \bar{D}_{p_t} \\ \bar{B}_{p_t} \end{bmatrix} = \left[\mathscr{K}(\bar{q}_{t-f+1,f})\mathcal{R}_{q_{t+1}}\right]^{\dagger} \begin{bmatrix} \mathcal{F}_{q_{t-f+1}}^{u,f} \\ \mathcal{F}_{q_{t-f+2}}^{u,f-1} \\ \vdots \\ \mathcal{F}_{q_t}^{u,1} \end{bmatrix}, \end{cases} \tag{4.32}$$

où

$$\mathcal{R}_{q_{t+1}} = \begin{bmatrix} I_{n_y} & \mathrm{O}_{n_y,n} \\ \mathrm{O}_{(f-1)n_y,n_y} & \mathcal{F}_{q_{t+1}}^{y,1:f-1} \end{bmatrix},$$

et

$$\mathscr{K}(\bar{q}_{t-f+1,f}) = \begin{bmatrix} \mathcal{K}_{q_{t-f+1}}^{f} & \mathrm{O}_{fn_y,n_y} & \cdots & \mathrm{O}_{fn_y,n_y} & \mathrm{O}_{fn_y,n_y} \\ \mathcal{K}_{q_{t-f+2}}^{f-1} & \mathcal{K}_{q_{t-f+2}}^{f} & \cdots & \mathrm{O}_{fn_y,n_y} & \mathrm{O}_{fn_y,n_y} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathcal{K}_{q_{t-1}}^{2} & \mathcal{K}_{q_{t-1}}^{3} & \cdots & \mathcal{K}_{q_{t-1}}^{f} & \mathrm{O}_{fn_y,n_y} \\ \mathcal{K}_{q_t}^{1} & \mathcal{K}_{q_t}^{2} & \cdots & \mathcal{K}_{q_t}^{f-1} & \mathcal{K}_{q_t}^{f} \end{bmatrix}$$

avec $\bar{q}_{t-f+1,f} = q_{t-f+1} \cdots q_t$ est une séquence d'états discrets et O_{fn_y,n_y} désigne une matrice de dimensions $(fn_y \times n_y)$ formée seulement avec des zéros.

Preuve. [Illustration] Les matrices $\bar{A}_{p_t}$ et $\bar{C}_{p_t}$ se déduisent directement des relations (4.31). Par contre, pour déterminer les matrices $\bar{B}_{p_t}$ et $\bar{D}_{p_t}$, on ne peut pas procéder en général par calcul direct en partant de (4.31) car les matrices $\mathcal{K}_{q_{t-f+1}}^{f} \in \mathbb{R}^{fn_y \times n_y}$ et $\mathcal{K}_{q_t}^{2:f}\mathcal{F}_{q_{t+1}}^{y,1:f-1} \in \mathbb{R}^{(f-1)n_y \times n}$ ne sont pas nécessairement de rang plein colonne. En effet, on a par exemple $\mathrm{rang}\left(\mathcal{K}_{q_t}^{2:f}\right) \leq \mathrm{rang}\left(\mathcal{K}_{q_t}\right) = fn_y - n$. Ainsi, $\mathrm{rang}\left(\mathcal{K}_{q_t}^{2:f}\mathcal{F}_{q_{t+1}}^{y,1:f-1}\right) \leq \min(n, fn_y - n)$. Or, il peut arriver que $fn_y - n$ soit strictement inférieur à n (par exemple, pour $n_y = 1$, $n \geq 2$ et $f = n+1$), ce qui implique que dans ce cas, $\mathcal{K}_{q_t}^{2:f}\mathcal{F}_{q_{t+1}}^{y,1:f-1}$ n'est pas de rang plein colonne.

Pour calculer $\bar{B}_{p_t}$ et $\bar{D}_{p_t}$, nous commençons par remarquer que pour tout t, $H(\bar{\lambda}_{t,f}) = \bar{H}(\bar{p}_{t,f})$, où $\bar{H}(\bar{p}_{t,f})$ est définie de la même manière que $H(\bar{\lambda}_{t,f})$ à partir des matrices $\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{C}_{p_t}, \bar{D}_{p_t}$. On a alors $\mathcal{K}_{q_t} H(\bar{p}_{t,f}) = \mathcal{F}_{q_t}^{u}$. En nous basant sur cette relation, nous exprimons le 1er bloc de n_u colonnes de $\mathcal{K}_{q_t} H(\bar{p}_{t,f})$ puis le 2ème bloc de colonnes de $\mathcal{K}_{q_{t-1}} H(\bar{p}_{t-1,f})$ et ainsi de suite jusqu'au $f$ème bloc de colonnes de $\mathcal{K}_{q_{t-f+1}} H(\bar{p}_{t-f+1,f})$. En posant

$$\mathcal{R}_{q_{t+1}}^{1:\alpha} = \begin{bmatrix} I_{n_y} & \mathrm{O}_{n_y,n} \\ \mathrm{O}_{\alpha n_y,n_y} & \mathcal{F}_{q_{t+1}}^{y,1:\alpha-1} \end{bmatrix},$$

avec $1 \leq \alpha \leq f$, on a

$$t,\ \text{bloc } 1 \quad \mathcal{K}_{q_t} \begin{bmatrix} D_{\lambda_t} \\ C_{\lambda_{t+1}} B_{\lambda_t} \\ \vdots \\ C_{\lambda_{t+f-1}} A_{\lambda_{t+f-2}} \cdots A_{\lambda_{t+1}} B_{\lambda_t} \end{bmatrix} = \mathcal{K}_{q_t}^{1:f} \mathcal{R}_{q_{t+1}}^{1:f} \begin{bmatrix} \bar{D}_{p_t} \\ \bar{B}_{p_t} \end{bmatrix} = \mathcal{F}_{q_t}^{u,1},$$

$$
\begin{array}{lcccc}
t-1,\ \text{bloc } 2 & \mathcal{K}_{q_{t-1}} \begin{bmatrix} \mathrm{O}_{n_y,n_u} \\ D_{\lambda_t} \\ C_{\lambda_{t+1}} B_{\lambda_t} \\ \vdots \\ C_{\lambda_{t+f-2}} A_{\lambda_{t+f-3}} \cdots A_{\lambda_{t+1}} B_{\lambda_t} \end{bmatrix} & = & \mathcal{K}^{2:f}_{q_{t-1}} \mathcal{R}^{1:f-1}_{q_{t+1}} \begin{bmatrix} \bar{D}_{p_t} \\ \bar{B}_{p_t} \end{bmatrix} & = & \mathcal{F}^{u,2}_{q_{t-1}}, \\
\vdots & \vdots & = & \ldots & = & \vdots \\
t-f+1,\ \text{bloc } f & \mathcal{K}_{q_{t-f+1}} \begin{bmatrix} \mathrm{O}_{n_y,n_u} \\ \mathrm{O}_{n_y,n_u} \\ \vdots \\ D_{\lambda_t} \end{bmatrix} & = & \mathcal{K}^{f}_{q_{t-f+1}} \mathcal{R}^{1}_{q_{t+1}} \begin{bmatrix} \bar{D}_{p_t} \\ \bar{B}_{p_t} \end{bmatrix} & = & \mathcal{F}^{u,f}_{q_{t-f+1}}.
\end{array}
$$

Ces relations sont inchangées si on remplace $A_{\lambda_t}, B_{\lambda_t}, C_{\lambda_t}, D_{\lambda_t}$ par $\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{C}_{p_t}, \bar{D}_{p_t}$. De même, on peut exprimer indifféremment $\bar{\Gamma}(\bar{\lambda}_{t,f})$ comme $\bar{\Gamma}(\bar{p}_{t,f})$ à partir des matrices $\bar{A}_{p_t}$, $\bar{B}_{p_t}$, $\bar{C}_{p_t}$, $\bar{D}_{p_t}$. Ainsi, les expressions (4.32) des matrices $\bar{B}_{p_t}, \bar{D}_{p_t}$ en résultent immédiatement. ■

Remarque 4.3. *Dans l'établissement des équations (4.32), aucune hypothèse n'a été formulée concernant la nature, l'origine ou le mécanisme de commutations. Cela signifie que la méthode proposée peut être appliquée dans un contexte où les commutations sont complètement arbitraires. La stabilité du système global n'est pas explicitement requise mais elle peut être importante pour assurer le bon conditionnement numérique des matrices de données.*

Aussi, puisque les formules (4.32) sont assez génériques, elles peuvent s'appliquer dans le cas particulier où les instants de commutation sont séparés par une certaine durée minimum (cf. Section 4.4).

Remarque 4.4. *Un modèle linéaire (cf. Chapitre 3) correspond à un cas particulier de modèle commutant dans lequel $s = 1$. Ainsi, les formules (4.32) peuvent s'appliquer aussi dans le cas plus simple d'un modèle linéaire avec $q_t = p_t = \lambda_t = 1$ pour tout instant t. De ce fait, toutes les matrices impliquées dans l'équation (4.32) deviennent constantes.*

Pour pouvoir appliquer la proposition 4.2, il est indispensable que les matrices $\mathcal{F}_{q_t}$ soient bien estimées et que l'état discret q_t soit exactement reconstruit. Pour illustrer l'importance de cette exigence, écrivons

la relation entrée-sortie (4.3) du système (4.1) en utilisant les matrices $\{\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{C}_{p_t}, \bar{D}_{p_t}\}$. On obtient

$$\begin{aligned} y(t) =& \bar{C}_{p_t}\bar{A}_{p_{t-1}}\dots\bar{A}_{p_0}\bar{x}(0) \\ &+ \bar{C}_{p_t}\bar{A}_{p_{t-1}}\dots\bar{A}_{p_1}\bar{B}_{p_0}u(0) \\ &+ \bar{C}_{p_t}\bar{A}_{p_{t-1}}\dots\bar{A}_{p_2}\bar{B}_{p_1}u(1) \\ &+ \quad \vdots \\ &+ \bar{C}_{p_t}\bar{A}_{p_{t-1}}\bar{B}_{p_{t-2}}u(t-2) \\ &+ \bar{C}_{p_t}\bar{B}_{p_{t-1}}u(t-1) \\ &+ \bar{D}_{p_t}u(t). \end{aligned}$$

En utilisant la relation (4.30), il vient

$$\begin{aligned} y(t) =& C_{\lambda_t}\boldsymbol{T}(\bar{\lambda}_{t,f})^{-1}\cdot\boldsymbol{T}(\bar{\lambda}_{t,f})A_{\lambda_{t-1}}\boldsymbol{T}(\bar{\lambda}_{t-1,f})^{-1}\dots\boldsymbol{T}(\bar{\lambda}_{1,f})A_{\lambda_0}\boldsymbol{T}(\bar{\lambda}_{0,f})^{-1}\cdot\boldsymbol{T}(\bar{\lambda}_{0,f})x(0) \\ &+ C_{\lambda_t}\boldsymbol{T}(\bar{\lambda}_{t,f})^{-1}\cdot\boldsymbol{T}(\bar{\lambda}_{t,f})A_{\lambda_{t-1}}\boldsymbol{T}(\bar{\lambda}_{t-1,f})^{-1}\cdots\boldsymbol{T}(\bar{\lambda}_{2,f})A_{\lambda_1}\boldsymbol{T}(\bar{\lambda}_{1,f})^{-1}\cdot\boldsymbol{T}(\bar{\lambda}_{1,f})B_{\lambda_0}u(0) \\ &+ C_{\lambda_t}\boldsymbol{T}(\bar{\lambda}_{t,f})^{-1}\cdot\boldsymbol{T}(\bar{\lambda}_{t,f})A_{\lambda_{t-1}}\boldsymbol{T}(\bar{\lambda}_{t-1,f})^{-1}\cdot\dots\boldsymbol{T}(\bar{\lambda}_{3,f})A_{\lambda_2}\boldsymbol{T}(\bar{\lambda}_{2,f})^{-1}\cdot\boldsymbol{T}(\bar{\lambda}_{2,f})B_{\lambda_1}u(1) \\ &+ \cdots \quad \vdots \quad \cdots \\ &+ C_{\lambda_t}\boldsymbol{T}(\bar{\lambda}_{t,f})^{-1}\cdot\boldsymbol{T}(\bar{\lambda}_{t,f})A_{\lambda_{t-1}}\boldsymbol{T}(\bar{\lambda}_{t-1,f})^{-1}\cdot\boldsymbol{T}(\bar{\lambda}_{t-1,f})B_{\lambda_{t-2}}u(t-2) \\ &+ C_{\lambda_t}\boldsymbol{T}(\bar{\lambda}_{t,f})^{-1}\cdot\boldsymbol{T}(\bar{\lambda}_{t,f})B_{\lambda_{t-1}}u(t-1) \\ &+ D_{\lambda_t}u(t). \end{aligned}$$

Ainsi, on voit que pour que le comportement entrée-sortie soit préservé, il faudrait que les matrices de transformation $T(\bar{\lambda}_{t,f})$ se compensent complètement. Nous disons alors que les matrices $\{\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{C}_{p_t}, \bar{D}_{p_t}\}$ sont *cohérentes* ou *concordantes*. Cependant, lorsque l'état discret q_t est mal estimé, alors l'équation (4.32) peut fournir des matrices non cohérentes. Cela aura pour effet d'introduire une inadéquation entre le comportement entrée-sortie du système réel et celui que l'on pourrait reconstruire à partir du modèle estimé.

Comme exemple, considérons un cas simple où $f = 3$ et $\nu = 2$. Alors, les matrices $\mathcal{F}_j$ à t et $t+1$ peuvent s'écrire

$$\mathcal{F}^y_{q_t} = \begin{bmatrix} \bar{C}_{p_t} \\ \hline \bar{C}_{p_{t+1}}\bar{A}_{p_t} \\ \bar{C}_{p_{t+2}}\bar{A}_{p_{t+1}}\bar{A}_{p_t} \end{bmatrix} \quad \text{et} \quad \mathcal{F}^y_{q_{t+1}} = \begin{bmatrix} \bar{C}_{p_{t+1}} \\ \bar{C}_{p_{t+2}}\bar{A}_{p_{t+1}} \\ \hline \bar{C}_{p_{t+3}}\bar{A}_{p_{t+2}}\bar{A}_{p_{t+1}} \end{bmatrix},$$

et on a normalement

$$\mathcal{F}^{y,2:f}_{q_t} = \mathcal{F}^{y,1:f-1}_{q_{t+1}}\bar{A}_{p_t}. \tag{4.33}$$

Remplaçons maintenant q_t et q_{t+1} par leurs estimées respectives $\hat{q}_t$ et $\hat{q}_{t+1}$ en considérant par exemple que q_t est bien estimé mais que q_{t+1} a été mal estimé, c'est-à-dire, $\hat{q}_t = q_t$ et $\hat{q}_{t+1} \neq q_{t+1}$. Alors très clairement, les matrices $\mathcal{F}^y_{\hat{q}_t}$ et $\mathcal{F}^y_{\hat{q}_{t+1}}$ ne satisfont plus à la relation (4.33) d'où une probable mauvaise estimation de $\bar{A}_{p_t}$.

Une question que l'on pourrait se poser maintenant est la suivante : est-il possible de retrouver au moins une réalisation $\{A_{\lambda_t}, B_{\lambda_t}, C_{\lambda_t}, D_{\lambda_t}\}$, $\lambda_t \in \mathcal{S}$, du modèle (4.1) en partant des matrices $\{\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{C}_{p_t}, \bar{D}_{p_t}\}$, $p_t \in \{1, \ldots, |\mathscr{D}_{f+1}|\}$, estimées ci-dessus par la relation (4.32) ? Bien évidemment, la réponse à cette question serait trivialement affirmative si les matrices $T(\bar{\lambda}_{t,f})$ de transformation étaient connues ou toutes égales. Malheureusement cela n'est pas le cas si l'on considère un cadre très général. Cependant, dans certains cas particuliers, il existe une correspondance directe entre les deux réalisations du système.

Exemple 4.3. *Par exemple, considérons un système MISO sous la forme (4.1), avec*

$$C_{\lambda_t} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

$$A_{\lambda_t} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a^0_{\lambda_t} & -a^1_{\lambda_t} & \cdots & -a^{n-1}_{\lambda_t} \end{bmatrix}.$$

pour tout $\lambda_t \in \mathcal{S}$, les matrices B_{λ_t} et C_{λ_t} étant quelconques. Alors pour tout instant t, on a $\Gamma(\bar{\lambda}_{t,n}) = I_n$ quelque soit le mécanisme de commutation. Ainsi, en prenant $f = n+1$, on peut prendre $\Lambda = \begin{bmatrix} I_n & 0_{n,1} \end{bmatrix}$ de sorte que $T(\bar{\lambda}_{t,f}) = \Lambda\Gamma(\bar{\lambda}_{t,f}) = I_n$ pour tout t. Il en résulte que

$$\left[\begin{array}{c|c} \bar{A}_{p_t} & \bar{B}_{p_t} \\ \hline \bar{C}_{p_t} & \bar{D}_{p_t} \end{array}\right] = \left[\begin{array}{c|c} A_{\lambda_t} & B_{\lambda_t} \\ \hline C_{\lambda_t} & D_{\lambda_t} \end{array}\right].$$

Dans un cas général, pour passer du modèle (4.32) ($|\mathscr{D}_{f+1}|$ sous-modèles) à un modèle du type (4.1) (s sous-modèles) il devra probablement être nécessaire d'employer des techniques de réduction de modèle comme dans le cas des modèles linéaires. Nous ne traitons pas de cet aspect ici.

4.3.6 Exemple numérique

A titre illustratif, nous considérons maintenant un exemple numérique de système à commutations, composé de deux sous-modèles dynamiques d'ordre 2 avec des matrices définies par

$$\begin{cases} A_1 = \begin{bmatrix} 1.25 & -0.49 \\ 1 & 0 \end{bmatrix} & B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ C_1 = \begin{bmatrix} 0.5 & 1 \end{bmatrix} & D_1 = 1.2, \end{cases}$$

$$\begin{cases} A_2 = \begin{bmatrix} 0.2 & 0.7 \\ 0.5 & 0 \end{bmatrix} & B_2 = \begin{bmatrix} 0.3 \\ 1 \end{bmatrix} \\ C_2 = \begin{bmatrix} 0.05 & 2 \end{bmatrix} & D_2 = 0.5. \end{cases}$$

Alors, en prenant la matrice de pondération

$$\Lambda = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix},$$

le système (4.23) est constitué des 8 sous-modèles suivants, obtenus à partir des matrices (A_1, B_1, C_1, D_1) et (A_2, B_2, C_2, D_2) définies ci-dessus :

$$\bar{\lambda} = 111, \quad \mathcal{F}_1 = \left[\begin{array}{cc|ccc} 1.2784 & -0.7102 & -0.9023 & 0.5682 & -0.6818 \\ 0.2784 & 0.2898 & -0.9023 & 0.5682 & -0.6818 \\ -0.2784 & 0.7102 & 0.9023 & -0.5682 & 0.6818 \end{array}\right]$$

$$\bar{\lambda} = 112, \quad \mathcal{F}_2 = \left[\begin{array}{cc|ccc} 1.2683 & -0.8071 & -0.8694 & 0.9455 & -0.2306 \\ 0.2683 & 0.1929 & -0.8694 & 0.9455 & -0.2306 \\ -0.2683 & 0.8071 & 0.8694 & -0.9455 & 0.2306 \end{array}\right]$$

$$\bar{\lambda} = 121, \quad \mathcal{F}_3 = \left[\begin{array}{cc|ccc} 1.2113 & -0.4547 & -0.6848 & -0.6427 & -0.9079 \\ 0.2113 & 0.5453 & -0.6848 & -0.6427 & -0.9079 \\ -0.2113 & 0.4547 & 0.6848 & 0.6427 & 0.9079 \end{array}\right]$$

$$\bar{\lambda} = 122, \quad \mathcal{F}_4 = \left[\begin{array}{cc|ccc} 1.3759 & -0.5873 & -1.2183 & -1.2955 & -0.3943 \\ 0.3759 & 0.4127 & -1.2183 & -1.2955 & -0.3943 \\ -0.3759 & 0.5873 & 1.2183 & 1.2955 & 0.3943 \end{array}\right]$$

$$\bar{\lambda} = 211, \quad \mathcal{F}_5 = \left[\begin{array}{cc|ccc} 0.7145 & -0.1625 & 0.1958 & -0.0810 & -0.6624 \\ -0.2855 & 0.8375 & 0.1958 & -0.0810 & -0.6624 \\ 0.2855 & 0.1625 & -0.1958 & 0.0810 & 0.6624 \end{array}\right]$$

$$\bar{\lambda} = 212, \quad \mathcal{F}_6 = \left[\begin{array}{cc|ccc} 0.7246 & -0.2762 & 0.1888 & 0.3090 & -0.2242 \\ -0.2754 & 0.7238 & 0.1888 & 0.3090 & -0.2242 \\ 0.2754 & 0.2762 & -0.1888 & -0.3090 & 0.2242 \end{array}\right]$$

$$\bar{\lambda} = 221, \quad \mathcal{F}_7 = \left[\begin{array}{cc|ccc} 0.8621 & -0.1896 & 0.0946 & -0.6786 & -0.8070 \\ -0.1379 & 0.8104 & 0.0946 & -0.6786 & -0.8070 \\ 0.1379 & 0.1896 & -0.0946 & 0.6786 & 0.8070 \end{array}\right]$$

$$\bar{\lambda} = 222, \quad \mathcal{F}_8 = \left[\begin{array}{cc|ccc} 0.7742 & -0.1290 & 0.1548 & -1.2355 & -0.3226 \\ -0.2258 & 0.8710 & 0.1548 & -1.2355 & -0.3226 \\ 0.2258 & 0.1290 & -0.1548 & 1.2355 & 0.3226 \end{array}\right],$$

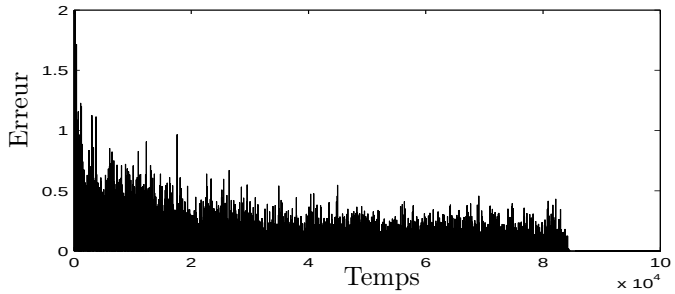
où le trait vertical sépare les sous-matrices $\mathcal{F}_j^u$ et $\mathcal{F}_j^y$.

Pour estimer les paramètres de ce modèle par l'algorithme de la sous-section 4.3.4, nous utilisons une entrée d'excitation sous forme de bruit blanc. Les séquences de commutation sont générées de façon aléatoire. Afin de pouvoir exciter suffisamment tous les 8 sous-modèles du système, nous générons un échantillon de données de taille 100 000. Puis nous ajoutons sur la sortie, un bruit blanc dans une proportion d'un RSB de 30 dB.

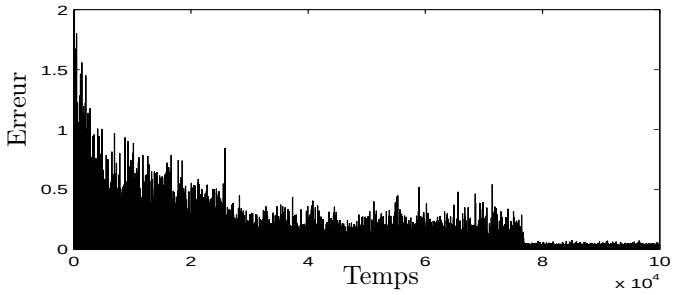
Nous présentons alors sur la figure 4.2 l'évolution du minimum

$$\min_{j=1,\ldots,s_o} \frac{1}{\delta_j(t-1)} \left\| y_f(t) - \hat{\mathcal{F}}_j(t-1)\varphi(t) \right\|_2 \tag{4.34}$$

de l'erreur normalisée qui apparaît dans (4.28). Nous pouvons constater que cette erreur ne converge pas de façon monotone vers zéro. Cela est dû au fait suivant. L'algorithme proposé à la page 114 consiste en réalité en un ensemble de s_o sous-algorithmes d'identification récursive qui opèrent en parallèle. Chaque sous-algorithme vise à identifier les paramètres associés à un seul état discret. Ainsi, en fonction de la façon dont les différents états discrets du modèle (4.26) sont visités par le système, les s_o sous-algorithmes ne convergent pas nécessairement en même temps, d'où le manque apparent de monotonie dans l'évolution de l'erreur (4.34) représentée sur la figure 4.2. Ici, en partant d'une initialisation aléatoire, la convergence se produit après le traitement d'environ 80 000 observations. En absence de bruit (Figure 4.2-(a)), l'erreur (4.34) s'annule complètement après convergence. Par contre, la présence de bruit (Figure 4.2-(b)) dans les observations provoque, lorsque la convergence se produit, une stabilisation de l'erreur à une valeur constante et relativement proche de zéro.
Sur la figure 4.3, nous représentons l'état discret réel du système en même temps que l'état discret estimé (en présence de bruit) sur un horizon volontairement restreint à [0, 300] pour des raisons de lisibilité. Il apparaît que l'état discret estimé correspond à l'état discret réel sauf, du fait de la présence du bruit, pour quelques points pour lesquels l'inférence de l'état discret est ambigüe.



(a) Sans bruit



(b) Avec bruit : RSB = 30 dB.

FIGURE 4.2 – Convergence de l'algorithme d'identification de la sous-section 4.3.4.

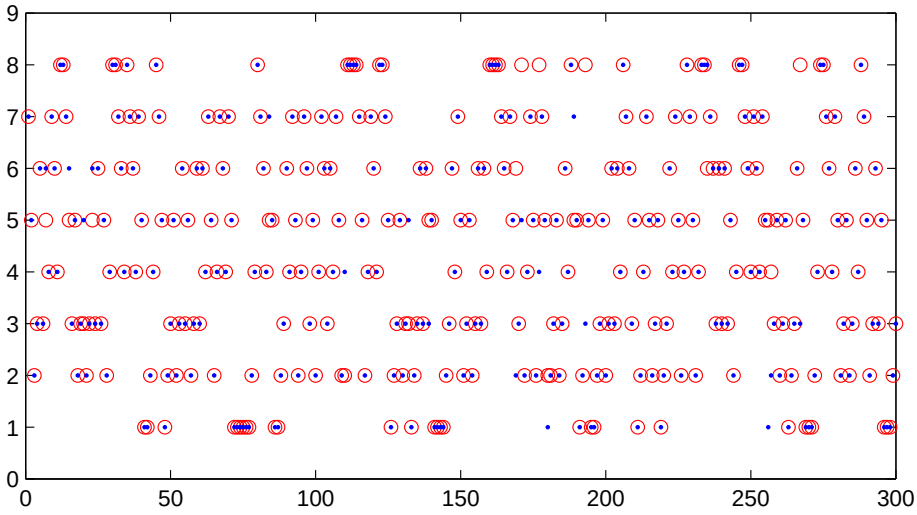


FIGURE 4.3 – Etat discret estimé en présence de bruit (cercle) et état discret réel (point) sur l'intervalle [0, 300].

Maintenant nous donnons les estimées obtenues par l'algorithme de la page 114, pour les matrices $\mathcal{F}_j$, $j = 1, \ldots, s$.

$$\hat{\mathcal{F}}_1 = \left[\begin{array}{cc|ccc} 1.2784 & -0.7094 & -0.9124 & 0.5852 & -0.6788 \\ 0.2784 & 0.2906 & -0.9124 & 0.5852 & -0.6788 \\ -0.2784 & 0.7094 & 0.9124 & -0.5852 & 0.6788 \end{array}\right]$$

$$\hat{\mathcal{F}}_2 = \left[\begin{array}{cc|ccc} 1.2661 & -0.8055 & -0.8815 & 0.9332 & -0.2356 \\ 0.2661 & 0.1945 & -0.8815 & 0.9332 & -0.2356 \\ -0.2661 & 0.8055 & 0.8815 & -0.9332 & 0.2356 \end{array}\right]$$

$$\hat{\mathcal{F}}_3 = \left[\begin{array}{cc|ccc} 1.2222 & -0.4603 & -0.6993 & -0.6330 & -0.9156 \\ 0.2222 & 0.5397 & -0.6993 & -0.6330 & -0.9156 \\ -0.2222 & 0.4603 & 0.6993 & 0.6330 & 0.9156 \end{array}\right]$$

$$\hat{\mathcal{F}}_4 = \left[\begin{array}{cc|ccc} 1.3819 & -0.5896 & -1.2202 & -1.3135 & -0.3934 \\ 0.3819 & 0.4104 & -1.2202 & -1.3135 & -0.3934 \\ -0.3819 & 0.5896 & 1.2202 & 1.3135 & 0.3934 \end{array}\right].$$

$$\hat{\mathcal{F}}_5 = \left[\begin{array}{cc|ccc} 0.7149 & -0.1613 & 0.1949 & -0.0836 & -0.6731 \\ -0.2851 & 0.8387 & 0.1949 & -0.0836 & -0.6731 \\ 0.2851 & 0.1613 & -0.1949 & 0.0836 & 0.6731 \end{array}\right]$$

$$\hat{\mathcal{F}}_6 = \left[\begin{array}{cc|ccc} 0.7243 & -0.2765 & 0.1875 & 0.3071 & -0.2234 \\ -0.2757 & 0.7235 & 0.1875 & 0.3071 & -0.2234 \\ 0.2757 & 0.2765 & -0.1875 & -0.3071 & 0.2234 \end{array}\right]$$

$$\hat{\mathcal{F}}_7 = \left[\begin{array}{cc|ccc} 0.8591 & -0.1847 & 0.0880 & -0.6887 & -0.8137 \\ -0.1409 & 0.8153 & 0.0880 & -0.6887 & -0.8137 \\ 0.1409 & 0.1847 & -0.0880 & 0.6887 & 0.8137 \end{array}\right]$$

$$\hat{\mathcal{F}}_8 = \left[\begin{array}{cc|ccc} 0.7850 & -0.1423 & 0.1731 & -1.2439 & -0.3292 \\ -0.2150 & 0.8577 & 0.1731 & -1.2439 & -0.3292 \\ 0.2150 & 0.1423 & -0.1731 & 1.2439 & 0.3292 \end{array}\right]$$

On peut noter que les estimées $\hat{\mathcal{F}}_j$ sont presque égales aux matrices réelles $\mathcal{F}_j$. De plus, il est intéressant de remarquer que les structures des matrices $\hat{\mathcal{F}}_j$ correspondent parfaitement à celles des vraies matrices respectives.

Maintenant, il nous faut retrouver une réalisation du système à partir des $\hat{\mathcal{F}}_j$. Grâce à la formule (4.32), les matrices C_j, $j = 1, \ldots, 8$, peuvent être directement extraites comme respectivement la première ligne de $\hat{\mathcal{F}}_j$, $j = 1, \ldots, 8$.

$$\hat{C}_1 = \begin{bmatrix} 1.2784 & -0.7094 \end{bmatrix}, \quad \hat{C}_2 = \begin{bmatrix} 1.2661 & -0.8055 \end{bmatrix}, \quad \hat{C}_3 = \begin{bmatrix} 1.2222 & -0.4603 \end{bmatrix},$$

$$\hat{C}_4 = \begin{bmatrix} 1.3819 & -0.5896 \end{bmatrix}, \quad \hat{C}_5 = \begin{bmatrix} 0.7149 & -0.1613 \end{bmatrix}, \quad \hat{C}_6 = \begin{bmatrix} 0.7243 & -0.2765 \end{bmatrix},$$

$$\hat{C}_7 = \begin{bmatrix} 0.8591 & -0.1847 \end{bmatrix}, \quad \hat{C}_8 = \begin{bmatrix} 0.7850 & -0.1423 \end{bmatrix}.$$

Par contre, pour estimer $\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{D}_{p_t}$, il est nécessaire, en se basant sur les matrices $\hat{\mathcal{F}}_j$, de reconstruire d'abord l'état discret q_t sur une certaine fenêtre temporelle (arbitraire) sur laquelle tous les 8 sous-modèles ont été visités. Pour permettre réellement l'extraction des matrices, la reconstruction de l'état discret a besoin d'être exacte. Cependant, cette condition est difficile à réaliser en pratique pour au moins deux raisons essentielles :

- La détermination de l'état discret q_t est une tâche de classification des données $\varphi(t) = \begin{bmatrix} (\Lambda y_f(t))^\top & u_f(t)^\top \end{bmatrix}^\top$ selon leur sous-modèle générateur respectif. On peut à cet effet, utiliser un critère $J(\hat{\mathcal{F}}_j, \varphi(t))$ semblable par exemple à (4.28). Mais ce critère étant défini à partir de matrices estimées $\hat{\mathcal{F}}_j$ (qui comportent probablement des incertitudes), peut être source d'erreur de classification.
- Même dans le cas où les matrices $\hat{\mathcal{F}}_j$ sont parfaitement connues, c'est-à-dire égales aux vraies $\mathcal{F}_j$, une formule du type (4.28) ne permet pas nécessairement d'inférer l'état discret de façon unique. Par exemple, il peut exister deux états i et j, $i \neq j$, et un certain indice temporel tels que $J(\hat{\mathcal{F}}_i, \varphi(t)) = J(\hat{\mathcal{F}}_j, \varphi(t)) = 0$. Dans ce cas, la détermination de l'état discret est ambigüe à moins d'exclure explicitement ces types de situations en formulant l'hypothèse que pour tout t, il existe un unique $j \in \{1, \ldots, |\mathscr{D}_{f+1}|\}$ vérifiant $y_f(t) = \mathcal{F}_j \varphi(t)$.

Ici, en vue d'obtenir les matrices $\bar{A}_{p_t}, \bar{B}_{p_t}, \bar{D}_{p_t}$, nous reconstruisons l'état discret q_t sur les 100 premiers échantillons. Puisque les commutations sont supposées refléter la configuration la plus générale, p_t peut prendre $|\mathscr{D}_4| = s^4 = 16$ valeurs distinctes tandis que q_t prend au plus $|\mathscr{D}_3| = s^3 = 8$ valeurs différentes.

La matrice $\hat{D}_{p_t}$ prend les deux valeurs suivantes :

$$\hat{D}_1 = 1.2067, \qquad \hat{D}_2 = 0.4988.$$

Les 16 valeurs de la matrices $\hat{A}_{p_t}$ sont données (dans le désordre) par :

$$\begin{bmatrix} -0.1724 & 1.2889 \\ 0.1094 & 0.6584 \end{bmatrix}, \begin{bmatrix} 0.2098 & 0.6294 \\ 0.7800 & -0.0481 \end{bmatrix}, \begin{bmatrix} -0.2049 & 1.0327 \\ -0.7618 & 1.4515 \end{bmatrix}, \begin{bmatrix} -0.3693 & 1.3631 \\ -0.9262 & 1.7819 \end{bmatrix},$$

$$\begin{bmatrix} -0.0602 & 0.7001 \\ -0.5924 & 1.3111 \end{bmatrix}, \begin{bmatrix} 0.3571 & 0.6577 \\ -0.4067 & 0.8369 \end{bmatrix}, \begin{bmatrix} -0.1407 & 1.4311 \\ 0.1412 & 0.8006 \end{bmatrix}, \begin{bmatrix} 0.2719 & 0.9516 \\ -0.1725 & 0.8723 \end{bmatrix},$$

$$\begin{bmatrix} 0.3781 & 0.6362 \\ 0.9483 & -0.0413 \end{bmatrix}, \begin{bmatrix} 0.4251 & 0.6783 \\ -0.3387 & 0.8575 \end{bmatrix}, \begin{bmatrix} -0.2394 & 1.1762 \\ 0.1907 & 0.4608 \end{bmatrix}, \begin{bmatrix} -0.2011 & 1.0759 \\ 0.2290 & 0.3606 \end{bmatrix},$$

$$\begin{bmatrix} 0.0623 & 0.5805 \\ 0.6136 & 0.1335 \end{bmatrix}, \begin{bmatrix} 0.0277 & 0.6245 \\ -0.5045 & 1.2354 \end{bmatrix}, \begin{bmatrix} -0.0287 & 0.6795 \\ 0.5226 & 0.2325 \end{bmatrix}, \begin{bmatrix} 0.2218 & 1.1560 \\ -0.2226 & 1.0767 \end{bmatrix}.$$

Les 16 valeurs prises par la matrice $\hat{B}_{p_t}$ sont données (dans le désordre) par :

$$\begin{bmatrix} 3.2438 \\ 1.7403 \end{bmatrix}, \begin{bmatrix} 0.3492 \\ 1.2989 \end{bmatrix}, \begin{bmatrix} 0.8883 \\ -0.0092 \end{bmatrix}, \begin{bmatrix} 2.3000 \\ 3.4366 \end{bmatrix}, \begin{bmatrix} 3.6232 \\ 2.1197 \end{bmatrix}, \begin{bmatrix} 1.1832 \\ 0.6272 \end{bmatrix}, \begin{bmatrix} 0.2650 \\ 0.8074 \end{bmatrix}, \begin{bmatrix} 1.1567 \\ 0.6118 \end{bmatrix},$$

$$\begin{bmatrix} 2.5688 \\ 0.8731 \end{bmatrix}, \begin{bmatrix} 1.0584 \\ 0.5105 \end{bmatrix}, \begin{bmatrix} 1.1937 \\ 0.6457 \end{bmatrix}, \begin{bmatrix} 0.4672 \\ 1.0096 \end{bmatrix}, \begin{bmatrix} 2.7987 \\ 1.1135 \end{bmatrix}, \begin{bmatrix} 1.6209 \\ 3.2152 \end{bmatrix}, \begin{bmatrix} 0.9068 \\ 0.0104 \end{bmatrix}, \begin{bmatrix} 1.8187 \\ 3.4130 \end{bmatrix}.$$

Il apparaît à l'issue de cet exemple, que l'estimation directe d'un modèle d'état commutant du type (4.1), à partir de données entrée-sortie, est très complexe. La difficulté de l'identification de ce modèle tient essentiellement à

- la nécessité d'éliminer l'état continu $x(t)$ dans les équations du système en l'exprimant sur une fenêtre finie de taille $f \geq \nu$,
- la nécessité de formuler une hypothèse forte d'observabilité complète sur tout horizon fini $[t,\, t+f-1]$,
- l'augmentation exponentielle du nombre d'états discrets (de l'ordre de s^f) en fonction du nombre d'états discrets s du modèle (4.1) et de l'horizon f considéré,
- la difficulté très évidente à estimer un nombre de sous-modèles aussi élevé que dans l'équation (4.23) puis à en extraire des matrices du système,
- l'obtention d'un modèle non minimal (en termes de nombre de sous-modèles) dont il est fastidieux de retrouver une représentation minimale.

En conclusion, la formulation précédente du problème d'identification de modèles d'état commutants (SLC) s'est avérée numériquement très coûteuse (cf. Tableau 4.1). Cette complexité est liée principalement au fait que l'état continu du système est inconnu et que les commutations sont supposées être potentiellement très rapides. Pour estimer l'état $x(t)$, on est alors amené à considérer une fenêtre de taille valant au moins l'indice d'observabilité du système. Nous avons ainsi converti le modèle récurrent d'état initial (4.1) en des relations entrée-sortie comme (4.17), (4.20) ou (4.23), exprimant des commutations entre un nouvel ensemble d'états discrets de cardinalité importante. Malheureusement, ce nouveau nombre d'états discrets augmente exponentiellement en fonction du nombre d'états discrets du modèle

initial (4.1) et de son indice d'observabilité. Ainsi, la solution présentée est sévèrement limitée à des systèmes de petites dimensions. Un autre problème important est lié à la notion de modèle minimal pour des systèmes hybrides. En l'absence d'une formulation claire de propriétés éventuelles pouvant caractériser un tel modèle (s'il existe), l'estimation du modèle (4.1) à partir de données entrée-sortie reste ambiguë puisque plusieurs modèles de cette forme (avec des ordres et des nombres d'états discrets différents) peuvent reproduire ces données.

4.4 Commutations avec temps de séjour

4.4.1 Seconde formulation du problème d'identification

Les commutations considérées dans la section 4.3 sont arbitraires dans le sens où le système peut commuter à tout instant d'un état discret à un quelconque autre état discret. Cependant, en pratique de tels systèmes pourraient souffrir de sévères problèmes de stabilité. Ainsi, il pourrait être d'un intérêt pratique de relaxer cette hypothèse. Nous supposons donc dans cette section que les instants de commutations sont séparés par un certain temps minimum τ_{dwell} que nous appelons temps de séjour. Cela nous permet de considérer l'identification d'un modèle plus complexe (cf. Eq. (4.35)) que celui représenté par l'équation (4.1). Par exemple, nous pouvons supposer maintenant que les différents sous-modèles peuvent être d'ordres inégaux.

Dans le chapitre 3, nous avons étudié le problème de l'identification d'un seul système linéaire à partir de ses données entrée-sortie aussi bien hors ligne que dans un contexte en ligne. Quelques nouvelles méthodes d'identification structurée des sous-espaces ont été introduites pour l'estimation de l'ordre ainsi que des paramètres d'un système linéaire MIMO. L'une de ces méthodes a été employée dans la sous-section 4.3.3 pour l'identification de modèles commutants dans le contexte général où les commutations peuvent être arbitrairement rapides. Nous allons maintenant généraliser plus particulièrement la méthode de la Section 3.4 du chapitre 3 à l'identification des SLC. Possédant la propriété de permettre une fixation de la base d'état, cette méthode semble bien appropriée à l'identification des SLC.
Le modèle de SLC considéré est le modèle d'état à temps discret suivant

$$\begin{cases} x(t+1) = A_{\lambda_{t+1},\lambda_t} x(t) + B_{\lambda_{t+1},\lambda_t} u(t) \\ \quad y(t) = C_{\lambda_t} x(t) + D_{\lambda_t} u(t) + v(t), \end{cases} \tag{4.35}$$

où $u(t) \in \mathbb{R}^{n_u}$, $y(t) \in \mathbb{R}^{n_y}$ et $x(t) \in \mathbb{R}^{n_{\lambda_t}}$ sont respectivement les vecteurs d'entrée, de sortie et d'état. Les indices $\lambda_t \in \{1, 2, \ldots\}$ désignent l'état discret qui est supposé être une séquence déterministe mais inconnue, n_{λ_t} est la dimension du processus d'état à l'instant t, $A_{\lambda_{t+1},\lambda_t} \in \mathbb{R}^{n_{\lambda_{t+1}} \times n_{\lambda_t}}$, $B_{\lambda_{t+1},\lambda_t} \in \mathbb{R}^{n_{\lambda_{t+1}} \times n_u}$, $C_{\lambda_t} \in \mathbb{R}^{n_y \times n_{\lambda_t}}$, $D_{\lambda_t} \in \mathbb{R}^{n_y \times n_u}$ sont les matrices du système à l'instant t et $\{v(t)\} \in \mathbb{R}^{n_y}$ représente un processus de bruit blanc centré. Les processus stochastiques $\{u(t)\}$, $\{x(t)\}$, $\{y(t)\}$ et $\{v(t)\}$ indexés par l'ensemble $\mathbb{Z}$ des entiers, sont supposés ergodiques et stationnaires [74]. Nous supposons aussi que les hypothèses A1-A4 de la section 3.4 sont vérifiées par chaque sous-système du système (4.35).
Par analogie avec la définition des Generalized Jump Markov Linear Systems (GJMLS) [96], le système

(4.35) peut être interprété comme un système linéaire commutant généralisé. Ici, les occurrences de deux commutations consécutives sont supposées être raisonnablement séparées dans le temps, de sorte que les matrices rectangulaires du type $A_{\lambda_{t+1},\lambda_t}$ n'apparaissent qu'assez rarement. Ainsi, les matrices $A_{\lambda_{t+1},\lambda_t}$ sont principalement carrées sauf aux quelques instants de commutation où un changement d'ordre se produit.
Nous notons les matrices $A_{\lambda_{t+1},\lambda_t}$ et $B_{\lambda_{t+1},\lambda_t}$ respectivement par A_{λ_t} et B_{λ_t} dans le cas où les états $x(t+1)$ et $x(t)$ sont de même dimension. De cette façon, pour $\lambda_t = j$, on peut utiliser simplement (A_j, B_j, C_j, D_j) et n_j pour désigner les matrices et l'ordre du $j$ème sous-modèle. Mais quand $x(t+1)$ et $x(t)$ ont des dimensions différentes on peut supposer que les matrices de transition $A_{\lambda_{t+1},\lambda_t}$ sont par exemple de la forme

$$\begin{aligned} A_{\lambda_{t+1},\lambda_t} &= \mathcal{T}_{\lambda_{t+1},\lambda_t} A_{\lambda_t} \\ B_{\lambda_{t+1},\lambda_t} &= \mathcal{T}_{\lambda_{t+1},\lambda_t} B_{\lambda_t} \end{aligned} \tag{4.36}$$

avec

$$\mathcal{T}_{\lambda_{t+1},\lambda_t} = \begin{cases} \begin{bmatrix} I_{n_{\lambda_{t+1}}} & 0 \end{bmatrix}, \text{ si } n_{\lambda_{t+1}} < n_{\lambda_t}, \\ \begin{bmatrix} I_{n_{\lambda_t}} \\ 0 \end{bmatrix}, \text{ si } n_{\lambda_{t+1}} > n_{\lambda_t}, \\ I_{n_{\lambda_t}}, \text{ si } n_{\lambda_{t+1}} = n_{\lambda_t}. \end{cases} \tag{4.37}$$

Problème 4.2. *Etant donné des observations $\{u(t)\}_{t=1}^{\infty}$ et $\{y(t)\}_{t=1}^{\infty}$ des processus d'entrée et de sortie générées par un modèle tel que (4.35), estimer récursivement les paramètres $\left(A_j, B_j, C_j, D_j\right)$, le nombre et les ordres des sous-modèles $\{n_j\}_{j=1,2,\ldots}$.*

Afin d'accomplir proprement cette tâche, nous faisons l'hypothèse que chaque fois que le système visite un état discret λ_t donné, il y demeure pendant une certaine durée minimum appelée temps de séjour τ_{dwell}. Par ailleurs la méthode que nous allons présenter ne requiert pas que le nombre de sous-modèles soit fini ni que les ordres $\{n_j\}$ soient égaux. De plus, aucune contrainte autre que celle du temps de séjour minimum n'est imposée sur le mécanisme de commutation.

4.4.2 Identification en ligne des sous-modèles

Munis de l'hypothèse de temps de séjour minimum, nous proposons d'appliquer en ligne la méthode d'identification développée au Chapitre 3 (Section 3.4), à l'estimation et à la labelisation en ligne des paramètres des sous-modèles au fur et à mesure que ces sous-modèles sont visités par le système. En procédant ainsi, il n'est pas nécessaire que le nombre de sous-modèles soit fini ou connu. De plus, cette stratégie permet de suivre d'éventuelles évolutions des paramètres des sous-modèles qui composent le système. Un autre avantage de notre méthode est qu'elle permet d'éviter que certains modes du système soient purement ignorés comme cela peut être le cas avec les méthodes qui procèdent hors ligne. En effet, puisque ces dernières méthodes opèrent sur une collection achevée de données, tout mode qui n'aurait pas été visité par le système durant la collection des données ne peut être détecté par un

algorithme d'identification. En revanche, une limitation apparente de notre méthode est la nécessité d'un temps de séjour minimum dans les états discrets. En pratique, l'hypothèse de temps de séjour (exprimé en fonction de la période d'échantillonnage) n'est pas aussi restrictive qu'elle peut paraître car, de très rapides commutations pourraient induire de sévères problèmes de stabilité. Ainsi, l'hypothèse de temps de séjour en elle-même ne constitue pas une limitation forte. Ce qui importe c'est quel doit être l'ordre de grandeur minimal de ce temps de séjour minimum pour garantir la validité de l'algorithme d'identification.

La solution proposée ici au problème 4.2 consiste à identifier *en ligne* les paramètres des différents sous-modèles du système ainsi que les ordres (qui sont potentiellement différents d'un sous-modèle à un autre) de ces sous-modèles et leur nombre.

Nous proposons d'appliquer l'algorithme 5.2 (chapitre 3, section 3.4) à l'identification du système (4.35). Concrètement, nous construisons incrémentalement grâce à l'identification, une batterie de sous-modèles continuellement mise à jour et qui servira à reconnaître les occurrences futures d'un même mode de fonctionnement. Comme nous le verrons, cela comporte trois sous-problèmes principaux :

- l'estimation de l'ordre et des paramètres du sous-modèle courant : il s'agit de l'extraction des paramètres d'un modèle linéaire sur un intervalle de temps suffisamment long où un seul sous-modèle est actif. Cela peut être naturellement réalisé grâce à l'un des algorithmes du chapitre 3 et aussi grâce à l'algorithme discuté dans la sous-section 4.3.3. Nous utiliserons à titre indicatif l'algorithme du propagateur étendu de la section 3.4.
- la détection des instants de commutation : cette tâche fait référence au problème qui consiste à reconnaître dans un délai relativement bref qu'une commutation s'est produite. Bien évidemment, la caractérisation précise de l'instant réel de la commutation est très problématique dans un contexte bruité. En ce qui nous concerne, une imprécision dans la caractérisation des instants de commutation n'a pas d'incidence réelle sur le résultat, du moins tant que la détection se fait raisonnablement vite après l'occurrence réelle de la commutation. Les instants de commutation sont détectés grâce à l'algorithme d'estimation de l'ordre.
- la classification des sous-modèles identifiés : puisque la procédure d'identification permet d'enregistrer progressivement les paramètres des différents sous-modèles à mesure que ceux-ci apparaissent, il convient de se demander si le modèle courant existe déjà dans la base de sous-modèles enregistrés (auquel cas on devrait le fusionner avec son représentant déjà connu) ou s'il s'agit d'un nouveau sous-modèle, c'est-à-dire non encore visité par le système (auquel cas le nombre de sous-modèles devrait être incrémenté de un). Cette classification passe bien entendu par la définition d'une métrique appropriée mesurant la proximité entre des systèmes dynamiques et la fixation d'un seuil de décision en fonction du niveau du bruit dans les données d'apprentissage.

Globalement, l'identification d'un modèle commutant du type (4.35) diffère essentiellement de celle d'un seul modèle linéaire par la gestion que l'on doit faire des instants de commutations. Ainsi, nous commençons par étudier au voisinage d'une commutation, les effets d'une commutation sur la valeur de l'ordre estimé par l'algorithme 3.1 (cf. Chapitre 3, Section 3.4).

Pour ce faire, nous faisons, pour des raisons de lisibilité, un bref rappel sur le principe de l'algorithme de

base qui sera employé pour réaliser l'identification du système commutant (cf. Chapitre 3). Supposons que sur un certain intervalle de temps $[t_1,\ t_2]$, l'état discret λ_t soit constant et égal à un certain i. Alors, en combinant linéairement les sorties, nous transformons le système MIMO (4.35) en un système MISO ayant le même processus d'état comme suit :

$$\begin{cases} x(t+1) = A_i x(t) + B_i u(t) \\ \quad y(t) = C_i x(t) + D_i u(t) + v(t), \end{cases} \tag{4.38a}$$

$$\Downarrow$$

$$\begin{cases} x(t+1) = A_i x(t) + B_i u(t) \\ \quad y_a(t) = \bar{c}_i^\top x(t) + \bar{d}_i^\top u(t) + \gamma^\top v(t), \end{cases} \tag{4.38b}$$

où $y_a(t) = \gamma^\top y(t)$, $\bar{c}_i^\top = \gamma^\top C_i$ et $\bar{d}_i^\top = \gamma^\top D_i$. Il suffit alors que le système (4.38b) soit observable c'est-à-dire que $\Gamma_n(A_i, \bar{c}_i^\top) = \Gamma_n(A_i, \gamma^\top C_i)$ soit de rang plein n_i pour qu'il soit possible d'estimer l'état et donc l'ordre (qui est commun aux deux formes de modèles (4.38a) et (4.38b)) à partir des entrées-sorties du modèle (4.38b). L'avantage de cette transformation est que le modèle (4.38b) étant MISO et observable, il peut se mettre dans la base canonique d'observabilité. Dans la même base, on peut ensuite calculer les paramètres de (4.38a).

Maintenant, nous introduisons comme au Chapitre 3, les notations $H_q^i = H_q(A_i, C_i, B_i, D_i)$ et $\Gamma_q^i = \Gamma_q(A_i, C_i)$. De même, soient $\Psi_f^i = \Gamma_f\big(A_i, \bar{c}_i^\top\big) \in \mathbb{R}^{f\times n_i}$, $\mathcal{H}_f^i = H_f\big(A_i, B_i, \bar{c}_i^\top, \bar{d}_i^\top\big) \in \mathbb{R}^{f\times fn_u}$, où $\bar{c}_i^\top = \gamma^\top C_i$, $\bar{d}_i^\top = \gamma^\top D_i$ et $\gamma \in \mathbb{R}^{n_y}$ est tel que rang $\big(\Gamma_n(A_i, \gamma^\top C_i) = n_i$, $i = 1, 2, \ldots$ Considérons aussi des vecteurs $u_f(t) \in \mathbb{R}^{fn_u}$ et $y_f(t) \in \mathbb{R}^{fn_y}$ définis comme en (4.9). Pour une commodité de notations, nous définissons également

$$\begin{aligned} \Omega_{n_i}^i &= \begin{bmatrix} A_i^{n_i-1}B_i & \cdots & A_iB_i & B_i \end{bmatrix} \\ \mathcal{R}^i &= \begin{bmatrix} a_0^i & \cdots & \cdots & a_{n_i-1}^i \\ 0 & a_0^i & \cdots & a_{n_i-2}^i \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_0^i \end{bmatrix} \quad \text{et} \quad \mathcal{S}^i = \begin{bmatrix} 1 & & & \\ a_{n_i-1}^i & 1 & 0 & \\ \vdots & \vdots & \ddots & \\ a_1^i & \cdots & \cdots & 1 \end{bmatrix}, \end{aligned} \tag{4.39}$$

où $a_0^i, \cdots, a_{n_i-1}^i$ sont les coefficients du polynôme caractéristique de A_i. Finalement, en se servant des matrices (4.39), nous définissons

$$\mathcal{M}^{ij} = \begin{bmatrix} \left(\mathcal{R}^i\Psi_{n_i}^i + \mathcal{S}^i\Psi_{n_i}^j T_{j,i} A_i^{n_i}\right) & \mathcal{S}^i\Big[\left(\Psi_{n_i}^j T_{j,i} - \Psi_{n_i}^i\right)\Omega_{n_i}^i & \left(\mathcal{H}_{n_i}^j - \mathcal{H}_{n_i}^i\right)\Big] \end{bmatrix}, \tag{4.40}$$

où nous rappelons que $T_{j,i}$ est la matrice de transition (4.37) du sous-modèle i vers le sous-modèle j.

En revenant à notre problème de détection, rappelons que nous envisageons de construire un indicateur du changement de mode. Cela peut bien sûr être réalisé de multiples façons [14]. Par exemple, on peut choisir de suivre l'évolution de la variance des paramètres identifiés en ligne. Après convergence, ceux-ci

devraient se stabiliser, de sorte que l'occurrence d'une commutation s'accompagnera d'un saut plus ou moins détectable dans les valeurs prises par cette variance selon que les deux sous-modèles entre lesquels s'effectue la commutation sont fortement distincts et selon aussi le niveau de bruit. Au delà de cette possibilité, nous allons présenter ci-dessous un résultat théorique selon lequel l'estimateur de l'ordre courant peut servir de détecteur de commutation.

Avant la commutation. Considérons que seul le sous-modèle i est actif sur l'intervalle de temps $[\tau - \tau_{\text{dwell}},\, \tau - 1]$, où τ est un instant de commutation et τ_{dwell} désigne le temps de séjour. Supposons que le bruit $\{v(t)\}$ soit identiquement nul dans (4.35). L'objectif étant d'étudier les changements éventuels induits par l'occurrence d'une commutation dans l'ordre du système tel qu'estimé l'algorithme 3.1, nous pouvons nous focaliser sur la sortie combinée $y_a(t)$ définie en (4.38b). Définissons $\tau^o = \tau - \tau_{\text{dwell}}$, un entier f satisfaisant $\max(n_j) < f \ll \tau_{\text{dwell}}$, et

$$\mathcal{X}_{\tau^o|\bar{t}} = \begin{bmatrix} x(\tau^o) & \cdots & x(\bar{t}) \end{bmatrix} \in \mathbb{R}^{n_i \times (\bar{t}-\tau^o+1)}, \tag{4.41}$$

$$\mathcal{U}_{\tau^o|\bar{t},f} = \begin{bmatrix} u_f(\tau^o) & \cdots & u_f(\bar{t}) \end{bmatrix} \in \mathbb{R}^{fn_u \times (\bar{t}-\tau^o+1)}, \tag{4.42}$$

$$\mathcal{Y}_{\tau^o|\bar{t},f} = \begin{bmatrix} y_{a,f}(\tau^o) & \cdots & y_{a,f}(\bar{t}) \end{bmatrix} \in \mathbb{R}^{f \times (\bar{t}-\tau^o+1)}, \tag{4.43}$$

où $\bar{t} = t - f + 1$, t est l'instant présent, $y_a(t)$ est la sortie combinée définie par (4.38b) et $u_f(k)$ et $y_{a,f}(k)$, $k = \tau^o, \cdots, \bar{t}$, sont définis comme dans (4.9) à partir respectivement de $u(t)$ et $y_a(t)$. Supposons maintenant que l'hypothèse suivante, relative à la persistance d'excitation, est vérifiée.

Hypothèse 4.3. *Le signal d'entrée $\{u(t)\}$ est suffisamment excitant d'ordre f pour le sous-modèle i dans le sens où il existe $\bar{t} \in [\tau^o, \tau - f]$, avec $f \geq n_i$, tel que*[1]

$$\text{rang}\left(\begin{bmatrix} \mathcal{X}_{\tau^o|\bar{t}} \\ \mathcal{U}_{\tau^o|\bar{t},f} \end{bmatrix}\right) = fn_u + n_i.$$

A partir du système (4.38b), nous pouvons écrire l'équation de données

$$\mathcal{Y}_{\tau^o|\bar{t},f} = \Psi_f^i \mathcal{X}_{\tau^o|\bar{t}} + \mathcal{H}_f^i \mathcal{U}_{\tau^o|\bar{t},f}. \tag{4.44}$$

Considérons un instant $t = \bar{t} + f - 1$, avec $\bar{t}$ tel que l'hypothèse 4.3 soit satisfaite. En utilisant le théorème de Cayley-Hamilton [57], nous savons que les $f - n_i$ dernières lignes de Ψ_f^i peuvent s'exprimer comme une combinaison de ses n_i premières lignes, c'est-à-dire, on peut trouver une matrice $\mathcal{P}^i \in \mathbb{R}^{(f-n_i)\times n_i}$ telle que $\Psi_f^i(n_i+1:f,:) = \mathcal{P}^i \Psi_{n_i}^i$ (cf. Chapitre 3, Sous-section 3.3.2). Ainsi, en multipliant l'équation (4.44) à droite par $\Pi^{\perp}_{\mathcal{U}_{\tau^o|\bar{t},f}} \Lambda^{1/2}_{\tau^o|\bar{t}}$, où $\Lambda_{\tau^o|\bar{t}} = \text{diag}\left(\lambda^{\bar{t}-\tau^o}, \ldots, \lambda, 1\right)$ et λ est le facteur d'oubli, on obtient

$$\begin{bmatrix} \mathcal{Y}_{\tau^o|\bar{t},n_i} \\ \mathcal{Y}_{\tau^o+n_i|\bar{t}+n_i,f-n_i} \end{bmatrix} \Pi^{\perp}_{\mathcal{U}_{\tau^o|\bar{t},f}} \Lambda^{1/2}_{\tau^o|\bar{t}} = \begin{bmatrix} I_{n_i} \\ \mathcal{P}^i \end{bmatrix} \bar{Z}_{\tau^o|\bar{t}}, \tag{4.45}$$

1. On peut en fait montrer (cf. Proposition 3.1) que si l'entrée $\{u(t)\}$ est SE$(f + n_i)$, alors cette relation est satisfaite.

avec $\bar{Z}_{\tau^o|\bar{t}} = \left(\Psi^i_{n_i}\mathcal{X}_{\tau^o|\bar{t}}\Pi^{\perp}_{\mathcal{U}_{\tau^o|\bar{t},f}}\right)\Lambda^{1/2}_{\tau^o|\bar{t}}$. Notons que d'un point de vue théorique, la factorisation RQ utilisée dans la proposition 3.2 pour implémenter la projection orthogonale revient en fait à multiplier l'équation de données (4.44) par $\Pi^{\perp}_{\mathcal{U}_{\tau^o|\bar{t},f}}$ (cf. preuve de la proposition 3.2). Ainsi, la matrice $\Delta_f(t)$ définie comme au Chapitre 3, (Sous-section 3.4.3), peut être simplement obtenue comme le carré de (4.45), soit

$$\begin{aligned}\Delta_f(t) &= \Sigma(t)(1:f,1:f)\\ &= \left(\mathcal{Y}_{\tau^o|\bar{t},f}\Pi^{\perp}_{\mathcal{U}_{\tau^o|\bar{t},f}}\Lambda^{1/2}_{\tau^o|\bar{t}}\right)\left(\mathcal{Y}_{\tau^o|\bar{t},f}\Pi^{\perp}_{\mathcal{U}_{\tau^o|\bar{t},f}}\Lambda^{1/2}_{\tau^o|\bar{t}}\right)^{\top}\\ &= \begin{bmatrix}\bar{Z}_{\tau^o|\bar{t}}\bar{Z}^{\top}_{\tau^o|\bar{t}} & \bar{Z}_{\tau^o|\bar{t}}\bar{Z}^{\top}_{\tau^o|\bar{t}}(\mathcal{P}^i)^{\top}\\ \mathcal{P}^i\bar{Z}_{\tau^o|\bar{t}}\bar{Z}^{\top}_{\tau^o|\bar{t}} & \mathcal{P}^i\bar{Z}_{\tau^o|\bar{t}}\bar{Z}^{\top}_{\tau^o|\bar{t}}(\mathcal{P}^i)^{\top}\end{bmatrix}.\end{aligned} \tag{4.46}$$

Avec l'hypothèse 4.3, il apparaît clairement grâce à la Proposition 3.2 que $\text{rang}(\bar{Z}_{\tau^o|\bar{t}}) = n_i$. Ainsi, en utilisant l'équation précédente, on obtient aussi $\text{rang}(\Delta_f(t)) = \text{rang}(\Delta_{n_i}(t)) = n_i$ avant l'occurrence d'une commutation.

Après la commutation. Lorsque le système commute à l'instant τ d'un sous-modèle i à un sous-modèle j par exemple, l'équation (4.44) n'est plus satisfaite parce que la matrice de Hankel formée avec les sorties contiendra des observations provenant de deux sous-modèles différents. Nous donnons ci-dessous une illustration des changements qui affectent l'équation du système durant la transition de i à j. Avec $\bar{t} = \tau - f$, on peut écrire sur l'horizon $[\tau - f,\ \tau + f - 1]$,

$$\begin{array}{lcccc}
\tau-1, & \left[\begin{array}{c} y_f(\bar{t}) \\ \hline -- \end{array}\right] & = & \left[\begin{array}{c} \Gamma^i_f x(\bar{t}) \\ \hline -- \end{array}\right] & + \quad \left[\begin{array}{c} H^i_f u_f(\bar{t}) \\ \hline -- \end{array}\right] \\
\tau, & \left[\begin{array}{c} y_{f-1}(\bar{t}+1) \\ \hline y_1(\bar{t}+f) \end{array}\right] & = & \left[\begin{array}{c} \Gamma^i_{f-1} x(\bar{t}+1) \\ \hline \Gamma^j_1 x(\bar{t}+f) \end{array}\right] & + \quad \left[\begin{array}{c} H^i_{f-1} u_{f-1}(\bar{t}) \\ \hline H^j_1 u_1(\bar{t}+f) \end{array}\right] \\
\vdots & \vdots & = & \vdots & + \quad \vdots \\
\tau+f-2, & \left[\begin{array}{c} y_1(\bar{t}+f-1) \\ \hline y_{f-1}(\bar{t}+f) \end{array}\right] & = & \left[\begin{array}{c} \Gamma^i_1 x(\bar{t}+f-1) \\ \hline \Gamma^j_{f-1} x(\bar{t}+f) \end{array}\right] & + \quad \left[\begin{array}{c} H^i_1 u_1(\bar{t}+f-1) \\ \hline H^j_{f-1} u_{f-1}(\bar{t}+f) \end{array}\right] \\
\tau+f-1, & \left[\begin{array}{c} -- \\ \hline y_f(\bar{t}+f) \end{array}\right] & = & \left[\begin{array}{c} -- \\ \hline \Gamma^j_f x(\bar{t}+f) \end{array}\right] & + \quad \left[\begin{array}{c} -- \\ \hline H^j_f u_f(\bar{t}+f) \end{array}\right].
\end{array}$$

Les indices temporels $\tau - 1$, τ, $\cdots$, $\tau + f - 2$ et $\tau + f - 1$ de la première colonne de ce tableau font référence aux derniers indices respectivement des vecteurs $y_f(\bar{t})$, $y_1(\bar{t}+f)$, $\cdots$, $y_{f-1}(\bar{t}+f)$ et $y_f(\bar{t}+f)$, avec $\bar{t} + f = \tau$.

Il apparaît comme un fait intuitif qu'une telle transition devrait occasionner une sur-estimation de l'ordre par l'algorithme 3.1. Cette idée est confortée par le fait que les données engendrées à la fois par les sous-

modèles i et j (ces derniers étant supposés assez distincts) ne peuvent plus être ajustées à un seul modèle linéaire. Ainsi, la structure particulière de la matrice $\Sigma(t)$ (cf. Eq. (4.46) et Section 3.4) qui permettait l'extraction de l'ordre se trouve maintenant cassée. Dans ce qui suit, nous nous attachons à dériver une condition sous laquelle la transition (décrite ci-dessus) induite par la commutation dans l'équation de données ne provoquerait pas en principe une modification de l'ordre estimé par l'algorithme 3.1.

Proposition 4.3 ([11])**.** *Soit τ un instant de commutation auquel le système commute d'un sous-modèle i à un sous-modèle j. Soient $\tau_o = \tau - n_i$, $\tau^o = \tau - \tau_{dwell}$, où τ_{dwell} est le temps de séjour minimum et considérons que le bruit $\{v(t)\}$ est identiquement nul dans l'équation (4.35). Supposons que*

- *seul le sous-modèle i a été actif sur $[\nu,\ \tau-1]$,*
- *l'hypothèse 4.3 est vérifiée pour un certain $\bar{t} \in [\nu,\ \tau-f+1]$, où f est strictement supérieur à tous les ordres des sous-modèles du système ($\max(n_q) < f \ll \tau_{dwell}$),*
- *chaque sous-modèle du système est individuellement minimal,*
- *le vecteur de pondération γ est tel que $\operatorname{rang}\left(\Psi^q_{n_q}\right) = n_q$ pour toute valeur q de l'état discret.*

Alors, $\operatorname{rang}\left(\Delta_{n_i+1}(t)\right) = \operatorname{rang}\left(\Delta_{n_i}(t)\right) = n_i$ *pour tout $t \in [\tau,\ \tau+f]$ si et seulement si*

$$\begin{bmatrix} x(\tau_o) \\ u_{2n_i}(\tau_o) \end{bmatrix} \in \operatorname{null}\left(\mathcal{M}^{ij}\right), \tag{4.47}$$

où $\Delta_r(t)$ est défini comme ci-dessus et $\mathcal{M}^{ij}$ est défini comme en (4.40), $x(\tau_o) \in \mathbb{R}^{n_i}$. La notation $\operatorname{null}\left(\mathcal{M}^{ij}\right)$ *fait référence à l'espace noyau de $\mathcal{M}^{ij}$.*

Preuve. Considérons $t \in [\tau,\ \tau+f]$ et définissons $\tau_o = \tau - n_i$ et $\tau_1 = \tau + n_i$. Puisque nous nous intéressons ici au rang de $\Delta_{n_i+1}(t)$, nous pouvons poser $f = l = n_i + 1$. Alors, considérons les équations du système commutant, écrites sur l'intervalle $[\nu,\ \tau_1]$:

$$\begin{aligned} \mathcal{Y}\Pi^{\perp}_{\mathcal{U}} &= \left[\begin{array}{c|c} \mathcal{Y}_{\nu|\tau_o-1,n_i} & \mathcal{Y}_{\tau_o|\tau-1,n_i} \\ \hline \mathcal{Y}_{\nu+n_i|\tau-1,1} & \mathcal{Y}_{\tau|\tau_1-1,1} \end{array}\right]\Pi^{\perp}_{\mathcal{U}} \\ &= \left[\begin{array}{c|c} \Psi^i_{n_i}\mathcal{X}_{\nu|\tau_o-1} & \Psi_{\text{mix}} \\ \hline \Psi^i_l(l,:)\mathcal{X}_{\nu|\tau_o-1} & \psi^{\top}_{\text{mix}} \end{array}\right]\Pi^{\perp}_{\mathcal{U}} + \left[\begin{array}{c|c} \mathcal{H}^i_{n_i}\mathcal{U}_{\nu|\tau_o-1,n_i} & \mathcal{H}_{\text{mix}} \\ \hline \mathcal{H}^i_l(l,:)\mathcal{U}_{\nu|\tau_o-1,l} & h^{\top}_{\text{mix}} \end{array}\right]\Pi^{\perp}_{\mathcal{U}}, \end{aligned} \tag{4.48}$$

où

$$\mathcal{U} = \mathcal{U}_{\nu|\tau-1,l} \in \mathbb{R}^{ln_u \times (\tau-\nu)}, \quad \mathcal{Y} = \mathcal{Y}_{\nu|\tau-1,l} \in \mathbb{R}^{l \times (\tau-\nu)},$$

$$
\begin{aligned}
\psi_{\text{mix}}^{\top} &= \begin{bmatrix} \bar{c}_j^{\top} x(\tau) & \bar{c}_j^{\top} A_j x(\tau) & \cdots & \bar{c}_j^{\top} A_j^{n_i-1} x(\tau) \end{bmatrix}, \\
h_{\text{mix}}^{\top} &= \begin{bmatrix} M_1^j u_1(\tau) & M_2^j u_2(\tau) & \cdots & M_{n_i}^j u_{n_i}(\tau) \end{bmatrix}, \\
\Psi_{\text{mix}} &= \begin{bmatrix} \begin{bmatrix} \Psi_{n_i}^i x(\tau_o) \\ -- \end{bmatrix}, & \begin{bmatrix} \Psi_{n_i}^i(2:n_i,:) x(\tau_o) \\ \Psi_1^j x(\tau) \end{bmatrix}, & \cdots, & \begin{bmatrix} \Psi_{n_i}^i(n_i,:) x(\tau_o) \\ \Psi_{n_i-1}^j x(\tau) \end{bmatrix} \end{bmatrix}, \\
\mathcal{H}_{\text{mix}} &= \begin{bmatrix} \begin{bmatrix} \mathcal{H}_{n_i}^i u_{n_i}(\tau_o) \\ -- \end{bmatrix}, & \begin{bmatrix} \mathcal{H}_{n_i}^i(2:n_i,:) u_{n_i}(\tau_o) \\ \mathcal{H}_1^j u_1(\tau) \end{bmatrix}, & \cdots, & \begin{bmatrix} \mathcal{H}_{n_i}^i(n_i,:) u_{n_i}(\tau_o) \\ \mathcal{H}_{n_i-1}^j u_{n_i-1}(\tau) \end{bmatrix} \end{bmatrix},
\end{aligned}
$$

avec $M_1^j = \bar{d}_j^{\top}$ et $M_q^j = \begin{bmatrix} \bar{c}_j^{\top} A_j^{q-2} B_j & \cdots & \bar{c}_j^{\top} B_j & \bar{d}_j^{\top} \end{bmatrix}$ pour $q \geq 2$. Précisons au passage que le dernier vecteur de la matrice $\mathcal{Y}$ qui apparaît dans l'équation (4.48) est construit de l'instant $\tau - 1$ à l'instant $\tau_1 - 1$. Donc le plus grand index temporel des données impliquées dans l'équation (4.48) est $\tau_1 - 1$.

Nous commençons par montrer que les n_i premières lignes de la matrice $\mathcal{Y}\Pi_{\mathcal{U}}^{\perp}$ dans la relation (4.48), sont linéairement indépendantes. L'hypothèse 4.3 est supposée satisfaite pour un certain $\bar{t} < \tau - f + 1$. Alors, les matrices de données $\mathcal{Y}_{\nu|\bar{t},l}$ et $\mathcal{U}_{\nu|\bar{t},l}$ sont générées par un modèle linéaire (en l'occurrence, le sous-modèle i), et nous pouvons donc utiliser la proposition 3.2 pour conclure que $\begin{bmatrix} \mathcal{Y}_{\nu|\bar{t},l} \\ \mathcal{U}_{\nu|\bar{t},l} \end{bmatrix}$ possède $n_i + ln_u$ colonnes linéairement indépendantes. En écrivant

$$
\begin{bmatrix} \mathcal{Y} \\ \mathcal{U} \end{bmatrix} = \left[\begin{array}{c|c} \mathcal{Y}_{\nu|\bar{t},l} & \mathcal{Y}_{\bar{t}+1|\tau-1,l} \\ \mathcal{U}_{\nu|\bar{t},l} & \mathcal{U}_{\bar{t}+1|\tau-1,l} \end{array}\right],
$$

il apparaît clairement que

$$
\text{rang}\left(\begin{bmatrix} \mathcal{Y} \\ \mathcal{U} \end{bmatrix}\right) \geq n_i + ln_u.
$$

En utilisant maintenant le lemme 3.1, on obtient facilement rang $\left(\mathcal{Y}\Pi_{\mathcal{U}}^{\perp}\right) \geq n_i$. Ce résultat étant vrai pour $l \geq n_i$, les n_i premières de $\mathcal{Y}\Pi_{\mathcal{U}}^{\perp} \in \mathbb{R}^{l \times (\tau-\nu)}$ sont indépendantes. D'autre part, notons que

$$
\Delta_l(\tau_1 - 1) = \left(\mathcal{Y}\Pi_{\mathcal{U}}^{\perp}\Lambda_{\tau-1}^{1/2}\right)\left(\mathcal{Y}\Pi_{\mathcal{U}}^{\perp}\Lambda_{\tau-1}^{1/2}\right)^{\top},
$$

où la matrice des poids $\Lambda_{\tau-1}$ défini comme ci-dessus est non singulière. Ainsi,
rang $(\Delta_l(\tau_1 - 1))$ = rang $\left(\mathcal{Y}\Pi_{\mathcal{U}}^{\perp}\right)$. Par conséquent, nous étudierons dans la suite plutôt le rang de $\mathcal{Y}\Pi_{\mathcal{U}}^{\perp}$ au lieu de celui de $\Delta_l(\tau_1 - 1)$.

Il reste maintenant à montrer que la (n_i+1)-ème ligne de $\mathcal{Y}\Pi_{\mathcal{U}}^{\perp}$ (et donc celle de $\Delta_l(\tau_1 - 1)$) est une combinaison linéaire de ses n_i premières ligne si et seulement si la relation (4.47) est satisfaite. Cela est équivalent à dire qu'il existe $\alpha \in \mathbb{R}^{n_i}$ tel que

$$
\begin{aligned}
\Big[\Psi_l^i(l,:)\mathcal{X}_{\nu|\tau_o-1} + \mathcal{H}_l^i(l,:)\mathcal{U}_{\nu|\tau_o-1,l} - \alpha^{\top}\left(\Psi_{n_i}^i\mathcal{X}_{\nu|\tau_o-1} + \mathcal{H}_{n_i}^i\mathcal{U}_{\nu|\tau_o-1,n_i}\right)\Big| \\
\psi_{\text{mix}}^{\top} + h_{\text{mix}}^{\top} - \alpha^{\top}\left(\Psi_{\text{mix}} + \mathcal{H}_{\text{mix}}\right)\Big]\Pi_{\mathcal{U}}^{\perp} = 0.
\end{aligned} \tag{4.49}
$$

Par suite, la matrice qui multiplie $\Pi_{\mathcal{U}}^{\perp}$ dans (4.49) peut être réécrite comme une combinaison linéaire des lignes de $\mathcal{U}$. Soit $\mathcal{U} = \left[\mathcal{U}^o \;\middle|\; \mathcal{U}^1\right]$, $\mathcal{U}^o = \mathcal{U}_{\nu|\tau_o-1,l}$, une partition de $\mathcal{U}$ conforme à celle de l'équation (4.48). Alors il

existe $r \in \mathbb{R}^{ln_u}$ tel que

$$\begin{cases} \Psi_l^i(l,:)\mathcal{X}_{\nu|\tau_o-1} - \alpha^\top \Psi_{n_i}^i \mathcal{X}_{\nu|\tau_o-1} + \left(\begin{bmatrix} -\alpha^\top \mathcal{H}_{n_i}^i & 0 \end{bmatrix} + \mathcal{H}_l^i(l,:)\right)\mathcal{U}^o = r^\top \mathcal{U}^o \\ \psi_{\text{mix}}^\top + h_{\text{mix}}^\top - \alpha^\top (\Psi_{\text{mix}} + \mathcal{H}_{\text{mix}}) = r^\top \mathcal{U}^1. \end{cases} \tag{4.50}$$

En utilisant le théorème de Cayley-Hamilton, on a $\Psi_l^i(l,:) = \bar{c}_i^\top A_i^{n_i} = -(a^i)^\top \Psi_{n_i}^i$, où $a^i = \begin{bmatrix} a_0^i & \cdots & a_{n_i-1}^i \end{bmatrix} \in \mathbb{R}^{n_i}$ est un vecteur formé avec les coefficients du polynôme caractéristique de A_i. Grâce à cette propriété, la première équation de (4.50) devient

$$-\left((a^i)^\top + \alpha^\top\right)\Psi_{n_i}^i \mathcal{X}_{\nu|\tau_o-1} + \left(\begin{bmatrix} -\alpha^\top \mathcal{H}_{n_i}^i & 0 \end{bmatrix} + \mathcal{H}_l^i(l,:)\right)\mathcal{U}^o = r^\top \mathcal{U}^o. \tag{4.51}$$

En multipliant cette relation par $\Pi_{\mathcal{U}^o}^\perp$, il vient

$$-\left((a^i)^\top + \alpha^\top\right)\Psi_{n_i}^i \mathcal{X}_{\nu|\tau_o-1}\Pi_{\mathcal{U}^o}^\perp = 0. \tag{4.52}$$

Puisque $\Psi_{n_i}^i \mathcal{X}_{\nu|\tau_o-1}\Pi_{\mathcal{U}^o}^\perp$ est de rang plein ligne, l'équation (4.52) permet de déduire que $\alpha = -a^i$. Alors, en retournant à l'équation (4.51), nous obtenons $r^\top = \begin{bmatrix} (a^i)^\top \mathcal{H}_{n_i}^i & 0 \end{bmatrix} + \mathcal{H}_l^i(l,:)$ parce que $\mathcal{U}^o$ est de rang plein ligne (cf. Hypothèse 4.3). Si l'on remplace maintenant les expressions de α et r dans la seconde équation de (4.50), on trouve après quelques réductions algébriques

$$\mathcal{S}^i\left(\Psi_{n_i}^j x(\tau) + (\mathcal{H}_{n_i}^j - \mathcal{H}_{n_i}^i)u_{n_i}(\tau)\right) + \mathcal{R}^i \Psi_{n_i}^i x(\tau_o) - \mathcal{S}^i \Psi_{n_i}^i \Omega_{n_i}^i u_{n_i}(\tau_o) = 0,$$

où $\mathcal{R}^i$, $\mathcal{S}^i$, $\Omega_{n_i}^i$ sont donnés par (4.39). Remarquons maintenant qu'on peut exprimer $x(\tau)$ comme

$$x(\tau) = T_{ji}\left(A_i^{n_i} x(\tau_o) + \Omega_{n_i}^i u_{n_i}(\tau_o)\right).$$

En portant cette expression dans l'équation précédente, nous obtenons le résultat (4.47) recherché. ■

Il est facile de vérifier que pour $i = j$, la condition (4.47) est satisfaite pour tout état $x(\tau_o)$, et toute séquence d'entrée $u_{2n_i}(\tau_o)$. Ainsi, pour que la commutation (de i vers j) soit détectable sur $[\tau,\ \tau + f]$ à travers une inspection du rang fourni par l'algorithme 3.1, on a besoin de supposer que les sous-modèles i et j sont suffisamment différents dans un certain sens. Plus précisément, les paramètres des sous-modèles i et j et l'entrée $\{u(t)\}$ du système sont tels que

$$\begin{bmatrix} x(\tau_o)^\top & u_{2n_i}(\tau_o)^\top \end{bmatrix}^\top \notin \text{null}\left(\mathcal{M}^{ij}\right). \tag{4.53}$$

Si cette condition est vérifiée pour toutes les commutations possibles du système, alors chaque commutation aura pour effet d'accroître le rang de $\Delta_f(t)$ selon la Proposition 4.3.
Cela signifie que tout changement dans les dynamiques ou les zeros du système seront alors (rigoureusement, dans le cas il n'y a pas de bruit dans les données, approximativement dans le cas contraire) détectables par notre algorithme [1] puisqu'un tel changement induit un accroissement de l'ordre (estimé) du système (cf. aussi [22]).

1. même si cette commutation ne s'accompagne pas nécessairement d'un changement d'ordre.

Remarque 4.5. *Notons que l'identification des matrices et de l'état discret du système est basée sur la sortie auxiliaire $y_a(t)$ qui est la sortie du système MISO*

$$\begin{cases} x(t+1) = A_{\lambda_{t+1},\lambda_t} x(t) + B_{\lambda_{t+1},\lambda_t} u(t) \\ y_a(t) = \bar{c}_{\lambda_t}^\top x(t) + \bar{d}_{\lambda_t}^\top u(t), \end{cases} \tag{4.54}$$

où $\bar{c}_{\lambda_t}^\top = \gamma^\top C_{\lambda_t}$, $\bar{d}_{\lambda_t}^\top = \gamma^\top D_{\lambda_t}$. Ainsi, il pourrait se produire un problème de discernabilité des sous-modèles. Pour le voir, considérons par exemple deux modes i et j avec des matrices (A_i, B_i, C_i, D_i) et (A_j, B_j, C_j, D_j) dans le système original (4.35) avec $A_i = A_j$, $B_i = B_j$ et $C_i \neq C_j$ et $D_i \neq D_j$. Alors, dans le système (4.54), ces deux modes sont décrits par $(A_i, B_i, \gamma^\top C_i, \gamma^\top D_i)$ et $(A_j, B_j, \gamma^\top C_j, \gamma^\top D_j)$. Cependant, si $\gamma \in \text{null}(C_i^\top - C_j^\top) \cap \text{null}(D_i^\top - D_j^\top)$, les modes i et j qui étaient différents dans (4.35) deviennent indiscernables dans (4.54). Heureusement, en choisissant un vecteur de pondération γ de façon aléatoire comme suggéré par la proposition 3.3, de telles situations dégénérées peuvent être évitées presque sûrement. Ce fait peut être démontré en suivant une procédure similaire à celle adoptée dans dans la preuve de la proposition 3.3.

Remarque 4.6. *La condition (4.53) suggère que la possibilité de détecter une commutation survenant entre deux sous-modèles i et j est subordonnée d'une part aux paramètres des sous-modèles concernés et d'autre part aux propriétés de l'entrée d'excitation et dans une certaine mesure, à la valeur de l'état initial du système considéré. Deux sous-modèles très distincts de part leurs paramètres peuvent avoir un comportement similaire (dans le sens où la condition (4.53) n'est pas satisfaite) pour une certaine entrée d'excitation (cf. par exemple [47] pour une étude sur des modèles monovariables à temps continu).*

La gestion de la période de transition du sous-modèle i vers un sous-modèle j est un problème délicat. En effet, un problème relatif à cette période est par exemple celui du possible changement de la base d'état dans laquelle les matrices du système sont estimées. Dans [119], l'état est calculé à chaque instant de commutation et cela, afin de ramener les matrices des sous-modèles dans des bases d'états cohérentes (cf. Sous-section 2.4.2). Mais ce procédé exigerait que l'on connaisse avec exactitude les instants de commutations ; ce qui n'est pas le cas en pratique, car ces instants sont détectés avec plus ou moins de retard surtout dans un contexte bruité. L'intérêt de notre méthode pour l'identification de systèmes à commutations est qu'elle garantit que les matrices des sous-modèles estimés seront relatives à la base canonique d'observabilité. Cependant, il nous faut distinguer la notion de *base commune* de celle de *bases cohérentes* pour les sous-modèles. Lorsque les matrices des sous-modèles sont relatives à la même base d'état, elles sont directement comparables. Par contre la notion de cohérence des bases signifie que le modèle hybride global formé à partir des sous-modèles explique bien les données entrée-sortie (cf. Section 4.3). Ainsi, les matrices des sous-modèles identifiées ici sont relatives à la même base mais ne sont pas nécessairement cohérentes.

Dans le contexte d'identification en ligne, il convient de noter que si l'on continue à adapter les paramètres après l'occurrence d'une commutation, les paramètres obtenus pour le sous-modèle i pourraient être faussés par suite d'une mise à jour avec des données mixtes (générées par deux sous-modèles différents). Ainsi, une fois qu'une commutation est détectée, il est impératif d'arrêter l'apprentissage du

sous-modèle i et d'enregistrer les estimées finales de ses paramètres. Puis, l'algorithme 3.1 est re-initialisé avec la création d'un nouveau sous-modèle. Comme il peut exister un décalage δ entre le vrai instant de commutation τ et celui $\hat{\tau}$ détecté par l'algorithme 3.1, on sauvegardera plutôt les valeurs des paramètres correspondant à l'instant $\hat{\tau} - \delta$. En pratique, on ne connaît pas δ. Cependant, sous les conditions de la proposition 4.3, ce décalage δ est théoriquement inférieur ou égal à f. Ainsi, ce décalage présumé peut être choisi tel que $\hat{\delta} \geq f$.

Soit $\mathscr{S}$ l'ensemble des sous-modèles graduellement enregistrés à mesure qu'ils sont identifiés, et désignons par s un compteur du nombre de sous-modèles. Nous notons formellement par M_{λ_t} le sous-modèle courant et définissons $\theta(t)$ comme une vectorisation des $f+1$ premiers paramètres de Markov :

$$\theta(t) = \text{vec}\left(\begin{bmatrix} D(t)^\top & \left(\Gamma_f(t)B(t)\right)^\top \end{bmatrix}^\top\right) \in \mathbb{R}^{(f+1)n_y n_u},$$

où $D(t)$ indique par exemple la matrice D en cours d'estimation. Dans la suite, nous appellerons fusion de deux sous-modèles M_{λ_t} avec M_m l'opération qui consiste à remplacer les paramètres de M_{λ_t} par une somme pondérée des paramètres de M_{λ_t} et M_m, c'est-à-dire, $M_{\lambda_t} \leftarrow \alpha M_{\lambda_t} + (1-\alpha)M_m$, avec $0 < \alpha < 1$. Soit maintenant $d\left(M_{\lambda_t}, M_j\right)$ une certaine métrique entre les sous-modèles M_{λ_t} et M_j. Une métrique entre des modèles dynamiques peut être définie de différentes façons (cf. par exemple [34]). Pour des raisons de simplicité, nous utilisons ici la distance euclidienne

$$d\left(M_{\lambda_t}, M_j\right) = \sqrt{\left(\theta(t) - \theta_j\right)^\top \left(\theta(t) - \theta_j\right)}$$

entre les vecteurs des paramètres de Markov des sous-modèles M_{λ_t} et M_j. L'algorithme 4.1 ci-dessous estime les ordres et les paramètres de chacun des sous-modèles du système commutant (4.35) en même temps qu'il classe les sous-modèles obtenus.

Algorithme 4.1 Détection de commutations et identification et classification des sous-modèles

1. Initialisation : Faire $\mathscr{S} \leftarrow \emptyset$, $s \leftarrow 1$ et créer un sous-modèle M_{λ_t}. Fixer des seuils η_o et η.
2. Mettre à jour M_{λ_t} en utilisant l'algorithme 3.1 jusqu'à ce qu'une commutation soit détectée (grâce à la proposition 4.3).
3. Sauvegarder alors M_{λ_t} : $\mathscr{S} \leftarrow \mathscr{S} \cup \{M_{\lambda_t}\}$, et créer un nouveau sous-modèle.
4. Quand les estimées des paramètres ont convergé dans le sens $\|\theta(t) - \theta(t-1)\| < \eta_o \theta(t-1)$ pour un certain seuil $\eta_o > 0$ et pour un certain t, alors classer le sous-modèle M_{λ_t} comme suit. Soit η un certain seuil, et

$$M_m = \arg\min_{M_j \in \mathscr{S}} d\left(M_{\lambda_t}, M_j\right).$$

   Si $d\left(M_{\lambda_t}, M_m\right) < \eta$, alors, fusionner M_{λ_t} et M_m,
   Si $d\left(M_{\lambda_t}, M_m\right) \geq \eta$, alors M_{λ_t} est un nouveau sous-modèle et ainsi, faire $s \leftarrow s+1$.
5. Aller à l'étape 2 et continuer cette procédure tant qu'il y a des données à traiter.

4.5 Conclusion

Nous avons étudié dans ce chapitre le problème qui consiste dans l'estimation des paramètres d'un modèle linéaire commutant à partir de ses mesures entrée-sortie. Une difficulté majeure de ce problème réside dans la complexité du modèle. En effet, puisque l'état continu du modèle est inconnu *a priori*, nous devons l'exprimer en fonction des données entrée-sortie, des paramètres et de l'état discret du système qui sont également inconnus. Cette opération s'accompagne malheureusement d'une augmentation exponentielle du nombre de sous-modèles constitutifs du système, conduisant dans le cas général, à une complexité bien pénalisante. Pour pallier cette difficulté, nous nous sommes plutôt intéressés au cas plus pratique où il existe un certain délai minimum entre deux commutations consécutives. Un des algorithmes présentés au Chapitre 3 a alors été généralisé à l'estimation des SLC.
L'application de cette méthode ne nécessite pas une connaissance des ordres et du nombre des sous-modèles, les ordres pouvant de plus différer d'un sous-modèle à un autre. Nous identifions en ligne à la fois l'état discret à travers des techniques de détection de changements, les ordres et le nombre de sous-modèles. Les ordres sont estimés grâce à une condition de rang portant sur les données. Quant au nombre de sous-modèles il est incrémentalement identifié et corrigé via des techniques de classification de l'ensemble des sous-modèles dynamiques.

Dans cette procédure, des données sont inévitablement perdues durant les transitions d'un sous-modèle à un autre. En comparaison avec les techniques existantes, l'objectif ici n'était pas tant de classer complètement toutes les données générées par chaque sous-modèle mais plutôt d'obtenir les paramètres du sous-modèle courant et adapter récursivement ses paramètres dans l'attente d'une nouvelle commutation. Un inconvénient notable des méthodes opérant hors ligne pour l'identification de systèmes à commutations est que, chaque fois que la base de données servant à l'identification ne couvre pas tous les modes, on ne peut pas identifier complètement les sous-modèles du système. Notre méthode permet de surmonter cette difficulté puisqu'elle permet de suivre en continu l'apparition d'éventuels nouveaux modes de fonctionnement. De plus, elle peut, de part sa conception, s'appliquer aussi bien à des systèmes à temps invariant qu'à des systèmes dont les paramètres varient lentement dans le temps.

Cependant, il est bien difficile de quantifier théoriquement le temps de séjour minimum (quoiqu'il apparaisse intuitivement que celui-ci doit être fonction des dimensions du système considéré) requis pour la bonne marche de cet algorithme d'identification. C'est pourquoi nous proposons dans le prochain chapitre de nous débarrasser de cette hypothèse en considérant cette fois des modèles MIMO ARX.

5

Identification de modèles MIMO SARX

Sommaire

5.1 Introduction

Nous avons vu dans le chapitre précédent que l'estimation de modèles d'état commutants nécessitait pour être faisable, que l'on fasse l'hypothèse d'un certain temps de séjour minimum du système dans les différents états discrets. Sans cette hypothèse, la résolution du problème d'identification se heurte en effet à une explosion du nombre de sous-modèles. Afin de surmonter cette contrainte qui peut être très restrictive surtout pour des systèmes d'ordre élevé, nous considérons dans ce chapitre, toujours le problème de l'identification de systèmes à commutations, mais avec des sous-modèles linéaires décrits cette fois par des modèles ARX. L'avantage des modèles ARX est qu'ils permettent justement de traiter des commutations complètement arbitraires à la condition que le système global reste stable.

Nous nous intéressons plus particulièrement à l'estimation des paramètres de modèles ARX multivariables commutants pour lesquels le nombre de sous-modèles et leurs ordres respectifs (potentiellement différents d'un sous-modèle à un autre) sont supposés être simultanément inconnus. C'est un problème difficile à cause du couplage entre les paramètres du modèle et l'état discret qui sont tous inconnus. Nous surmontons cette difficulté en éliminant algébriquement l'état discret (qui est inconnu), dans les équations du système commutant SARX. Cette procédure algébrique conduit à un ensemble de polynômes de découplage hybride [129] qui s'annulent sur l'ensemble des données entrée-sortie indépendamment de l'état discret. On peut alors procéder à l'identification des coefficients de ces polynômes en utilisant des techniques classiques de régression linéaire. Etant donné ces polynômes, les paramètres des sous-modèles peuvent être obtenus par différentiation.

En fait, notre méthode est basée sur une technique plus générale appelée GPCA (Generalized Principal Component Analysis) [129] qui permet de séparer un mélange de données distribuées sur plusieurs sous-espaces, à travers l'estimation et la différentiation de polynômes homogènes s'annulant sur l'ensemble de ces sous-espaces. Contrairement au cas de l'identification de modèles SISO SARX [131], [77], où un seul polynôme *annihilateur* est utilisé pour réaliser une transformation polynomiale de données appartenant à un mélange d'hyperplans, l'identification de modèles MIMO SARX implique un nombre potentiellement inconnu $n_h \geq 1$ de polynômes homogènes indépendants qui s'annulent sur des sous-espaces qui ne sont plus des hyperplans (c'est-à-dire des sous-espaces de co-dimension supérieure à un). Afin de construire convenablement les régresseurs auxquels va s'appliquer la transformation polynomiale, nous commençons par identifier (Section 5.3) les ordres des sous-modèles ainsi que le nombre d'états discrets à partir de conditions de rang portant sur les données entrée-sortie. Ainsi, étant donné le nombre de sous-modèles, nous calculons, sous certaines hypothèses, le nombre n_h de polynômes annihilateurs puis nous identifions les paramètres des modèles ARX à partir des dérivées des polynômes. Cette procédure constitue une solution analytique au problème d'identification des SARX. Cependant, le nombre de coefficients de polynômes à estimer croît exponentiellement avec le nombre de sorties et le nombre de sous-modèles, induisant ainsi un coût calculatoire élevé.

Pour nous affranchir de cet inconvénient, nous envisageons en section 5.4 un autre schéma d'identification dans lequel les données entrée-sortie sont d'abord projetées dans un espace de plus petite dimension puis ajustées à un seul polynôme de découplage dont on peut obtenir un partitionnement

des données de régression selon les différents sous-modèles. Les paramètres de chaque sous-modèle sont ensuite identifiés à partir de la partition obtenue.
En partant de ces simplifications, nous étudions en section 5.5, une extension de la méthode mentionnée ci-dessus à l'identification récursive. Un seul polynôme homogène est alors estimé pour l'extraction instantanée de l'état discret et les moindres carrés récursifs sont utilisés pour la mise à jour des paramètres correspondant à l'état discret identifié.

Nous terminons ce chapitre par la présentation, dans la section 5.6, d'une technique récursive qui procède alternativement à l'estimation de l'état discret et à la mise à jour des paramètres. Ce dernier schéma, à défaut d'être aussi performant en pratique que la méthode algébrique, possède l'avantage d'être bien moins coûteux.

5.2 Formulation du problème

Nous considérons un modèle MIMO SARX de la forme

$$y(t) = \sum_{i=1}^{n_{\lambda_t}} A^i_{\lambda_t} y(t-i) + \sum_{i=0}^{n_{\lambda_t}} B^i_{\lambda_t} u(t-i) + e(t), \tag{5.1}$$

où $y(t) \in \mathbb{R}^{n_y}$ est le vecteur de sortie, $u(t) \in \mathbb{R}^{n_u}$ est le vecteur d'entrée, $\lambda_t \in \{1, \dots, s\}$ est l'état discret, n_{λ_t} est l'ordre du sous-modèle indexé par λ_t, s est le nombre de sous-modèles du système SARX et $\left\{A^i_j\right\}_{j=1,\dots,s}^{i=1,\dots,n_j} \in \mathbb{R}^{n_y \times n_y}$ et $\left\{B^i_j\right\}_{j=1,\dots,s}^{i=1,\dots,n_j} \in \mathbb{R}^{n_y \times n_u}$ sont les matrices de paramètres associées. Le bruit de procédé représenté par $e(t) \in \mathbb{R}^{n_y}$ est supposé blanc gaussien. Dans la représentation (5.1), il peut exister pour certains sous-modèles j un entier $\delta_j < n_j$ tel que $B^i_j = 0$ pour $i > \delta_j$ mais nous requérons que $A^{n_j}_j \neq 0$ pour tout j.

Problème 5.1. *Etant donné des mesures entrée-sortie $\{u(t), y(t)\}_{t=1}^N$ générées par un modèle SARX de la forme (5.1), et des bornes supérieures sur les ordres du système $\bar{n} \geq \max(n_j)$ et sur le nombre de sous-modèles $\bar{s} \geq s$, estimer le nombre s de sous-modèles, les ordres $\{n_j\}_{j=1}^s$ ainsi que les paramètres $\left\{A^i_j, B^i_j\right\}_{j=1,\dots,s}^{i=1,\dots,n_j}$.*

5.3 Identification algébrique de systèmes MIMO à commutations

Pour commencer la procédure d'identification, nous définissons les matrices de paramètres

$$\begin{aligned} Q_j &= \begin{bmatrix} B^{n_j}_j & A^{n_j}_j & \cdots & B^1_j & A^1_j & B^0_j & A^0_j \end{bmatrix} \in \mathbb{R}^{n_y \times (n_j+1)(n_u+n_y)}, \\ P_j &= \begin{bmatrix} 0_{n_y \times q_j} & Q_j \end{bmatrix} \in \mathbb{R}^{n_y \times K},\ j = 1, \cdots, s, \end{aligned} \tag{5.2}$$

et le vecteur de régression

$$x_n(t) = \begin{bmatrix} u(t-n)^\top & y(t-n)^\top & \cdots & u(t-1)^\top & y(t-1)^\top & u(t)^\top & -y(t)^\top \end{bmatrix}^\top \in \mathbb{R}^K, \tag{5.3}$$

avec $n = \max_j(n_j)$, $A_j^0 = I_{n_y}$, $q_j = (n - n_j)(n_u + n_y)$ et $K = (n+1)(n_u + n_y)$. Il est important pour la suite, de bien noter que la dernière matrice bloc A_j^0 de Q_j vaut l'identité.

Pour l'instant, nous supposons que les données ne sont sujettes à aucun bruit de mesure, c'est-à-dire que $e(t) = 0$ dans l'équation (5.1). Alors, les équations définissant un système SARX de la forme (5.1) pourraient être réécrites sous la forme

$$\big(P_1 x_n(t) = 0\big) \vee \cdots \vee \big(P_s x_n(t) = 0\big), \tag{5.4}$$

où le symbole $\vee$ fait référence à l'opérateur logique *ou*. La stratégie de la méthode algébro-géométrique développée par Vidal *et al.* [131], et rapportée dans le chapitre 2 de ce manuscrit, consiste à affranchir algébriquement le système des sn_y équations (5.4) de la dépendance de l'état discret λ_t qui est inconnu. Pour ce faire, nous procédons d'une façon similaire au cas des modèles SISO SARX [131]. Nous formons à partir de (5.4), le produit des équations le long de chaque ensemble $(i_1, \cdots, i_s) \in \{1, \cdots, n_y\}^s$ de lignes à raison d'une ligne par sous-modèle (voir Figure 5.1), i_j étant le numéro de la ligne de P_j impliquée dans le produit. Nous procédons ainsi pour toutes les combinaisons de $(i_1, \cdots, i_s)$ possibles dans $\{1, \cdots, n_y\}^s$. L'avantage de cette procédure est qu'elle permet d'obtenir un ensemble de contraintes polynomiales

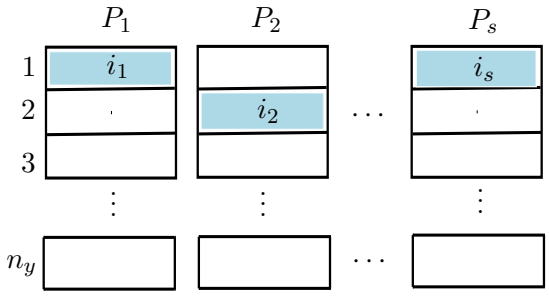


FIGURE 5.1 – Découplage du système d'équations (5.4) en polynômes homogènes de degré s.

$\prod_{j=1}^{s} \left(\theta_{i_j}^\top x_n(t)\right) = 0$, avec $\theta_{i_j}^\top = P_j(i_j, :)$, avec $(i_1, \cdots, i_s) \in \{1, \ldots, n_y\}^s$, qui sont satisfaites par toutes les données indépendamment de leur sous-modèle générateur. Par conséquent, les équations (5.4) sont équivalentes à un ensemble de $(n_y)^s$ (non nécessairement indépendants) polynômes homogènes [1] $p_{i_1,\cdots,i_s}$ en $x_n(t)$ de la forme

$$\begin{aligned} p_{i_1,\cdots,i_s}(z) = \prod_{j=1}^{s} \left(\theta_{i_j}^\top z\right) &= \sum h_{i_1,\cdots,i_s}^{n_1,\cdots,n_K} z_1^{n_1} \cdots z_K^{n_K} \\ &= h_{i_1,\cdots,i_s}^\top \nu_s(z), \end{aligned} \tag{5.5}$$

avec $(i_1, \cdots, i_s) \in \{1, \cdots, n_y\}^s$. Ici, $\nu_s : \mathbb{R}^K \to \mathbb{R}^{M_s(K)}$, avec $M_s(K) = \binom{K+s-1}{s} = \frac{(K+s-1)!}{s!(K-1)!}$, est l'application de Veronese [52] qui associe à $z \in \mathbb{R}^K$, le vecteur de tous les monômes de degré s, $z_1^{s_1} \cdots z_K^{s_K}$,

1. Un polynôme homogène est un polynôme dont tous les monômes ont le même degré total. Par exemple, le polynôme $x^2 + xy + y^2$ est homogène tandis que $x^2 + x + y$ n'est pas homogène.

$s_1 + \cdots + s_K = s$, organisés dans un ordre lexicographique décroissant. Ainsi, chaque $p_{i_1,\cdots,i_s}$ est un polynôme homogène de degré s avec comme vecteur de coefficients $h_{i_1,\cdots,i_s} \in \mathbb{R}^{M_s(K)}$ et tous les monômes de degré s en K variables assemblés en forme de vecteur dans $\nu_s(z) \in \mathbb{R}^{M_s(K)}$. En fait, chaque vecteur $h_{i_1,\cdots,i_s}$ est le produit tensoriel symétrique, noté $\text{Sym}\,(\theta_{i_1} \otimes \cdots \otimes \theta_{i_s})$, où $\otimes$ désigne le produit de Kronecker [26], des vecteurs $\theta_{i_1}, \cdots, \theta_{i_s}$, c'est-à-dire, le vecteur des coefficients du polynôme produit $p(z) = (\theta_{i_1}^\top z) \cdots (\theta_{i_s}^\top z)$. Ainsi, on peut écrire

$$p(z) = (\theta_{i_1}^\top z) \cdots (\theta_{i_s}^\top z) = \text{Sym}\,(\theta_{i_1} \otimes \cdots \otimes \theta_{i_s})^\top \nu_s(z) = h_{i_1,\cdots,i_s}^\top \nu_s(z).$$

Exemple. Pour donner une illustration de la forme des polynômes (5.5), nous prenons l'exemple d'un système commutant, composé de deux sous-modèles donnés par (cf. Eq. (5.4))

$$\big(P_1 x = 0\big) \vee \big(P_2 x = 0\big), \tag{5.6}$$

où

$$P_1 = \begin{bmatrix} a^\top \\ b^\top \end{bmatrix}, \; P_2 = \begin{bmatrix} c^\top \\ d^\top \end{bmatrix},$$

avec a et x définis par $a = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^\top$ et $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top$. Les vecteurs b, c et d sont définis d'une façon similaire à a.
Le système d'équations (5.6) est équivalent à

$$\begin{cases} (a^\top x)(c^\top x) = 0 \\ (a^\top x)(d^\top x) = 0 \\ (b^\top x)(c^\top x) = 0 \\ (b^\top x)(d^\top x) = 0. \end{cases}$$

dont le développement conduit à un ensemble de polynômes $p_{i_1,i_2}(x) = h_{i_1,i_2}^\top \nu_2(x)$, $i_1, i_2 \in \{1, 2\}$, du type (5.5) avec un vecteur de monômes

$$\nu_2(x) = \begin{bmatrix} x_1^2 & x_1x_2 & x_1x_3 & x_2^2 & x_2x_3 & x_3^2 \end{bmatrix}^\top,$$

et des vecteurs de coefficients correspondants

$$\begin{aligned}
h_{1,1} &= \begin{bmatrix} a_1c_1 & a_1c_2 + a_2c_1 & a_1c_3 + a_3c_1 & a_2c_2 & a_2c_3 + a_3c_2 & a_3c_3 \end{bmatrix}^\top, \\
h_{1,2} &= \begin{bmatrix} a_1d_1 & a_1d_2 + a_2d_1 & a_1d_3 + a_3d_1 & a_2d_2 & a_2d_3 + a_3d_2 & a_3d_3 \end{bmatrix}^\top, \\
h_{2,1} &= \begin{bmatrix} b_1c_1 & b_1c_2 + b_2c_1 & b_1c_3 + b_3c_1 & b_2c_2 & b_2c_3 + b_3c_2 & b_3c_3 \end{bmatrix}^\top, \\
h_{2,2} &= \begin{bmatrix} b_1d_1 & b_1d_2 + b_2d_1 & b_1d_3 + b_3d_1 & b_2d_2 & b_2d_3 + b_3d_2 & b_3d_3 \end{bmatrix}^\top.
\end{aligned}$$

Les h_{i_1,i_2}, $i_1, i_2 \in \{1,2\}$ ainsi définis sont en fait les produits tensoriels symétriques [52] des vecteurs de paramètres correspondants, c'est-à-dire, $h_{1,1} = \text{Sym}\,(a \otimes c)$, $h_{1,2} = \text{Sym}\,(a \otimes d)$ etc, où $\text{Sym}(\cdot)$ est l'opérateur de produit tensoriel symétrique. On peut par ailleurs noter l'existence d'une matrice $S(s,K) \in \mathbb{R}^{M_s(K)\times K^s}$ (ici $s=2$ et $K=3$) remplie avec des zéros et des uns et dépendant d'une part, du nombre s de termes impliqués dans le produit tensoriel (ici $s=2$) et d'autre part, de la longueur K du vecteur x (ici $K=3$), telle que $h_{1,1} = S(2,3)\,(a \otimes c)$, $h_{1,2} = S(2,3)\,(a \otimes d)$, $h_{2,1} = S(2,3)\,(b \otimes c)$, $h_{2,2} = S(2,3)\,(b \otimes d)$ avec

$$S(2,3) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.7}$$

Ainsi, les commutations entre les deux équations du système hybride (5.6) sont équivalentes au système unique d'équations polynômiales

$$H^\top \nu_2(x) = \begin{bmatrix} h_{1,1}^\top \\ h_{1,2}^\top \\ h_{2,1}^\top \\ h_{2,2}^\top \end{bmatrix} \nu_2(x) = 0.$$

C'est donc une équation linéaire par rapport aux coefficients $h_{i,j}$ des polynômes. La connaissance d'un ensemble fini et varié de x vérifiant cette équation, permet de calculer immédiatement les $h_{i,j}$. Une fois que ces vecteurs sont connus, on pourra (comme nous le verrons dans la suite) retourner aux paramètres a, b, c, d de l'équation (5.6) grâce à une factorisation polynomiale ou à une différentiation.

Interprétation de la transformation polynomiale. La relation (5.4) peut s'exprimer autrement comme

$$x_n(t) \in \mathscr{A} = \text{null}\,(P_1) \cup \cdots \cup \text{null}\,(P_s)\,. \tag{5.8}$$

Cela signifie que les données de régression $x_n(t)$ générées par le système hybride (5.1) sont distribuées dans l'union $\mathscr{A}$ des sous-espaces $\text{null}\,(P_j)$ encore appelé (dans la suite) arrangement [38] des sous-espaces $\text{null}\,(P_j)$. La transformation de Veronese $\nu_s : \mathbb{R}^K \to \mathbb{R}^{M_s(K)}$ projette ces données dans un espace de grande dimension $\mathbb{R}^{M_s(K)}$ dans lequel toutes les données, indépendamment de leur sous-modèle générateur, sont circonscrites dans un sous-espace unique contenant $\text{null}(\mathcal{H})$ où $\mathcal{H} = [h_{i_1,\cdots,i_s}]_{i_1,\cdots,i_s}$ est une matrice dont les colonnes sont les vecteurs de coefficients $h_{i_1,\ldots,i_s}$. Notons aussi que $h_{i_1,\ldots,i_s}$ est le produit tensoriel symétrique de l'ensemble indexé $\{\theta_{i_j}\}_{j=1}^s \in \mathbb{R}^K$ de lignes prises dans $\{P_j\}_{j=1}^s$, c'est-à-dire que $h_{i_1,\ldots,i_s} = \text{Sym}\,(\theta_{i_1} \otimes \cdots \otimes \theta_{i_s}) \in \mathbb{R}^{M_s(K)}$, où $\otimes$ dénote le produit de Kronecker. Ces vecteurs de coefficients vivent dans un sous-espace vectoriel de $\mathbb{R}^{M_s(K)}$ que nous désignerons par l'espace des polynômes homogènes de degré s qui s'annulent sur l'ensemble des données. Sur la figure 5.2, nous

illustrons la transformation appliquée à l'espace des données ainsi que celle induite dans l'espace des paramètres.

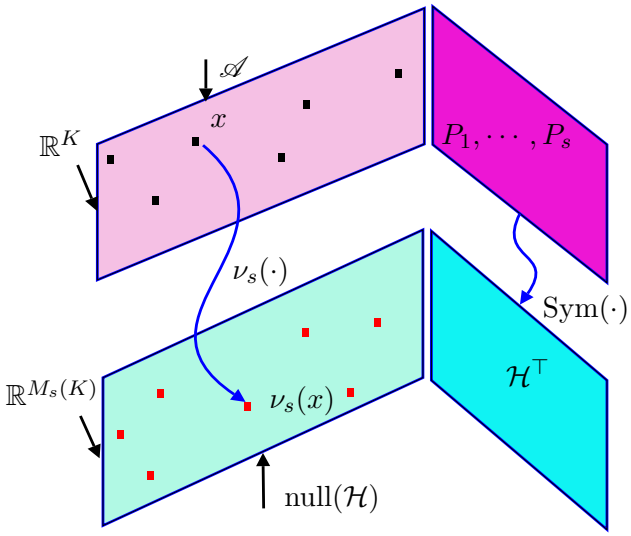


FIGURE 5.2 – Illustration de la transformation des données. Dans l'espace de régression $\mathbb{R}^K$, le sous-espace engendré par les données de régression $x_n(t)$ générées individuellement par le sous-modèle j, avec $j = 1, \ldots, s$, et celui engendré par les colonnes de la matrice $P_j^\top$, sont représentés par des plans orthogonaux. De même, dans l'espace des transformées $\mathbb{R}^{M_s(K)}$, les sous-espaces engendrés respectivement par les données $\nu_s(x_n(t))$ et les colonnes de la matrice $\mathcal{H}$, sont orthogonaux.

La méthode qui sera présentée dans les sections suivantes consiste à estimer d'abord l'arrangement $\mathscr{A}$ représenté par la matrice $\mathcal{H}$ puis à en déduire les différentes composantes null(P_j) de cet arrangement.

5.3.1 Nombre de sous-modèles connu et ordres connus et égaux

Dans cette sous-section, nous étudions le cas relativement simple où le nombre de sous-modèles s est connu, et que les ordres des sous-modèles sont connus et égaux à n. Selon (5.8), le régresseur $x_n(t)$ généré par le modèle hybride (5.1) appartient à une union de s sous-espaces $\{\text{null}(P_j)\}_{j=1}^s$. Une base de chacun de ces sous-espaces peut être estimée en utilisant l'algorithme GPCA [129] comme suit. A partir de l'ensemble $\{u(t), y(t)\}_{t=1}^N$ des données entrée-sortie disponibles, si nous construisons la matrice des régresseurs *transformés*

$$L(n,s) = \Big[\nu_s\big(x_n(n+1)\big) \quad \cdots \quad \nu_s\big(x_n(N)\big)\Big]^\top \in \mathbb{R}^{(N-n)\times M_s(K)}, \tag{5.9}$$

alors chacun des vecteurs de coefficients $h_{i_1,\cdots,i_s}$ qui définissent les polynômes (5.5) doit vérifier

$$L(n,s)h_{i_1,\cdots,i_s} = 0, \tag{5.10}$$

pour tout $(i_1, \cdots, i_s) \in \{1, \ldots, n_y\}^s$. Afin de déterminer les paramètres $h_{i_1,\cdots,i_s}$ à partir de l'équation (5.10), on a besoin de calculer l'espace noyau de la matrice de données embarquées $L(n,s)$. Et en déterminant l'espace noyau de $L(n,s)$, on obtient une base du sous-espace de $\mathbb{R}^{M_s(K)}$ engendré par les coefficients des polynômes homogènes de degré s qui s'annulent sur les données de régression. Dans ce qui suit, une telle base, de dimension n_h (inconnue *a priori*) sera notée

$$H = \begin{bmatrix} h_1 & \cdots & h_{n_h} \end{bmatrix} \in \mathbb{R}^{M_s(K) \times n_h}. \tag{5.11}$$

En fait, les éléments de cette base n'ont pas nécessairement la même structure que les vecteurs du type $h_{i_1,\cdots,i_s}$ définis ci-dessus. Cela signifie que H n'est pas nécessairement une sous-matrice de $\mathcal{H}$ mais probablement une certaine combinaison linéaire des colonnes de celle-ci.

Dans la mesure où les données sont parfaites et suffisamment riches pour que la dimension de l'espace noyau de $L(n,s)$ soit exactement égale à n_h, la matrice H peut être calculée à partir de (5.10) comme une base quelconque de $\text{null}(L(n,s))$ en utilisant par exemple une Décomposition en Valeurs Singulières (DVS) de $L(n,s)$. Plus précisément, si

$$L(n,s) = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix},$$

est une DVS de $L(n,s)$, avec Σ_1 une matrice diagonale de dimension $M_s(K) - n_h$, et contenant les valeurs singulières non nulles de $L(n,s)$ alors $H = V_2$ est une base de $\text{null}(L(n,s))$. Disposant de la base H de coefficients des polynômes homogènes *annihilateurs* de degré s, il nous faut en déduire les matrices de paramètres P_j définies en (5.2). Pour ce faire, considérons pour tout $j \in \{1, \ldots, s\}$, un point $z_j \in \mathbb{R}^K$ vérifiant

$$z_j \in \text{null}(P_j) \quad \text{et} \quad z_j \notin \text{null}(P_i), \forall i \neq j, \tag{5.12}$$

c'est-à-dire, z_j appartient au sous-espace $\text{null}(P_j)$ à l'exclusion de tous les autres sous-espaces $\text{null}(P_i), i \neq j$. Il suffit alors, comme cela est démontré dans [129] et [124], d'évaluer la dérivée (par rapport à z) du vecteur $\mathcal{Q}(z) = \begin{bmatrix} p_1(z) & \cdots & p_{n_h}(z) \end{bmatrix} = \nu_s(z)^\top H \in \mathbb{R}^{1 \times n_h}$ des polynômes définis par H pour obtenir une base du sous-espace $\text{span}(P_j^\top) = \text{null}(P_j)^\perp$ qui est orthogonal à $\text{null}(P_j)$ (cf. Algorithme 5.1). La matrice de paramètres $P_j^\top$ du sous-modèle indexé par j (il s'agit en fait de sa transposée) peut ensuite être calculée comme la base particulière de $\text{span}(P_j^\top)$ qui possède une matrice identité à la fin, conformément à la définition (5.2) des matrices P_j. Quant à la détermination concrète de vecteurs $\{z_j\}_{j=1}^s$ obéissant à la propriété (5.12), elle peut se faire simplement en procédant à une sélection parmi les vecteurs de données $x_n(t)$. En effet, chaque fois que $\nabla\mathcal{Q}(x_n(t)) \neq 0$, où $\nabla\mathcal{Q}(x_n(t))$ est le gradient de $\mathcal{Q}(z)$ évalué en $x_n(t)$, $\text{span}(P_{\lambda_t}^\top)$ est engendré par les colonnes de $\nabla\mathcal{Q}(x_n(t))$. On peut donc en déduire la matrice P_{λ_t} pour tout instant.

Mais comme on n'a pas besoin d'estimer les matrices de paramètres pour chaque instant, on peut, comme alternative, choisir s régresseurs z_j vérifiant (5.12) comme à la section 5.4.1 et ainsi obtenir s matrices de paramètres $\{P_j\}_{j=1}^s$ à partir des dérivées des polynômes *annihilateurs*, évaluées en $\{z_j\}_{j=1}^s$.

L'algorithme 5.1 ci-dessous résulte de l'algorithme GPCA [129] pour le calcul, dans un contexte déterministe, des matrices de paramètres du système $\{P_j\}_{j=1}^{s}$ à partir des données entrée-sortie.

Algorithme 5.1 (Identification de systèmes MIMO SARX basée sur l'algorithme GPCA)

Etape 1 : Trouver, par DVS, une base $H \in \mathbb{R}^{M_s(K)\times n_h}$ de l'espace noyau de $L(n,s)$ et noter $\mathcal{Q}(z) = \begin{bmatrix} p_1(z) & \cdots & p_{n_h}(z) \end{bmatrix} = \nu_s(z)^\top H$ la base correspondante de polynômes *annihilateurs* de degré s.

Etape 2 : Differentier $\mathcal{Q}(z)$ par rapport à z :

$$\nabla\mathcal{Q}(z) = \begin{bmatrix} \frac{\partial p_1(z)}{\partial z} & \cdots & \frac{\partial p_{n_h}(z)}{\partial z} \end{bmatrix} = \left(\frac{\partial \nu_s(z)}{\partial z}\right)^\top H \in \mathbb{R}^{K\times n_h}. \tag{5.13}$$

Etape 3 : Obtenir par DVS une base $T_j \in \mathbb{R}^{K\times n_y}$ de $\text{span}(P_j^\top)$ comme l'espace image de la matrice $\nabla\mathcal{Q}(z_j)$, $j = 1, \cdots, s$, avec $z_j \in \text{null}(P_j)$ mais $z_j \notin \text{null}(P_i)$, pour tout $i \neq j$.

Etape 4 : Soit $T_j^\top = \begin{bmatrix} T_j^1 & T_j^2 \end{bmatrix} \in \mathbb{R}^{n_y\times K}$ une partition de $T_j^\top$ telle que $T_j^2 \in \mathbb{R}^{n_y\times n_y}$. Alors T_j^2 est nécessairement inversible et on a $P_j = \left(T_j^2\right)^{-1} T_j^\top \in \mathbb{R}^{n_y\times K}$, $j = 1, \cdots, s$.

Mais en pratique, les données entrée-sortie pourraient être affectées par du bruit. Dans ce cas, même avec l'hypothèse que les ordres et le nombre de sous-modèles sont connus *a priori*, la matrice $L(n,s)$ sera très vraisemblablement de rang plein et ainsi, on ne pourrait plus déterminer la base H des coefficients de polynômes *annihilateurs*. Néanmoins, il est possible d'exploiter des techniques de sélection de modèles comme le critère d'information de Akaike [2] pour estimer n_h à partir des données bruitées. L'efficacité de ce type de méthodes varie malheureusement beaucoup en fonction du niveau de bruit. Il serait donc plus intéressant de connaître d'avance la dimension n_h de cette base. De cette manière, H pourrait être approximée par les vecteurs singuliers (à droite) de $L(n,s)$ qui correspondent à ses n_h plus faibles valeurs singulières. Mais comme les matrices P_j ne sont pas connues, il n'est pas facile de trouver n_h dans un contexte général. Cependant, en faisant certaines hypothèses sur l'intersection entre les espaces noyau des matrices P_j, il est possible, comme présenté dans les propositions 5.1 et 5.2, de dériver une formule analytique pour n_h.

Avant de continuer, il est nécessaire d'introduire quelques notations. Nous notons $\text{Sym}\left(P_1^\top \otimes \cdots \otimes P_s^\top\right)$, le produit tensoriel symétrique des matrices $P_i^\top \in \mathbb{R}^{K\times n_y}$, i=1,...,s, c'est-à-dire la matrice dont les colonnes sont les vecteurs du type $\text{Sym}\left(\theta_{i_1} \otimes \cdots \otimes \theta_{i_s}\right)$, où θ_{i_j} est une colonne quelconque de $P_j^\top$. Dans le cadre de notre étude, l'ordre dans lequel sont ordonnées les colonnes de cette matrice n'a pas d'importance. Alors, de manière plus générale que l'illustration donnée par (5.7), il existe une matrice $S(s,K)$ remplie de uns et de zeros telle que

$$\mathcal{H} = \text{Sym}\left(P_1^\top \otimes \cdots \otimes P_s^\top\right) = S(s,K)\left(P_1^\top \otimes \cdots \otimes P_s^\top\right).$$

Dans un cas général, le nombre n_h de polynômes homogènes indépendants qui s'annulent sur l'arrangement $\mathscr{A}$ vérifie $n_h \geq \text{rang}(\mathcal{H})$; il n' y a pas toujours égalité. Ainsi, nous nous intéressons maintenant à

la question de savoir sous quelles conditions suffisantes $n_h = \text{rang}(\mathcal{H})$. Cette investigation nous amènera au résultat intermédiaire formulé par la proposition 5.1 puis à la proposition 5.2.

Proposition 5.1. *Considérons des matrices $P_i \in \mathbb{R}^{n_y \times K}$, $i = 1, \ldots, s$ et soit, comme défini plus haut, le produit tensoriel symétrique $\mathcal{H} = \text{Sym}\left(P_1^\top \otimes \cdots \otimes P_s^\top\right)$. Supposons que*

$$\sum_{i=1}^{s} \text{rang}(P_i^\top) < K + s - 1. \tag{5.14}$$

et que pour tout $\{i_1, \cdots, i_q\} \subset \{1, \cdots, s\}$,

$$\text{rang}\left(\left[P_{i_1}^\top, \cdots, P_{i_q}^\top\right]\right) = \min\left(K, \sum_{j=1}^{q} \text{rang}(P_{i_j})\right). \tag{5.15}$$

Alors le nombre n_h de polynômes homogènes linéairement indépendants qui s'annulent sur l'arrangement de sous-espaces $\mathscr{A} = \text{null}(P_1) \cup \cdots \cup \text{null}(P_s)$ est égal à $\text{rang}\,(\mathcal{H})$.

Preuve. Une preuve de cette proposition est incluse dans l'annexe A.5. ■

L'hypothèse (5.15) correspond à une propriété importante de l'arrangement $\mathscr{A}$ appelée transversalité [78]. Cette propriété traduit le fait suivant. Les éléments de tout sous-ensemble de sous-espaces $\text{null}(P_{i_j})^\perp = \text{span}(P_{i_j}^\top)$, $j = 1, \ldots, q$, de l'arrangement $\mathscr{A}$, n'ont pas d'intersection [1] chaque fois que $\text{rang}\left(\left[P_{i_1}^\top, \cdots, P_{i_q}^\top\right]\right) < K$. Maintenant, nous pouvons formuler le résultat qui nous permettra de déterminer le nombre n_h de polynômes linéairement indépendants qui s'annulent sur l'ensemble algébrique $\mathscr{A}$, connaissant le nombre s d'états discrets.

Proposition 5.2. *Soit $\mathcal{H}$ le produit tensoriel symétrique d'un ensemble de matrices $P_1^\top, \ldots, P_s^\top$ dans $\mathbb{R}^{K \times n_y}$. Plus précisement, $\mathcal{H}$ est la matrice dont les colonnes sont constituées de tous les vecteurs de $\mathbb{R}^{M_s(K)}$ de la forme $\text{Sym}\,(b_{i_1} \otimes \cdots \otimes b_{i_s})$, où $b_{i_1}, \ldots, b_{i_s}$ sont respectivement des colonnes de $P_1^\top, \cdots, P_s^\top$. Si pour tout $\{i_1, \cdots, i_q\} \subset \{1, \cdots, s\}$, $\sum_{j=1}^{s} \text{rang}(P_{i_j}^\top) - s < K$ et la condition (5.15) est satisfaite, alors $\text{rang}(\mathcal{H}) = \prod_{j=1}^{s} \text{rang}(P_j)$.*

Preuve. Une preuve de cette proposition est incluse dans l'annexe A.6. ■

Des résultats qui précèdent (Propositions 5.1 et 5.2), nous savons que, sous l'hypothèse de transversalité de l'arrangement $\cup_{j=1}^{s} \text{null}(P_j)$, où les P_j sont définis par (5.2), si $\sum_{i=1}^{s} \text{rang}(P_i^\top) < K + s - 1$, alors le nombre n_h de polynômes homogènes indépendants qui s'annulent sur $\mathscr{A} = \cup_{j=1}^{s} \text{null}(P_j)$ est égal à $\text{rang}(\mathcal{H})$. Concrètement, si l'arrangement des sous-espaces $\mathscr{A}$ satisfait à la propriété de transversalité, et si de plus la dimension K de l'espace de régression vérifie $K = (n+1)(n_u + n_y) > s(n_y - 1) + 1$, alors

1. Il est sous-entendu, intersection différente de $\{0\}$.

les propositions 5.1 et 5.2 permettent de trouver n_h par la formule $n_h = \prod_{j=1}^{s} \text{rang}(P_j) = (n_y)^s$ puisque $\text{rang}(P_j) = n_y$ pour tout j. Bien que cette formule soit moins générale que celle donnée par exemple dans [38], elle est bien plus facile à évaluer. De plus, son application ne nécessite pas la connaissance de la dimension K de l'espace de régression qui peut être indisponible puisque cette dernière dimension est définie par l'ordre maximum $n = \max(n_j)$. Dans le reste de cette section, nous supposerons, sauf mention contraire, que les conditions de la proposition 5.1 sont remplies.

En résumé, étant donné l'ordre n (commun à tous les sous-modèles) et le nombre s de sous-modèles, les matrices de paramètres P_j définies en (5.2) peuvent être directement estimées via l'algorithme 5.1. Dans le cas où les données sont sujettes à du bruit, la base H des coefficients des polynômes homogènes qui s'annulent sur l'arrangement $\mathscr{A} = \cup_{j=1}^{s} \text{null}(P_j)$ ne peut plus être estimée directement puisque n_h ne peut plus être facilement déterminé à partir des données ($L(n,s)$ étant très vraisemblablement de rang plein). Pour surmonter ce problème, nous supposons que l'arrangement $\mathscr{A}$ est transversal et que $\sum_{i=1}^{s} \text{rang}(P_i^\top) < K + s - 1$. Nous montrons alors que $n_h = (n_y)^s$. Ainsi, l'algorithme 5.1 peut toujours être appliqué dans le cas de données bruitées mais à la différence que la base H définie par (5.11) est approximée par les vecteurs singuliers (de droite) de $L(n,s)$ qui sont associés à ses $n_h = (n_y)^s$ plus faibles valeurs singulières.

5.3.2 Nombre de sous-modèles inconnu et ordres inconnus et potentiellement différents

Considérons maintenant le cas plus délicat où ni les ordres, ni le nombre de sous-modèles ne sont connus et où les ordres peuvent différer d'un sous-modèle à un autre. Il en résulte que n_h est aussi inconnu. Cela signifie que nous devons déterminer tous les paramètres du modèle SARX (5.1) directement à partir des données. Afin d'estimer ces paramètres, nous commençons par identifier les ordres et le nombre de sous-modèles. Une fois que cette tâche est accomplie, l'algorithme 5.1 pourra être appliqué à une certaine sous-matrice de $L(n,s)$ que nous définirons dans la suite.
Pour des raisons de clarté, nous étudierons d'abord (§5.3.2.1) le cas où les données ne sont pas affectées par du bruit puis (§5.3.2.2) le cas où les données sont potentiellement bruitées.

5.3.2.1 Données non bruitées

Afin de pouvoir présenter nos développements, nous avons besoin d'introduire quelques notations. Pour r et l, des entiers positifs, nous utiliserons les mêmes définitions pour $x_r(t)$ et $L(r,l)$ que précédemment (cf. Equations (5.3) et (5.9)). Sans perte de généralité, nous désignerons par $n = n_1 \geq n_2 \geq \cdots \geq n_s$ les ordres des différents sous-modèles qui constituent le système SARX et soit $\rho = \begin{bmatrix} n_1 & \cdots & n_s \end{bmatrix} \in \mathbb{N}^s$, un vecteur consistant en une énumération des ordres dans un ordre décroissant. Il résulte de (5.2) et (5.3) que les équations définissant le modèle SARX (5.1) pourraient être réécrites (avec une paramétrisation minimale) comme

$$(Q_1 x_{n_1}(t) = 0) \vee \cdots \vee (Q_s x_{n_s}(t) = 0)\,, \tag{5.16}$$

où $x_{n_j}(t) \in \mathbb{R}^{K_j}$, $K_j = (n_j+1)(n_u+n_y)$ et $Q_j \in \mathbb{R}^{n_y \times K_j}$ pour $j = 1, \ldots, s$ (cf. Eq. (5.2)). Comme à la section 5.3, nous pouvons éliminer l'opérateur $\vee$ dans (5.16) en prenant le produit de s équations à raison d'une équation par sous-modèle. Cela conduit à un système d'équations polynomiales sur les données entrée-sortie de la forme [1]

$$\left(\theta_1^\top x_{n_1}(t)\right) \cdots \left(\theta_s^\top x_{n_s}(t)\right) = h_\rho^\top \eta_\rho\big(x_n(t)\big) = 0, \tag{5.17}$$

où $\theta_j^\top \in \mathbb{R}^{1\times K_j}$ est une ligne quelconque de Q_j, pour $j = 1, \ldots, s$, et $\eta_\rho\big(x_n(t)\big)$ est un vecteur obtenu à partir de $\nu_s\left(x_n(t)\right)$ en supprimant certains monômes. En effet $\eta_\rho\big(x_n(t)\big)$ ne contient pas nécessairement tous les monômes de degré s parce que $n_j \leq n$ pour tout $j = 1, \ldots, s$.

Pour pouvoir définir l'ensemble des monômes à éliminer, posons $z = x_n(t)$ et considérons un monôme quelconque $z_1^{\alpha_1} \cdots z_K^{\alpha_K}$, $\alpha_1 + \cdots + \alpha_K = s$. De la définition de $x_n(t)$ en (5.3), on peut constater que l'élément $z_j^{\alpha_j}$ est contenu dans un monôme de $\eta_\rho\big(x_n(t)\big)$ si le nombre de régresseurs $x_{n_i}(t)$ de longueur $K_i \geq K_1 - j + 1$ (c'est-à-dire, le nombre de régresseurs où il y a z_j) est supérieur ou égal à α_j. Ainsi, pour que le monôme $z_1^{\alpha_1} \cdots z_K^{\alpha_K}$ soit inclus dans $\eta_\rho\big(x_n(t)\big)$, il faudrait que $k_j \geq \alpha_j$ pour tout $j = 1, \ldots, K$, où $k_j = \text{card}\left(\{i : K_i \geq K_1 - j + 1\}\right)$. En exploitant cette analyse, nous pouvons montrer que l'ensemble des monômes à éliminer de $\nu_s(x_n(t))$ pour obtenir $\eta_\rho\big(x_n(t)\big)$, peut être indexé par l'ensemble $\mathscr{I}_\rho$ d'exposants [2] $(\alpha_1, \cdots, \alpha_K)$ qui satisfont à $\alpha_1 + \cdots + \alpha_j > k_j$ pour $j \leq K_1 - K_s$.

Prenons un exemple. Considérons le cas d'un système monovariable de deux sous-modèles d'ordres respectifs 1 et 0. On a

$$x_1(t) = \begin{bmatrix} u(t-1) & y(t-1) & u(t) & -y(t) \end{bmatrix}^\top \text{ et } x_0(t) = \begin{bmatrix} u(t) & -y(t) \end{bmatrix}^\top$$

de longueurs respectives $K_1 = 4$ et $K_2 = 2$. En notant $a = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}^\top$, $b = \begin{bmatrix} b_3 & b_4 \end{bmatrix}^\top$, $z = x_1(t)$ et $\rho = \begin{bmatrix} 1 & 0 \end{bmatrix}$, le vecteur $\eta_\rho(z)$ contient seulement les monômes qui sont présents dans le produit

$$\begin{aligned}(a^\top x_1(t)) \cdot (b^\top x_0(t)) &= (a_1 z_1 + a_2 z_2 + a_3 z_3 + a_4 z_4)(b_3 z_3 + b_4 z_4) \\ &= a_1 b_3 z_1 z_3 + a_1 b_4 z_1 z_4 + a_2 b_3 z_2 z_3 + a_2 b_4 z_2 z_4 \\ &\quad + a_3 b_3 z_3^2 + (a_3 b_4 + a_4 b_3) z_3 z_4 + a_4 b_4 z_4^2,\end{aligned}$$

c'est-à-dire que

$$\eta_\rho(z) = \begin{bmatrix} z_1 z_3 & z_1 z_4 & z_2 z_3 & z_2 z_4 & z_3^2 & z_3 z_4 & z_4^2 \end{bmatrix}^\top$$

alors que

$$\nu_2(z) = \begin{bmatrix} \boldsymbol{z_1^2} & \boldsymbol{z_1 z_2} & z_1 z_3 & z_1 z_4 & \boldsymbol{z_2^2} & z_2 z_3 & z_2 z_4 & z_3^2 & z_3 z_4 & z_4^2 \end{bmatrix}^\top.$$

1. L'équation (5.17) donne juste une idée de la forme d'un polynôme *annihilateur* quelconque dans le cas où les ordres des sous-modèles sont différents. Comme à la section précédente, nous considérons toutes les combinaisons $(i_1, \cdots, i_s)$ possibles de lignes de $Q_1, \cdots, Q_s$. 2. En considérant l'ordre lexicographique, nous désignerons aussi par $\mathscr{I}_\rho$ l'ensemble des indices (d'entrées de $\nu_s(x_n(t))$) correspondant à l'ensemble des exposants ainsi définis.

On peut donc voir que $\eta_\rho(z)$ peut être obtenu à partir de $\nu_s(z)$ en supprimant les composantes indexées par $\mathscr{I}_\rho = \begin{bmatrix}1 & 2 & 5\end{bmatrix}$. Cet ensemble $\mathscr{I}_\rho$ peut être déterminé comme décrit ci-dessus.

Avec les notations que nous venons d'introduire, nous définissons une nouvelle matrice de données *embarquées* dans $\mathbb{R}^{(N-n)\times\left(M_s(K_1)-|\mathscr{I}_\rho|\right)}$, par

$$V_\rho = \Big[\eta_\rho\big(x_n(n+1)\big), \cdots, \eta_\rho\big(x_n(N)\big)\Big]^\top \tag{5.18}$$

qui désigne simplement la matrice $L(n,s)$ avec $|\mathscr{I}_\rho|$ colonnes en moins, avec $n = \rho(1)$. De la même manière que précédemment, l'espace noyau de V_ρ contient les vecteurs de coefficients des polynômes *annihilateurs*. Cependant, nous ne pouvons pas calculer directement ces coefficients, puisque nous ne connaissons ni le vecteur d'ordres ρ du système, ni le nombre s de sous-modèles. Dans la suite, nous montrons que ρ et s peuvent être extraits des données sous l'hypothèse que les données sont suffisamment riches. Plus précisément,

Définition 5.1. *Nous dirons que les données $\{u(t), y(t)\}_{t=1}^N$ sont suffisamment excitantes pour le système SARX (5.1) si l'espace noyau de la matrice V_ρ définie en (5.18) est de dimension exactement égale à n_h, c'est-à-dire ,*

$$\operatorname{rang}(V_\rho) = M_s(K_1) - n_h - |\mathscr{I}_\rho|\,. \tag{5.19}$$

Notons que la définition 5.1 suppose implicitement que tous les états discrets ont été suffisamment visités par le système. Si nous désignons par $X^j_{n_j} = \begin{bmatrix}x_{n_j}(t^j_1) & \cdots & x_{n_j}(t^j_{N_j})\end{bmatrix}$, la matrice de données en relation avec l'état discret j, où t^j_k, $k = 1, \ldots, N_j$, sont les instants t tels que $\lambda_t = j$, alors une condition nécessaire mais pas suffisante à la réalisation de la condition (5.19) est que $\operatorname{rang}(X^j_{n_j}) = K_j - \operatorname{rang}(Q_j) = K_j - n_y$ pour tout j. Autrement dit, les colonnes de $X^j_{n_j}$ doivent engendrer complètement le sous-espace noyau $\operatorname{null}(Q_j)$. Pour voir le fondement de ce résultat, on peut remarquer le fait suivant. En notant H_ρ, la matrice H définie précédemment mais dans laquelle on a supprimé les lignes indexées par $\mathscr{I}_\rho$ (qui sont en fait nulles), il est nécessaire, pour réaliser $V_\rho H_\rho = 0$ avec $\operatorname{rang}(H) = \operatorname{rang}(H_\rho) = n_h$, que $Q_1 X^1_{n_1} = 0, \ldots, Q_s X^s_{n_s} = 0$. On voit donc que si pour un certain j, $\operatorname{rang}(X^j_{n_j}) < K - n_y$, ou de manière équivalente, si $\dim\big(\operatorname{null}({X^j_{n_j}}^\top)\big) > n_y$, alors il est probable qu'on ait plus de n_h vecteurs linéairement indépendants dans $\operatorname{null}(V_\rho)$, ce qui n'est pas conforme à la condition (5.19).

Afin d'estimer le nombre s de sous-modèles et le vecteur d'ordres ρ, on pourrait, sous l'hypothèse que le système SARX est suffisamment excité par les données de mesure collectées au sens de la définition 5.1, envisager de tester incrémentalement différentes valeurs du couple $(\rho, s) \in \mathbb{N}^s \times \mathbb{N}$, jusqu'à ce que la condition (5.19) soit réalisée. Puisque des majorants $\bar{s}$ du nombre d'états discrets et $\bar{n}$ de l'ensemble des ordres sont supposés connus, le champ de recherche pourrait être limité à $[0,\ \bar{n}]\times[1,\ \bar{s}]$. Malheureusement, dans une telle procédure, il n'est pas garanti que (5.19) soit réalisé par un seul couple (ρ, s). Quand bien même ce couple de valeurs existerait, on ne sait pas *a priori* dans quelle direction il faut diriger la prospection de valeurs. De plus, un problème crucial est que n_h est aussi inconnu ; ce qui rend impossible l'exécution du test d'arrêt (5.19).

Le résultat suivant permet d'éviter ces problèmes.

Theorème 5.1. *Soit $\bar{s} \geq s$ un majorant du nombre de sous-modèles et soit r un entier. Supposons que les données générées par le modèle (5.1) soient suffisamment excitantes dans le sens de la définition 5.1. Supposons de plus que $N_j \gg M_{\bar{s}}(K_1)$ pour tout $j = 1, \ldots, s$. Alors* $\dim\big(\text{null}(L(r,\bar{s}))\big) = 0$ *si et seulement si* $r < \max(n_j)$.

Preuve. Supposons $r < n_1 = \max(n_j)$. Soit q le nombre de sous-modèles dont les ordres sont inférieurs ou égaux à r. Pour tout $j = 1, \ldots, s$, nous définissons $X_r^j = \left[x_r(t_1^j), \cdots, x_r(t_{N_j}^j)\right] \in \mathbb{R}^{f \times N_j}$ comme ci-dessus, avec $t_k^j \in \{t : \lambda_t = j\}$. Soit maintenant $X^o = \left[x_r(t_1^o), \cdots, x_r(t_{N_o}^o)\right] \in \mathbb{R}^{f \times N_o}$, avec $f = (r+1)(n_u+n_y)$, une matrice dont les colonnes sont les régresseurs formés à partir des données générées par les $(s-q)$ sous-modèles d'ordres $n_j > r$. Puisque les données sont suffisamment excitantes, X^o doit être de rang plein ligne. En se servant du Lemme 5 dans [126], on obtient que $\text{rang}(\nu_{\bar{s}}(X^o)) = \min(N_o, M_{\bar{s}}(f)) = M_{\bar{s}}(f)$, avec $\nu_{\bar{s}}(X^o) = \left[\nu_{\bar{s}}(x_r(t_1^o)), \cdots, \nu_{\bar{s}}\left(x_r(t_{N_o}^o)\right)\right]$. Par conséquent, la matrice $L(r,\bar{s})$ est de rang plein, parce qu'elle est égale à $\left[\nu_{\bar{s}}(X^o), \nu_{\bar{s}}(X_r^{s-q+1}), \cdots, \nu_{\bar{s}}(X_r^s)\right]^\top$ à une permutation de lignes près.

Réciproquement, supposons que $r \geq n_1 = \max(n_j)$. Alors, la déficience en rang ligne de toutes les matrices de données X_r^j vaut au moins un. Cela signifie que pour tout $j = 1, \ldots, s$, il existe un vecteur non nul $b_j \in \mathbb{R}^f$ vérifiant $b_j^\top X_r^j = 0$. On peut alors vérifier que $\text{Sym}(b_1 \otimes \cdots \otimes b_s \otimes a_{s+1} \otimes \cdots \otimes a_{\bar{s}}) \in \text{null}(L(r,\bar{s}))$ pour tous vecteurs $a_i \in \mathbb{R}^f$. Ainsi, $\dim(\text{null}(L(r,\bar{s}))) \geq 1$. ■

Remarque 5.1. *Notons au passage que le théorème 2.1 présenté dans [131] pour un arrangement d'hyperplans et rappelé au Chapitre 2 (Section 2.4.1) de ce manuscrit, n'est pas applicable ici pour l'identification du nombre d'états discrets. En effet, selon ce résultat, il n'existe aucun polynôme homogène de degré strictement inférieur à s qui s'annule sur un ensemble de données suffisamment riches et distribuées sur une union d'hyperplans. Dans le cas considéré ici, les sous-espaces* $\text{null}(P_j)$, $j = 1, \ldots, s$ *ne sont plus des hyperplans et donc les conditions de ce théorème ne sont plus satisfaites. Un contre-exemple simple (cf. Figure 5.3) est l'union de trois lignes de $\mathbb{R}^3$ passant toutes par l'origine pour laquelle il existe un polynôme annihilateur de degré deux.*

Soient $\bar{s} \geq s$ et $\bar{n} \geq \max(n_j)$, des majorants respectivement du nombre de sous-modèles et des ordres. Grâce au théorème 5.1, nous pouvons estimer aussi bien le nombre de sous-modèles s que les ordres $\{n_j\}_{j=1}^s$ à partir du rang de la matrice de données *embarquées* $L(r,\bar{s})$. L'idée de base est que, chaque fois que r est strictement inférieur à un au moins des ordres, il n'existe aucun polynôme de degré $\bar{s} \geq s$ qui s'annule sur l'ensemble des données de régression, pourvu que $N - \bar{n} \gg s$ et que les données soient suffisamment excitantes. Ainsi, comme illustré par l'algorithme 5.2, nous pouvons obtenir le premier ordre n_1 en faisant $\bar{\rho} = \begin{bmatrix} r & \cdots & r \end{bmatrix} \in \mathbb{N}^{\bar{s}}$, de sorte que $V_{\bar{\rho}} = L(r,\bar{s})$, et ensuite commencer à décroître r en allant de $r = \bar{n}$ vers $r = 0$ jusqu'à ce que $\text{null}(V_{\bar{\rho}}) = \{0\}$ pour un certain r^*. Alors, on a $n_1 = r^* + 1$. Etant donné n_1, nous faisons l'affectation $\bar{\rho} = \begin{bmatrix} n_1 & r & \cdots & r \end{bmatrix} \in \mathbb{N}^{\bar{s}}$ et répétons la procédure en partant cette fois de $r = n_1$ et ainsi de suite, jusqu'à ce que tous les ordres de tous les s sous-modèles soient identifiés. Une fois que tous les ordres des s sous-modèles ont été correctement estimés, r peut même, sans inconvénient, aller à zéro pour les $\bar{s} - s$ autres sous-modèles présumés. Ainsi,

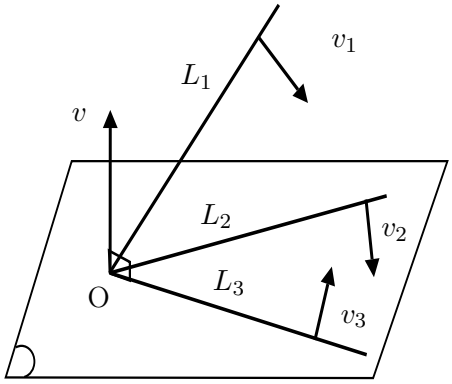


FIGURE 5.3 – Exemple d'un arrangement de trois lignes dans $\mathbb{R}^3$. L_1, L_2 et L_3 sont des lignes non coplanaires dans $\mathbb{R}^3$ et les vecteurs v_1, v_2, v_3, v sont tels que $v_1 \perp L_1$, $v_2 \perp L_2$, $v_3 \perp L_3$, $v \perp \text{span}(L_2, L_3)$, où le symbole $\perp$ signifie 'orthogonal à'. Considérons l'arrangement $\mathscr{A} = L_1 \cup L_2 \cup L_3$ des trois lignes. Alors, pour tout $x \in \mathscr{A}$, on a $(v_1^\top x)(v_2^\top x)(v_3^\top x) = 0$ mais on a également $(v^\top x)(v_3^\top x) = 0$. Ainsi, il existe un polynôme de degré $2 < s = 3$ qui s'annule sur $\mathscr{A}$. Mais si L_1, L_2 et L_3 étaient tous des hyperplans il n'existerait aucun polynôme homogène de degré strictement inférieur à s qui s'annulerait sur $\mathscr{A}$.

si l'on suppose que $n_j > 0$ pour $j = 1, \ldots, s$, alors le nombre de sous-modèles peut être obtenu comme le nombre d'ordres n_j strictement supérieurs à zéro.

Un avantage de l'algorithme 5.2 est qu'il ne nécessite pas une connaissance *a priori* de la dimension n_h de l'espace des polynômes homogènes annihilateurs. Quand tous les ordres sont correctement identifiés, la condition de suffisance d'excitation de la définition 5.1 garantit que la dimension de l'espace noyau de V_ρ sera exactement égale à n_h. Après avoir trouvé n_h comme dans la sous-section 5.3.1, il ne reste plus qu'à faire usage de l'algorithme 5.1 pour déterminer la base H_ρ de null(V_ρ). Ensuite cette base peut être complétée par des entrées nulles pour former une matrice $H \in \mathbb{R}^{M_s(K_1) \times n_h}$ telle que les lignes indexées par $\mathscr{I}_\rho$ soient nulles. Les autres étapes de l'algorithme 5.1 sont alors exécutées sans changement.

5.3.2.2 Données bruitées

L'algorithme proposé ci-dessus opérera correctement en absence de bruit. Cependant, avec des données bruitées, les multiples tests de rang requis dans l'algorithme 5.2 peuvent constituer une faiblesse de ce dernier, parce que les matrices impliquées dans ces tests seront vraisemblablement de rang plein (effet du bruit). Dans cette sous-section, nous discutons quelques améliorations possibles de la robustesse de l'algorithme en présence de bruit.

Rappelons que le but des tests de rang est de vérifier si la dimension de l'espace noyau de $V_{\bar{\rho}}$ est nulle pour un vecteur d'ordres donné $\bar{\rho}$. Ainsi, on n'a pas besoin de connaître avec exactitude le rang de $V_{\bar{\rho}}$ mais d'avoir une mesure de combien il est vraisemblable qu'il existe un vecteur non nul $h_{\bar{\rho}}$ satisfaisant $V_{\bar{\rho}} h_{\bar{\rho}} = 0$.

Une approche possible à la résolution de ce problème est d'inspecter la plus petite valeur singulière de $V_{\bar{\rho}}$ pour différents vecteurs $\bar{\rho}$. Par exemple, pour trouver n_1, soit $\bar{\rho}_{1,l} = \begin{bmatrix} l & \cdots & l \end{bmatrix} \in \mathbb{N}^{1 \times \bar{s}}$, $l = 0, \ldots, \bar{n}$,

Algorithme 5.2 (Identification des ordres et du nombre de sous-modèles)

Faire les affectations $j_o \leftarrow 1$, $n_j \leftarrow \bar{n}$, $j = 1, \ldots, \bar{s}$,
$K \leftarrow (\bar{n}+1)(n_u+n_y)$, $V \leftarrow L(\bar{n}, \bar{s})$,

1. Déterminer l'ordre maximum n_1 en utilisant le théorème 5.1
   - **Tant que** $\text{rang}(V) < M_{\bar{s}}(K)$, faire
     - $n_j \leftarrow n_1 - 1$ pour $j = 1, \ldots, \bar{s}$
     - $K \leftarrow (n_1+1)(n_u+n_y)$
     - $V \leftarrow M_{\bar{s}}(K)$ dernières colonnes de V
   - **Fin Tant que**
   - Obtenir l'ordre maximum comme $n_1 \leftarrow n_1 + 1$ et faire $n_j \leftarrow n_1$, $j = 1, \ldots, \bar{s}$
   - Faire $V \leftarrow L(n_1, \bar{s})$ and $K \leftarrow (n_1+1)(n_u+n_y)$
2. Trouver les ordres restants n_j, $j = 2, \ldots, \bar{s}$ en utilisant le théorème 5.1
   - $j_o \leftarrow j_o + 1$
   - **Tant que** $\text{rang}(V) < M_{\bar{s}}(K) - |\mathscr{I}_{\bar{\rho}}|$
     - $n_j \leftarrow n_{j_o} - 1$, $j = j_o, \ldots, \bar{s}$
     - $\bar{\rho} \leftarrow \begin{bmatrix} n_1 & \cdots & n_{\bar{s}} \end{bmatrix}$
     - Calculer $\mathscr{I}_{\bar{\rho}}$ et supprimer les colonnes correspondantes dans V
   - **Fin Tant que**
   - Obtenir l'ordre n_{j_o} : $n_j \leftarrow n_{j_o} + 1$, $j = j_o, \ldots, \bar{s}$
   - Faire $V \leftarrow L(n_1, \bar{s})$
3. Aller à l'étape 2 jusqu'à ce que $j_o = \bar{s}$ ou jusqu'à ce que $n_{j_o} = 0$
4. Déterminer le nombre de sous-modèles $s = \text{card}(\{j : n_j > 0\})$

et définissons $W_{\bar{\rho}_{1,l}} = \frac{1}{N-\bar{n}} V_{\bar{\rho}_{1,l}} V_{\bar{\rho}_{1,l}}^\top$ comme la matrice obtenue de $\frac{1}{N-\bar{n}} L(\bar{n}, \bar{s})^\top L(\bar{n}, \bar{s})$ en retirant ses colonnes et lignes indexées par $\mathscr{I}_{\bar{\rho}_{1,l}}$. Pour tout $l = 0, \cdots, \bar{n}$, notons $\sigma_{1,l}$ la plus petite valeur singulière de la matrice $W_{\bar{\rho}_{1,l}}$. Selon le théorème 5.1, $W_{\bar{\rho}_{1,l}}$ a au moins un vecteur non nul dans son espace noyau pour $l \geq n_1$ et ainsi, $\sigma_{1,n_1} \simeq \cdots \simeq \sigma_{1,\bar{n}} \simeq \varepsilon_{1,n_1} = \frac{1}{\bar{n}-n_1}(\sigma_{1,n_1+1} + \cdots + \sigma_{1,\bar{n}})$ et devront être assez petits en comparaison avec $\sigma_{1,0}, \cdots, \sigma_{1,n_1-1}$. Ainsi, pour déterminer n_1, on a besoin de chercher le plus petit entier $l \in \{0, \cdots, \bar{n}\}$ pour lequel $\sigma_{1,l} \simeq \varepsilon_{1,l}$ dans un certain sens.

En suivant cette procédure, l'algorithme 5.2 peut être implémenté d'une manière plus efficace pour la détermination des ordres. Avec $\hat{n}_0 = \bar{n}$, et en se donnant un seuil de décision ε_0, la routine suivante permet d'éviter les tests de rang de l'algorithme 5.2.

Etape j :

$$\bar{\rho}_{j,l} = \begin{bmatrix} \hat{n}_1 & \cdots & \hat{n}_{j-1} & l & \cdots & l \end{bmatrix}, \; l = 0, \cdots, \hat{n}_{j-1},$$

$$\sigma_{j,l} = \min \lambda \left(W_{\bar{\rho}_{j,l}} \right), \; l = 0, \cdots, \hat{n}_{j-1},$$

$$\varepsilon_{j,l} = \frac{1}{\hat{n}_{j-1} - l} \left(\sigma_{j,l+1} + \cdots + \sigma_{j,\hat{n}_{j-1}} \right), \; l = 0, \cdots, \hat{n}_{j-1},$$

$$\mathcal{S}_j = \left\{ l = 0, \cdots, \hat{n}_{j-1} : |\sigma_{j,l} - \varepsilon_{j,l}| < \varepsilon_o \right\},$$

$$\hat{n}_j = \begin{cases} \min \left\{ l : l \in \mathcal{S}_j \right\}, & \text{si} \quad \mathcal{S}_j \neq \emptyset \\ \hat{n}_{j-1} & \text{sinon,} \end{cases}$$

$$j \leftarrow j + 1,$$

où $\lambda(W_{\bar{\rho}_{j,l}})$ désigne l'ensemble des valeurs propres de $W_{\bar{\rho}_{j,l}} = \frac{1}{N-\bar{n}} V_{\bar{\rho}_{j,l}} V_{\bar{\rho}_{j,l}}^\top$. Ici, j indexe une étape de la procédure d'estimation des ordres ; il prend la valeur initiale $j = 1$ et est incrémenté jusqu'à $j = \bar{s}$ ou jusqu'à une valeur j^* telle que $\hat{n}_{j^*} = 0$ (comme dans l'algorithme 5.2). Ainsi, dans la notation du type $\bar{\rho}_{j,l}$, l'indice j indique l'ordre de quel sous-modèle est en cours d'estimation tandis que l est une valeur possible de l'ordre recherché.

5.4 Réduction de complexité à travers une approche projective

L'algorithme algébrique proposé dans la section précédente devient très prohibitif en terme de coût de calcul quand les dimensions du système SARX deviennent grandes. Cela est dû au fait que le régresseur $x_n(t) \in \mathbb{R}^{K_1}$, construit avec les n_y sorties et les n_u entrées est assez long, induisant une augmentation exponentielle de la dimension $M_s(K_1)$ de l'espace des polynômes homogènes de degré s en K_1 variables. De plus, le nombre n_h de polynômes à estimer est, à moins de faire des hypothèses spécifiques comme dans les propositions 5.1 et 5.2, inconnu et cela, même quand les ordres et le nombre de sous-modèles sont supposés connus.

Dans cette section, au lieu de tenter d'estimer un nombre inconnu et potentiellement élevé de polynômes, nous proposons une méthode numériquement bien plus simple pour identifier les paramètres du modèle. L'idée consiste à transformer le système MIMO en un système mono-sortie MISO, et ainsi pouvoir utiliser seulement un polynôme de découplage pour partitionner les données selon les différents sous-modèles ARX. Une fois les données correctement partitionnées, le problème d'identification du système se ramène à un problème de régression standard pour chaque état discret.

Pour ce faire, commençons par noter que le système (5.1) peut se mettre sous la forme [1]

$$y(t) = \sum_{i=1}^{n_{\lambda_t}} a_{\lambda_t}^i y(t-i) + \sum_{i=0}^{n_{\lambda_t}} F_{\lambda_t}^i u(t-i) + e(t), \tag{5.20}$$

1. Notons que les ordres n_j dans (5.20) peuvent être plus grands que ceux de (5.1). Avec un abus de notation, nous garderons les mêmes notations pour les ordres.

où $\left\{F_j^i\right\}_{i=1,\cdots,n_j}^{j=1,\cdots,s} \in \mathbb{R}^{n_y \times n_u}$ et $\left\{a_j^i\right\}_{i=1,\cdots,n_j}^{j=1,\cdots,s}$ sont les coefficients du polynôme $z^{n_j} - a_j^1 z^{n_j-1} - \cdots - a_j^{n_j}$ qui encode les pôles du $j$ème sous-modèle.

Soit $\gamma = \begin{bmatrix}\gamma_1 & \cdots & \gamma_{n_y}\end{bmatrix}^\top$ un vecteur de nombres réels non nuls et soit $y_o(t) = \gamma^\top y(t) \in \mathbb{R}$ une combinaison linéaire des sorties du système. Alors, l'équation (5.20) peut être transformée au système MISO suivant

$$y_o(t) = \sum_{i=1}^{n_{\lambda_t}} a_{\lambda_t}^i y_o(t-i) + \sum_{i=0}^{n_{\lambda_t}} \gamma^\top F_{\lambda_t}^i u(t-i) + \gamma^\top e(t). \tag{5.21}$$

Remarque 5.2. *Dans le but de séparer les données selon leur sous-modèle générateur respectif, on pourrait être tenté de considérer une seule sortie $y_j(t)$ dans l'équation (5.20) au lieu d'une combinaison de toutes les n_y sorties comme nous l'avons suggéré. Cependant, un problème dans une telle démarche est que, par suite d'une éventuelle compensation de zéros et de pôles, le système MISO qui a pour sortie $y_j(t)$, peut se trouver être commun à plusieurs sous-modèles différents rendant ainsi impossible une distinction de ces modes. En choisissant de façon aléatoire les poids de combinaison linéaire des sorties, de telles situations dégénérées peuvent être évitées presque sûrement.*

En introduisant la sortie combinée $y_o(t)$, on obtient seulement un polynôme de découplage hybride qui est plus facile à manipuler. Cependant, dans le même temps, les paramètres des différents sous-modèles ont été combinés. Cela soulève naturellement la question de savoir si cette opération préserve la discernabilité des différents sous-modèles constitutifs du système à commutations. En fait, selon la valeur du vecteur de poids γ, deux sous-modèles initialement différents dans (5.20) peuvent se réduire au même sous-modèle dans (5.21). Pour analyser ce risque, définissons

$$F_j = \begin{bmatrix}F_j^{n_j} & \cdots & F_j^1 & F_j^0\end{bmatrix} \in \mathbb{R}^{n_y \times (n_j+1)n_u} \quad \text{et} \quad a_j = \begin{bmatrix}a_j^{n_j} & \cdots & a_j^1\end{bmatrix}^\top \in \mathbb{R}^{n_j}. \tag{5.22}$$

De (5.21) on peut voir que deux modes différents i et j deviennent indiscernables après la transformation précédente par γ, s'ils ont le même ordre ($n_i = n_j$), les mêmes dynamiques ($a_i = a_j$) et si de plus $\left(F_i^\top - F_j^\top\right)\gamma = 0$, c'est-à-dire quand γ est dans $\text{null}(F_i^\top - F_j^\top)$. Si les matrices F_j étaient connues, on pourrait sélectionner immédiatement un γ qui ne satisfait pas à cette condition. Mais ces matrices sont précisément ce que nous recherchons. La question est, sans aucune connaissance des F_j, comment peut-on choisir γ de façon à garantir pour tous $i \neq j$, $\gamma \notin \text{null}(F_i^\top - F_j^\top)$. En fait, il n'est pas difficile de voir qu'en générant γ de façon aléatoire suivant une distribution uniforme par exemple, cette condition est satisfaite avec une probabilité de un.[1] Ainsi, deux sous-modèles qui sont distincts dans le système original (5.1) demeurent distincts après la transformation (5.21). Cependant, la séparabilité des modes, c'est-à-dire la mesure de combien proches ou distincts sont les modes, peut se trouver affectée.

1. Cela peut être démontré en suivant un raisonnement similaire à celui utilisé dans la preuve de la proposition 3.3.

Partant du modèle (5.21), nous redéfinissons le vecteur de paramètres $\bar{\theta}_j$ et le régresseur $\bar{x}_n(t)$ par

$$\begin{aligned}
\theta_j &= \begin{bmatrix} \gamma^\top F_j^{n_j} & a_j^{n_j} & \cdots & \gamma^\top F_j^1 & a_j^1 & \gamma^\top F_j^0 & 1 \end{bmatrix}^\top \in \mathbb{R}^{K_j},\ j = 1, \ldots, s, \\
\bar{\theta}_j &= \begin{bmatrix} 0_{q_j}^\top & \theta_j^\top \end{bmatrix}^\top \in \mathbb{R}^K, \\
\bar{x}_{n_j}(t) &= \begin{bmatrix} u(t-n_j)^\top & y_o(t-n_j) & \cdots & u(t)^\top & -y_o(t) \end{bmatrix}^\top \in \mathbb{R}^{K_j}, \\
\bar{x}_n(t) &= \begin{bmatrix} u(t-n)^\top & y_o(t-n) & \cdots & u(t)^\top & -y_o(t) \end{bmatrix}^\top \in \mathbb{R}^K,
\end{aligned} \tag{5.23}$$

avec K et K_j définis comme précédemment mais avec $n_y = 1$, c'est-à-dire, $K = (n+1)(n_u+1)$ et $K_j = (n_j+1)(n_u+1)$. On peut regarder la plus petite valeur singulière $\sigma_0(X(\gamma))$ de $X(\gamma) = \begin{bmatrix} \bar{x}_{\bar{n}}(\bar{n}+1) & \cdots & \bar{x}_{\bar{n}}(N) \end{bmatrix}$, comme une certaine mesure de combien il est vraisemblable que les données puissent être ajustées à un seul sous-espace de $\mathbb{R}^{\bar{K}}$, $\bar{K} = (\bar{n}+1)(n_u+1)$. En effet, il apparaît intuitivement que plus fortement les sous-espaces sont distincts, plus le nombre $\sigma_0(X(\gamma))$ devrait être grand. Pour le voir, supposons par exemple qu'on veuille ajuster à un seul sous-espace strict de $\mathbb{R}^{\bar{K}}$, de dimension d, des données du type $X(\gamma)$ qui se trouvent en réalité dans deux sous-espaces stricts de $\mathbb{R}^{\bar{K}}$, ayant la même dimension d. Si ces derniers sous-espaces sont peu distincts, alors on commet une erreur de modélisation relativement faible. Par contre, s'ils sont fortement distincts, la procédure résulte en une incohérence plus forte caractérisée par la valeur de $\sigma_0(X(\gamma))$.
En partant de cette remarque, nous suggérons, pour préserver la séparabilité des modes, de choisir γ par exemple de la façon suivante

$$\gamma^* = \arg\max_{\gamma:\|\gamma\|\leq 1} \frac{\sigma_0(X(\gamma))}{\sigma_{\max}(X(\gamma))}, \tag{5.24}$$

où $\sigma_{\max}(X(\gamma))$ est la plus grande valeur singulière de $X(\gamma)$. Puisque (5.24) peut être, dans bien des cas, un problème d'optimisation fastidieux, une alternative peut être de choisir plusieurs candidats γ de façon à ce que $\sigma_0(X(\gamma))$ soit dans une certaine proportion de $\sigma_{\max}(X(\gamma))$.

Etant donné γ, nous pouvons commencer la procédure d'identification. Une fois encore, nous nous affranchissons de la dépendance de l'équation du système, des commutations en considérant le polynôme de découplage suivant qui s'annule sur les données quelque soit leur sous-modèle générateur :

$$g\big(\bar{x}_n(t)\big) = \prod_{j=1}^{s} \Big(\bar{\theta}_j^\top \bar{x}_n(t)\Big) = h^\top \nu_s\big(\bar{x}_n(t)\big) = 0, \tag{5.25}$$

ce qui équivaut, avec un paramétrage minimal, à

$$\prod_{j=1}^{s} \Big(\theta_j^\top \bar{x}_{n_j}(t)\Big) = h_\rho^\top \eta_\rho\big(\bar{x}_n(t)\big) = 0. \tag{5.26}$$

La résolution de l'équation (5.25) est un cas particulier simple ($n_y = 1$) du cas étudié à la section 5.3.2. Nous commençons par déterminer les ordres et le nombre de sous-modèles du modèle (5.21) plus simplement que dans la section 5.3.2 grâce au théorème suivant.

Theorème 5.2. *Considérons les données* $\{u(t), y_o(t)\}_{t=1}^{N}$ *générées par un modèle MISO du type (5.20), constitué de* s *sous-modèles d'ordres* n_j, $j = 1, \ldots, s$. *Supposons que ces données soient suffisamment excitantes dans le sens de la définition 5.1. Désignons par* $\bar{n} \geq \max(n_j)$ *et* $\bar{s} \geq s$ *des majorants respectifs des ordres des sous-modèles et de leur nombre. Alors on a :*

1. $\text{null}(L(r, \bar{s})) = \{0\}$ *ssi* $r < \max(n_j)$.
2. $\text{null}(L(\bar{n}, l)) = \{0\}$ *ssi* $l < s$ *[131].*

En d'autres termes, si r est un entier strictement inférieur à l'un des ordres des différents sous-modèles, alors il n'existe aucun polynôme homogène de degré $\bar{s}$ s'annulant sur l'ensemble des données $x_r(t)$. De même, pour l strictement plus petit que le nombre réel s d'états discrets, il n' y a aucun polynôme homogène de degré l qui puisse s'annuler sur l'ensemble des $x_{\bar{n}}(t)$ même avec $\bar{n} \geq \max(n_j)$. Notons bien que cette dernière assertion est vraie seulement dans le cas d'un mélange de données distribuées sur des hyperplans (cf. Remarque 5.1), ce qui est le cas pour des données générées par le MISO SARX (5.21). Un exemple illustratif de ce fait est également donné par la figure 5.3.
En procédant comme dans l'algorithme 5.2, on peut utiliser le théorème 5.2 pour identifier les ordres et le nombre de sous-modèles. La procédure de détermination de $\bar{\theta}_j$ reste globalement la même que dans la sous-section 5.3.2 :

1. Trouver les ordres et le nombre de sous-modèles en faisant usage du théorème 5.2,
2. Obtenir h_ρ comme un élément non nul quelconque de $\text{null}(V_\rho)$ (qui doit être de dimension un lorsque les données sont suffisamment excitantes), et
3. Compléter si nécessaire h_ρ avec des zéros pour former un $h \in \mathbb{R}^{M_s(K)}$ de manière à ce que les entrées de h définies par $\mathscr{I}_\rho$ soient toutes nulles.

Disposant de h, les paramètres définis en (5.23) peuvent être obtenus de la dérivée du polynôme g (cf. Eq. (5.25)) comme démontré dans [131] :

$$\bar{\theta}_j = \frac{\nabla g(z_j)}{e_K^\top \nabla g(z_j)}, j = 1, \ldots, s, \tag{5.27}$$

où z_j est un point de $S_j \setminus \cup_{i \neq j}^{s} S_i$, $S_j = \left\{ x \in \mathbb{R}^K : \bar{\theta}_j^\top x = 0 \right\}$, e_K est un vecteur de longueur K avec 1 comme dernier élément et 0 partout ailleurs.

5.4.1 Classification des données

Le calcul de $\bar{\theta}_j$ pour chaque sous-modèle passe par la détermination d'un point z_j appartenant à S_j à l'exclusion de tout autre S_i, $i \neq j = 1, \ldots, s$. Ce point de S_j peut être pris comme $z_j = \bar{x}_n(\tau_j)$, où

$$\tau_j = \arg\min_{t \in \mathcal{D}_j} \left| \frac{\nabla g\left(\bar{x}_n(t)\right)^\top \bar{x}_n(t)}{e_K^\top \nabla g\left(\bar{x}_n(t)\right)} \right|, \tag{5.28}$$

avec $\mathcal{D}_1 = \{t : \nabla g(\bar{x}_n(t)) \neq 0\}$ et $\mathcal{D}_j = \mathcal{D}_1 \cap \{t : \bar{\theta}_i^\top \bar{x}_n(t) \neq 0, i = 1, ..., j-1\}$, pour $j > 1$.

Rappelons que l'estimation des vecteurs $\left\{\bar{\theta}_j\right\}_{j=1}^{s}$ associés à la sortie combinée $y_o(t)$ est seulement une étape intermédiaire dans notre procédé global de recouvrement des matrices a_j et F_j qui définissent chaque sous-système du système original (5.20). Maintenant, partant des paramètres $\bar{\theta}_j$ obtenus, nous pouvons procéder à la détermination de l'état discret du système (5.21) qui est le même que celui de (5.1) et ainsi calculer finalement le modèle recherché. Dans le but de prémunir le résultat de la classification de l'effet d'éventuelles données aberrantes, nous choisissons un seuil de performance $\varepsilon < 1$ pour définir les règles de décision :

$$\begin{aligned} &\text{Si} \quad \min_j \Delta(\bar{\theta}_j, \bar{x}_n(t)) > \varepsilon \left\|\bar{x}_n(t)\right\|, \quad \text{alors} \quad \lambda_t \quad \text{est indécidable.} \\ &\text{Si} \quad \min_j \Delta(\bar{\theta}_j, \bar{x}_n(t)) \leq \varepsilon \left\|\bar{x}_n(t)\right\|, \quad \text{alors} \quad \lambda_t = \arg\min_j \Delta\left(\bar{\theta}_j, \bar{x}_n(t)\right). \end{aligned}$$

Ici, $\Delta(\bar{\theta}_j, \bar{x}_n(t)) = \dfrac{\left|\bar{\theta}_j^\top \bar{x}_n(t)\right|}{\left\|\bar{\theta}_j\right\|}$ est la distance du point $\bar{x}_n(t)$ à l'hyperplan linéaire S_j défini par son vecteur normal $\bar{\theta}_j$. Cette classification nous permet de collecter dans $\mathscr{X}_j = \{t > \bar{n} : \lambda_t = j\} = \left\{t_1^j, \cdots, t_{N_j}^j\right\}$, $j = 1, \ldots, s$, les indices temporels des régresseurs générés par le sous-modèle j.

5.4.2 Estimation des paramètres des sous-modèles

En nous basant sur les résultats de la classification précédente, nous savons désormais quelle donnée correspond à quel mode générateur. Ainsi, il ne reste plus qu'à déterminer les paramètres de chaque mode j à partir des données indexées par $\mathscr{X}_j$. Considérons un sous-modèle linéaire j de (5.20), d'ordre n_j. Pour $t \in \mathscr{X}_j$, nous définissons

$$\Phi_j^y(t) := \begin{bmatrix} y(t-n_j) & \cdots & y(t-1) \end{bmatrix} \in \mathbb{R}^{n_y \times n_j}, \tag{5.29}$$

$$\phi_j^u(t) := \begin{bmatrix} u(t-n_j)^\top & \cdots & u(t)^\top \end{bmatrix}^\top \in \mathbb{R}^{(n_j+1)n_u}. \tag{5.30}$$

Les paramètres des sous-modèles de (5.20) peuvent enfin être calculés comme la solution du problème de régression linéaire suivant

$$y(t) = \begin{bmatrix} \Phi_j^y(t) & \phi_j^u(t)^\top \otimes I_{n_y} \end{bmatrix} \begin{bmatrix} a_j \\ \mathrm{vec}(F_j) \end{bmatrix} + e(t),\ t \in \mathscr{X}_j. \tag{5.31}$$

Cette équation est obtenue en faisant usage de l'identité $\mathrm{vec}(AXB) = (B^\top \otimes A)\mathrm{vec}(X)$, où le symbole $\otimes$ fait référence au produit de Kronecker et $\mathrm{vec}(\cdot)$ est l'opérateur de vectorisation. Notons que dans toute la procédure, les vecteurs a_j, $j = 1, \ldots, s$, sont estimés deux fois. La première estimée (obtenue à travers $\bar{\theta}_j$) est considérée comme grossière puisqu'elle résulte du choix d'un point d'évaluation z_j qui peut être affecté par du bruit. Il n'est pas vraiment nécessaire que cette première estimée soit assez précise ; elle a juste besoin de permettre une discrimination des différents modes en concurrence. La seconde estimée obtenue par (5.31) devrait en revanche être de meilleure qualité.

5.5 Identification récursive de modèles MIMO SARX

La méthode algébro-géométrique que nous avons présentée en deux versions dans les sections 5.3 et 5.4 de ce chapitre pour l'identification de MIMO SARX, s'applique à un ensemble fini de données dont la collection est achevée. En particulier, la version projective de la méthode (cf. Section 5.4) considère l'estimation d'un seul polynôme de découplage dont les dérivées évaluées en des points spécifiques z_j, permettent le partitionnement hors ligne des données, c'est-à-dire l'obtention de l'état discret. Cependant, en suivant cette procédure, il est difficile de concevoir, comment on pourrait par exemple, partitionner une infinité d'observations en un temps fini. En fait, la complexité numérique des algorithmes opérant hors ligne est directement fonction du nombre d'observations à traiter. Par ailleurs, il peut être nécessaire ou même indispensable dans certaines applications de la vie réelle, de traiter les données une à une, c'est-à-dire à mesure qu'elles sont capturées. Pour répondre à ce besoin pratique, nous discutons dans cette section une version récursive de la méthode algébrique pour des systèmes MIMO SARX. En nous inspirant des travaux de Vidal *et al.* dans [126], [53], [127] pour des systèmes SISO, nous allons présenter dans cette sous-section, un schéma d'identification récursive de modèles MIMO SARX, sous l'hypothèse que le nombre de sous-modèles ainsi que les ordres sont connus *a priori*. Dans cette méthode, la transformation polynomiale de la section 5.3 n'est plus utilisée que dans une étape intermédiaire d'extraction de l'état discret. Etant donné l'état discret, les paramètres des sous-modèles sont obtenus par la méthode classique des moindres carrés récursifs.

5.5.1 Schéma d'identification

La stratégie suivie dans la méthode qui sera présentée ci-dessous consiste à utiliser la transformation polynomiale (estimation et differentiation de polynômes) pour extraire l'état discret courant du système puis l'algorithme des moindres carrés récursifs pour mettre à jour les paramètres du sous-modèle actif à l'instant courant.

Ce principe est comparable à celui bien connu de l'algorithme EM (Expectation Maximization) dans lequel les données sont traitées séquentiellement. Une observation donnée est attribuée au mode qui, au vu d'un certain critère de décision, apparaît comme celui qui, de tous les modes, est le générateur le plus vraisemblable de cette observation. Par conséquent, les paramètres de ce dernier sous-modèle sont adaptés en fonction de l'observation tandis les paramètres des autres sous-modèles restent inchangés. Dans la mise en oeuvre d'un tel schéma, des seuils peuvent accessoirement être définis en vue d'éliminer automatiquement le plus de *points aberrants* possibles. Cela signifie qu'une observation sera classée seulement si son degré d'appartenance à un mode est supérieur à certain seuil de confiance. Sinon l'observation en question est dite inclassable et ainsi, elle est simplement rejetée ou mise en attente d'un traitement ultérieur.

Nous commençons par rappeler l'algorithme des Moindres Carrés Récursifs (MCR) [75]. Pour une matrice $A \in \mathbb{R}^{m\times d}$ à estimer à partir d'une équation de régression linéaire $y(t) = Ax(t)$, $t = 1, \ldots, \infty$, soit $\hat{A}(0)$ la valeur initiale de A et $L(0) = \sigma I_d$, avec σ un scalaire, la valeur initiale de la matrice d'autocorrelation de (en fait, il s'agit de son inverse) du vecteur d'observation $x(t)$. Alors, étant donné une nouvelle

observation $\big(x(t), y(t)\big)$, la mise à jour du couple $\Big\{\hat{A}(t-1), L(t-1)\Big\}$ par l'algorithme MCR, est donnée par l'algorithme 5.3 dans lequel α est un facteur d'oubli, $\alpha \in [0,\ 1]$.

Algorithme 5.3 $\Big\{\hat{A}(t), L(t)\Big\} = \text{MCR}\Big\{\hat{A}(t-1), L(t-1), \big(x(t), y(t)\big)\Big\}$

$$
\begin{aligned}
\varepsilon(t) &= y(t) - \hat{A}(t-1)x(t) \\
q(t) &= \frac{L(t-1)x(t)}{\alpha + x(t)^\top L(t-1)x(t)} \\
\hat{A}(t) &= \hat{A}(t-1) + \varepsilon(t)q(t)^\top \\
L(t) &= \alpha^{-1}L(t-1) - \alpha^{-1}q(t)x(t)^\top L(t-1),
\end{aligned}
$$

La convergence de cet algorithme a fait dans la littérature, l'objet de bien d'investigations [63]. Si les données $\big(x(t), y(t)\big)$ ne sont pas sujettes à des perturbations, c'est-à-dire si elles suivent rigoureusement le modèle linéaire $y(t) = Ax(t)$, alors en supposant que le facteur d'oubli α est strictement inférieur à l'unité et que la séquence $\{x(t)\}$ est excitante de façon persistante [4] dans le sens où il existe $m_o \in \mathbb{N}$, $\rho_1 > 0$, $\rho_2 > 0$ tels que

$$\rho_1 I_d < \sum_{t=q}^{q+m_o} x(t)x(t)^\top < \rho_2 I_d \quad \forall\, q, \tag{5.32}$$

alors on peut montrer que $\hat{A}(t)$ converge exponentiellement vers sa vraie valeur A pour toute valeur initiale $A(0)$ (cf. [63] pour une preuve). Dans la notation ci-dessus, la relation $A < B$ entre deux matrices signifie que $B - A$ définie-positive.

5.5.2 Reformulation du modèle SARX

Maintenant nous retournons au modèle considéré qui est toujours de la forme donnée par (5.20). Cependant, nous considérons plutôt la forme (5.31) qui résulte de (5.20) par un jeu d'écriture :

$$y(t) = \begin{bmatrix} \Phi^y_{\lambda_t}(t) & \phi^u_{\lambda_t}(t)^\top \otimes I_{n_y} \end{bmatrix} \begin{bmatrix} a_{\lambda_t} \\ \text{vec}(F_{\lambda_t}) \end{bmatrix} + e(t). \tag{5.33}$$

Pour des raisons de commodité, notons

$$
\begin{aligned}
\vartheta_{\lambda_t} &= \begin{bmatrix} a_{\lambda_t}^\top & \text{vec}(F_{\lambda_t})^\top \end{bmatrix}^\top \in \mathbb{R}^{(n_{\lambda_t} + (n_{\lambda_t}+1)n_u n_y)}, \\
\Psi_{\lambda_t}(t) &= \begin{bmatrix} \Phi^y_{\lambda_t}(t) & \phi^u_{\lambda_t}(t)^\top \otimes I_{n_y} \end{bmatrix} \in \mathbb{R}^{n_y \times (n_{\lambda_t} + (n_{\lambda_t}+1)n_u n_y)}.
\end{aligned} \tag{5.34}
$$

Alors, l'équation du modèle SARX s'écrit

$$y(t) = \Psi_{\lambda_t}(t)\vartheta_{\lambda_t} + e(t), \tag{5.35}$$

où la matrice d'information $\Psi_{\lambda_t}(t)$ est complètement formée à partir des mesures disponibles et ϑ_{λ_t} est le vecteur de paramètres (à estimer) correspondant au sous-modèle indexé par λ_t. A partir du modèle (5.35), il est techniquement difficile d'estimer directement le vecteur de paramètres ϑ_{λ_t} de façon récursive au moyen des MCR. Cette difficulté tient essentiellement au fait que $\Psi_{\lambda_t}(t)$ est ici une matrice au lieu d'un vecteur comme dans le cas classique de l'algorithme 5.3. De ce fait, l'application du lemme d'inversion matricielle (dont est dérivé l'algorithme MCR) [75] n'est plus bénéfique. Pour cette raison, nous procédons d'abord à une mise en forme du modèle (5.35) en vue d'une application des MCR. Dans ce but, considérons un ensemble de vecteurs $z(t) \in \mathbb{R}^{n_y}$ indexés par le temps tel que $z(t)^\top y(t) \neq 0$ presque chaque fois que $y(t) \neq 0$. Si nous multiplions l'équation (5.35) à gauche par $z(t)^\top$, il vient

$$\tilde{y}(t) = \phi_{\lambda_t}(t)^\top \vartheta_{\lambda_t} + \tilde{e}(t), \tag{5.36}$$

avec $\tilde{y}(t) = z(t)^\top y(t) \in \mathbb{R}$, $\tilde{e}(t) = z(t)^\top e(t) \in \mathbb{R}$ et $\phi_{\lambda_t}(t) = \Psi_{\lambda_t}(t)^\top z(t)$. Nous obtenons ainsi un modèle convenablement dimensionné pour l'application des MCR. Cependant, il nous faut encore choisir $z(t)$, ce qui peut se faire simplement en posant $z(t) = y(t)$ ou plus généralement $z(t) = y(t-k)$, avec k un entier naturel, ou encore en remplaçant $z(t)$ par une certaine combinaison linéaire filtrée des entrées et sorties passées.

5.5.3 Identification récursive

Nous arrivons maintenant à l'identification récursive du modèle SARX (5.20) à travers la relation (5.36). Cependant, la détermination de l'état discret est basée sur l'estimation du modèle auxiliaire (5.21), comme dans la section 5.4. Ce choix est justifié par le fait que le régresseur obtenu à partir de (5.21) est d'une dimension en général moins importante que celle du régresseur construit à partir de (5.36) (c'est-à-dire, $n_{\lambda_t} + (n_{\lambda_t}+1)n_u \leq n_{\lambda_t} + (n_{\lambda_t}+1)n_u n_y$) et donc plus approprié pour l'application de la transformation de Veronese.

Pour des raisons de simplicité [1], nous considérons seulement le cas où les ordres des sous-modèles sont connus et tous égaux à n. Nous supposons aussi que le nombre s de sous-modèles est connu *a priori* bien qu'en pratique la connaissance d'un majorant de celui-ci soit suffisante [126]. Alors, il est facile de remarquer que la dernière entrée du vecteur de coefficients h définissant le polynôme (5.25), vaut 1 et que le dernier élément de $\nu_s(\bar{x}_n(t))$ vaut $-\xi(t) \in \mathbb{R}$, avec $\xi(t) = (-1)^{s+1} y_o(t)^s$. Notons $h_0 \in \mathbb{R}^{M_s(K)-1}$ le vecteur constitué des $M_s(K)-1$ entrées restantes de h et $\varphi(t)$ les entrées de $\nu_s(\bar{x}_n(t))$ correspondantes. Ainsi, pour déterminer l'état discret, nous devons estimer de manière récursive le modèle linéaire $\xi(t) = h_0^\top \varphi(t)$ défini par le vecteur de paramètres hybride. Connaissant $h_0(t)$, nous déduisons

1. on pourrait étudier aussi le cas où ni les ordres, ni le nombre de sous-modèles ne sont connus. Malheureusement, dans ce cas, les règles de décisions sont si inextricables qu'il est difficile d'obtenir un algorithme efficace dans un contexte récursif. Notons toutefois qu'il n'est pas nécessaire de connaître le nombre exact de sous-modèles [126] ; la connaissance d'un majorant de ce nombre suffit à l'obtention de bonnes estimées.

$h(t) = \begin{bmatrix} h_0(t)^\top & 1 \end{bmatrix}^\top$ puis le polynôme annihilateur $g_{s,t}(z) = h(t)^\top \nu_s(z)$ et enfin le vecteur auxiliaire

$$\omega_h(t) = \frac{\nabla \nu_s(\bar{x}_n(t))^\top h(t)}{e_K^\top \nabla \nu_s(\bar{x}_n(t))^\top h(t)} \in \mathbb{R}^K,$$

dont le rôle ici est de permettre la détermination de l'état discret courant. Nous lançons en parallèle $s+1$ identifieurs MCR dont un algorithme par sous-modèle et un algorithme pour l'estimation du vecteur de paramètres hybride $h(t)$. Cela est implémenté selon les étapes suivantes.
Chaque fois qu'une nouvelle observation $(x_n(t), y(t))$ est mesurée,

1. Mettre à jour le vecteur de paramètres hybride $h_0(t)$

$$\begin{aligned} &\left\{h_0(t)^\top, P_h(t)\right\} = \text{MCR}\left\{h_0(t-1)^\top, P_h(t-1), \big(\varphi(t), \xi(t)\big)\right\}, \\ &\hat{h}(t) = \begin{bmatrix} h_0(t)^\top & 1 \end{bmatrix}^\top, \in \mathbb{R}^{M_s(K)} \\ &\hat{\omega}_h(t) = \frac{\nabla \nu_s\left(\bar{x}_n(t)\right)^\top \hat{h}(t)}{e_K^\top \nabla \nu_s\left(\bar{x}_n(t)\right)^\top \hat{h}(t)} \in \mathbb{R}^K. \end{aligned}$$

2. Déterminer le mode actif selon les règles

$$\begin{cases} \text{Si} \quad \Delta(\hat{\omega}_h(t), \bar{x}_n(t)) \leq \varepsilon, \quad \text{alors } \hat{\lambda}_t = \underset{j=1,\ldots,s}{\arg\max}\, \Delta(\hat{\omega}_h(t), \hat{\theta}_j(t-1)), \\ \text{Sinon, la paire } (\bar{x}_n(t), y(t)) \text{ est dite indécidable.} \end{cases}$$

où $\Delta(\hat{\omega}_h(t), \bar{x}_n(t)) = \dfrac{\left|\hat{\omega}_h(t)^\top \bar{x}_n(t)\right|}{\left\|\hat{\omega}_h(t)\right\| \cdot \|\bar{x}_n(t)\|}$ mesure la correlation entre les vecteurs $\hat{\omega}_h(t)$ et $\bar{x}_n(t)$ et $\varepsilon \in [0,1]$, un certain seuil de décision utilisé en vue d'éliminer de possibles *points aberrants*.

3. Mettre à jour les paramètres du mode actif $\hat{\lambda}_t$

$$\left\{\hat{\vartheta}_{\hat{\lambda}_t}(t)^\top, L_{\hat{\lambda}_t}(t)\right\} = \text{MCR}\left\{\hat{\vartheta}_{\hat{\lambda}_t}(t-1)^\top, L_{\hat{\lambda}_t}(t-1), \big(x_n(t), \tilde{y}(t)\big)\right\},$$

où $\vartheta_{\hat{\lambda}_t}$ est le vecteur de paramètres du sous-modèle $\hat{\lambda}_t$.

4. Pour tous les modes $j \neq \hat{\lambda}_t$, aucun changement dans les paramètres.

Dans cette procédure, nous rappelons que les vecteurs $\hat{\theta}_j(t)$ ne sont pas directement estimés mais construits à chaque étape à partir de $\hat{\vartheta}_j(t)$ et selon la formule (5.23), avec $n_j = \bar{n} = n$ pour tout $j = 1, \ldots, s$. Notons qu'une convergence des estimées $\hat{\vartheta}_{\hat{\lambda}_t}(t)$ est subordonnée d'une certaine manière, à celle de $\hat{\lambda}_t$ qui dépend à son tour de la convergence du vecteur de paramètres hybride $h(t)$. Des commentaires donnés dans la sous-section 5.5.1 sur l'algorithme MCR, nous savons que si la séquence de données $\{\varphi(t)\}$ satisfait à une condition d'excitation du type (5.32), alors $\hat{h}(t)$ converge exponentiellement vite vers $h = \begin{bmatrix} h_0^\top & 1 \end{bmatrix}$. Par conséquent $\hat{\vartheta}_j(t)$ aussi convergera vers ϑ_j pour tout j si nous supposons que tous les états discrets sont fréquemment visités par le système et cela, une infinité de fois quand $t \to \infty$.

Une remarque importante est que la condition d'excitation persistante (5.32) pour la détermination du vecteur $h_0(t)$ par les MCR impose une contrainte supplémentaire sur les commutations du système. En effet, avec $\varphi(t) = \begin{bmatrix} I_{M_s(K)-1} & 0_{M_s(K)-1,1} \end{bmatrix} \nu_s(\bar{x}_n(t))$, cette condition s'écrit : il existe $m_o \in \mathbb{N}$, $\beta_1 > 0$, $\beta_2 > 0$ tels que

$$\beta_1 I < \sum_{t=q}^{q+m_o} \varphi(t)\varphi(t)^\top < \beta_2 I \quad \forall\, q. \tag{5.37}$$

Mais pour qu'une telle contrainte soit respectée, il est nécessaire que tous les s modes du système soient suffisamment visités sur chaque fenêtre temporelle $[q,\ q+m_o]$. Pour plus de commentaires sur cet aspect, nous nous référons à l'article [126] qui traite de l'identification récursive d'un système SISO.
Cependant, pour avoir $\hat{\lambda}_t \to \lambda_t$ quand $t \to \infty$, il n'est pas vraiment nécessaire que $\hat{h}(t) \to h$. En référence au critère de décision dans l'algorithme précédent, pour obtenir $\hat{\lambda}_t = \lambda_t = i$, il suffit que les estimées $\hat{\theta}_k(t-1)$, $k = 1, \ldots, s$, vérifient $\Delta(\hat{\omega}_h(t), \hat{\theta}_i(t-1)) > \Delta(\hat{\omega}_h(t), \hat{\theta}_j(t-1))$ pour tout $j \neq i$, chaque fois que les vecteurs de paramètres réels $\theta_k(t-1)$ satisfont à $\Delta(\hat{\omega}_h(t), \theta_i(t-1)) > \Delta(\hat{\omega}_h(t), \theta_j(t-1))$ pour tout $j \neq i$.

5.6 Une alternative à l'approche algébrique

5.6.1 Motivation

L'approche algébro-géométrique fournit une solution exacte au problème d'identification de modèles SARX. En revanche, son application à des systèmes de grandes dimensions (ordres, nombre de sorties, d'entrées et de sous-modèles) pourrait se heurter à un problème important de dimensionalité. En effet, la dimension $M_s(K)$ de l'espace des données *embarquées* augmente rapidement avec le nombre s de sous-modèles et la dimension K de l'espace de régression, c'est-à-dire avec l'ordre maximum, le nombre d'entrées et le nombre de sorties. Ainsi, pour réaliser un compromis entre la précision et la dimensionalité, il peut être nécessaire de recourir à des techniques théoriquement moins performantes mais capables de tolérer des dimensions de systèmes plus élevées.
Nous présentons donc dans la suite une alternative à l'approche algébro-géométrique, basée sur une alternance entre l'assignation des données aux différents modes du modèle SARX et la mise à jour de leurs paramètres. Tandis que seule la connaissance d'un majorant du nombre s de sous-modèles est requise, nous supposons que les ordres des sous-modèles sont connus.
Le principe de la méthode est similaire à celui de la technique discutée dans la sous-section 5.5.3 à la différence que nous n'estimons plus aucun polynôme de découplage hybride pour la classification des données. Disposant des mesures entrées-sorties générées par un modèle SARX du type (5.36) sur un certain horizon de temps $[0,\ t-1]$, nous désirons à la fois assigner la paire de données courante $\big(\phi_{\lambda_t}(t), \tilde{y}(t)\big)$ au sous-modèle qui en est le générateur le plus vraisemblable et mettre à jour le vecteur de paramètres du sous-modèle correspondant.
Pour cela, nous représentons, en référence à l'algorithme 5.3, chaque mode i par le couple $\Big\{\hat{\vartheta}_i(t), L_i(t)\Big\}$, où $\hat{\vartheta}_i(t)$ est le vecteur de paramètres du sous-modèle (estimée) i à l'instant t. L'adéquation entre une

nouvelle donnée $(\phi_{\lambda_t}(t), \tilde{y}(t))$ et chacun des s sous-modèles est mesurée par un certain critère de décision.

5.6.2 Choix d'un critère de décision

Le rôle du critère de décision est de nous permettre de décider de l'attribution d'un couple de données $\left(\phi_{\lambda_t}(t), \tilde{y}(t)\right)$ à un mode de fonctionnement ou à un autre. Etant donné la rapidité possible des commutations, ce critère a besoin de permettre une décision instantanée c'est-à-dire faisant intervenir le moins possible les informations passées. Ainsi, le critère de décision peut être sélectionné comme une fonction directe

– de l'erreur *a priori*

$$\begin{aligned} \varepsilon_i(t) &= \tilde{y}(t) - \hat{\vartheta}_i(t-1)^\top \phi_i(t) \\ &= \bar{\vartheta}_i(t-1)^\top \bar{\phi}_i(t), i = 1, \ldots, s, \end{aligned} \tag{5.38}$$

avec $\bar{\phi}_{\lambda_t} = \begin{bmatrix} -\tilde{y}(t) & \phi_{\lambda_t}^\top \end{bmatrix}^\top$ et $\bar{\vartheta}_{\lambda_t} = \begin{bmatrix} 1 & \vartheta_{\lambda_t}^\top \end{bmatrix}^\top$,

– ou de l'erreur a posteriori

$$\begin{aligned} \eta_i(t) &= y(t) - \hat{\vartheta}_i(t)^\top \phi_i(t) \\ &= \hat{\bar{\vartheta}}_i(t)^\top \bar{\phi}_i(t) \\ &= \frac{\alpha \varepsilon_i(t)}{\alpha + \phi_i(t)^\top L_i(t-1) \phi_i(t)}, i = 1, \ldots, s. \end{aligned} \tag{5.39}$$

Les erreurs (5.38) et (5.39) peuvent être interprétées comme des projections orthogonales du vecteur $\bar{\phi}_i(t)$ respectivement sur les directions définies par $\hat{\bar{\vartheta}}_i(t-1)$ et $\hat{\bar{\vartheta}}_i(t)$. Puisque le but de l'exercice est de déterminer les directions $\bar{\vartheta}_i$, il convient de normaliser les vecteurs $\hat{\bar{\vartheta}}_i(t-1)$ et $\hat{\bar{\vartheta}}_i(t)$ dans les équations précédentes. Pour $\hat{\bar{\vartheta}}_i \neq 0$ et $\phi_i(t) \neq 0$, on obtient les erreurs normalisées comme

$$\varepsilon_i^o(t) = \frac{\hat{\bar{\vartheta}}_i(t-1)^\top \bar{\phi}_i(t)}{\left\|\hat{\bar{\vartheta}}_i(t-1)\right\| \cdot \left\|\phi_i(t)\right\|} \quad \text{et} \quad \eta_i^o(t) = \frac{\hat{\bar{\vartheta}}_i(t)^\top \bar{\phi}_i(t)}{\left\|\hat{\bar{\vartheta}}_i(t)\right\| \cdot \left\|\phi_i(t)\right\|}.$$

La décision consistera alors à comparer les normes des projections orthogonales de $\bar{\phi}_i(t)$ sur les directions $\hat{\bar{\vartheta}}_i(t)$, $i = 1, \ldots, s$. La normalisation introduite réduit cette comparaison à celle des corrélations (cosinus de l'angle $(\bar{\phi}_i(t), \hat{\bar{\vartheta}}_i(t))$) de $\bar{\phi}_i(t)$ avec le vecteur de paramètres $\hat{\bar{\vartheta}}_i(t)$, $i = 1, \ldots, s$. Les paramètres du mode le plus vraisemblablement assorti à la paire de mesures $(\phi(t), \tilde{y}(t))$ (la direction la moins corrélée avec $\bar{\phi}(t)$) seront ajustés en utilisant cette paire.

La forme du critère basé sur l'erreur a posteriori (5.39) est particulièrement intéressante pour être remarquée. En effet, la minimisation de ce critère comprend la minimisation du numérateur $\varepsilon_i(t)$ (erreur *a priori*) mais aussi la maximisation de la quantité $\phi_i(t)^\top L_i(t-1)\phi_i(t)$ qui influerait sur l'évolution éventuelle de $\vartheta_i(t)$ s'il devait se mettre à jour avec $(\phi_i(t), \tilde{y}(t))$. Faisant intervenir à la fois le couple $\{\vartheta_i(t-1), L_i(t-1)\}$ et la paire $(\phi_i(t), \tilde{y}(t))$, l'erreur (5.39) paraît plus appropriée pour décider s'il faut attribuer ou non $(\phi_{\lambda_t}(t), \tilde{y}(t))$ au mode i.

Il paraît évident que, suivant cette procédure, la qualité de la décision sera assez tributaire de la nature des modes en compétition, de leur nombre ainsi que de leur degré de séparabilité. Puisque l'estimation de l'état discret ne tient qu'à une comparaison entre des grandeurs, il n'est pas nécessaire que les vecteurs $\hat{\bar{\vartheta}}_i(t)$, $i = 1, \ldots, s$, soient exactement égaux aux vrais vecteurs $\bar{\vartheta}_i$, $i = 1, \ldots, s$, pour avoir $\hat{\lambda}_t = \lambda_t$. En effet, la détermination (en utilisant l'erreur a posteriori) de l'état sera correcte, si pour tout i, $\exists K < \infty$, $\forall t \geq K$, $\lambda_t = i \implies \left|\eta_j^o(t)\right| > \left|\eta_i^o(t)\right|, \forall j \neq i$. Par ailleurs, tant que l'état discret λ_t est correctement estimé, la convergence de $\hat{\bar{\vartheta}}_i$ sera assurée, pourvu que la collection de régresseurs qui se rapportent à chaque état discret vérifient la condition (5.32). Cependant, pour estimer au mieux l'état discret, une bonne initialisation est déterminante voire indispensable pour l'efficacité de cette méthode.

Finalement, nous résumons dans l'algorithme 5.4 ci-dessous, notre méthode que nous référençons dans la suite comme la méthode EM-MCR, c'est-à-dire, Moindres Carrés Récursifs couplés avec un procédé similaire à celui de l'algorithme Expérance-Maximisation.

Algorithme 5.4 EM-MCR

– Initialisation : Faire $L_i(0) := p_0 I$, avec $p_0 > 1$ et tirer les $\hat{\vartheta}_i(0)$, $i = 1, \ldots, s$ de façon aléatoire. On peut aussi utiliser une heuristique spécifique.
Pour $t = 1, \ldots, \infty$
– Estimer l'état discret :

$$\hat{\lambda}_t = \arg\min_{i=1,\ldots,s} \frac{1}{\|\phi_i(t)\|_2^2} \frac{\eta_i(t)^2}{1 + \left\|\hat{\vartheta}_i(t)\right\|_2^2},$$

où ϑ_i, $\phi_i(t)$ et $\eta_i(t)$ sont définis respectivement comme en (5.34), (5.36) et (5.39).
– Mettre à jour les paramètres du sous-modèle correspondant à $\hat{\lambda}_t$:

$$\left\{\hat{\vartheta}_{\hat{\lambda}_t}(t)^\top, L_{\hat{\lambda}_t}(t)\right\} = \text{MCR}\left\{\hat{\vartheta}_{\hat{\lambda}_t}(t-1)^\top, L_{\hat{\lambda}_t}(t-1), (\phi_{\hat{\lambda}_t}, \tilde{y}(t))\right\}.$$

– Les autres paramètres restent inchangés :

$$\forall\, j \neq \hat{\lambda}_t,\ \left\{\hat{\vartheta}_j(t), L_j(t)\right\} = \left\{\hat{\vartheta}_j(t-1), L_j(t-1)\right\}.$$

FinPour

Remarque 5.3. *Si on suppose que les instants de commutations sont séparés par un certain temps minimum suffisamment grand, alors on n'a plus besoin de lancer plusieurs algorithmes MCR en parallèle comme dans l'algorithme 5.4. On pourrait par exemple employer une procédure d'identification similaire à celle présentée au Chapitre 4 (Section 4.4) pour des modèles d'état. Ainsi, avec un seul algorithme, on procède à l'identification récursive du mode courant dans l'attente d'une éventuelle commutation qu'on peut caractériser à travers des techniques de détection. Une fois qu'une commutation est détectée les paramètres du mode courant sont sauvegardés et la même procédure est relancée pour l'estimation du mode suivant et ainsi de suite.*

En résumé, nous avons présenté et discuté un certain nombre d'algorithmes pour l'estimation de modèles du type SARX. La plupart de ces algorithmes (Sections 5.3, 5.4, 5.5) sont basés sur l'algorithme

général GPCA [129] et sont présentés conjointement avec une technique d'extraction des ordres et du nombre de sous-modèles à partir des données de mesure. Dans le cas de l'algorithme EM-MCR de la section 5.6, nous procédons de manière récursive à la classification des régresseurs en même temps qu'à la mise en jour des paramètres de chaque sous-modèle. Dans le tableau 5.1, nous dressons un récapitulatif des différentes méthodes discutées dans ce chapitre. Une partie de ces méthodes fera l'objet de tests de validation dans le chapitre 6.

Méthode	Hypothèses	Localisation
GPCA-MIMO	s, ρ inconnus	Section 5.3
GPCA-MISO	s, ρ inconnus	Section 5.4
GPCA-MISO(version récursive)	s, ρ connus	Section 5.5
EM-MCR	s, ρ connus	Section 5.6

TABLE 5.1 – Récapitulatif des méthodes discutées. Nous rappelons que $\rho = [n_1, \ldots, n_s]$ représente le vecteur des ordres des sous-modèles.

5.7 Conclusion

Nous avons présenté dans ce chapitre, une extension de la méthode algébrique de Vidal *et al.* à l'identification de systèmes MIMO commutants, décrits par des modèles ARX. Alors que dans la littérature existante, il est généralement fait au moins l'hypothèse que le nombre d'états discrets du système ou l'ordre (commun à tous les sous-modèles) sont connus, nous avons considéré le cas très délicat où ni le nombre de sous-modèles, ni leurs ordres ne sont connus. De plus, les ordres peuvent différer d'un sous-modèle dynamique à un autre.

Pour identifier les paramètres des différents sous-modèles, l'état discret qui est inconnu est élégamment éliminé dans les équations du système grâce à une transformation polynomiale des données de régression. Puis, les ordres et le nombre des sous-modèles sont estimés à travers une condition de rang portant sur les matrices de données entrée-sortie. Une fois ces caractéristiques du système correctement identifiés, les paramètres des différents sous-modèles sont calculés par l'application d'une technique appelée GPCA destinée au partitionnement de données issues de multiples sous-espaces.

Mais la méthode, dans sa forme analytique, ayant une complexité en nombre de polynômes qui dépend exponentiellement des dimensions du système, nous avons proposé dans la section 5.4, une seconde alternative basée sur l'estimation d'un système MISO construit grâce à une projection des données issues du système MIMO original.

Malgré cet allègement de la complexité en nombre de polynômes, il demeure un problème crucial de dimensionalité. Cela nous conduit à envisager une autre approche (cf. Section 5.6), moins performante que la méthode algébrique certes mais qui permet de réduire la dimensionalité. Cette dernière approche se construit autour d'une idée qui consiste à coupler les tâches de classification et d'estimation des paramètres.

6

Validation expérimentale

Sommaire

Dans ce chapitre, nous présentons quelques résultats de simulation numérique et expérimentaux. Une partie des méthodes développées dans les chapitres précédents sont testées, analysées et comparées à travers une série de tests. Ces tests visent notamment à évaluer qualitativement les performances des méthodes proposées. Pour ce faire, nous considérons d'abord quelques exemples de simulation numérique. Puis nous procédons, à partir de données expérimentales, à l'identification du modèle hybride d'un système industriel de montage automatique de composants électroniques sur des circuits imprimés. C'est un système qui est fréquemment utilisé dans la littérature pour la validation d'algorithmes d'identification de modèles hybrides [67], [65], [18], [77]. Enfin, un système hydrographique non-linéaire, consistant en un réseau de bras de rivières [41], est modélisé par un ensemble de sous-modèles affines commutants.

6.1 Exemples de systèmes commutants

6.1.1 Identification en ligne de modèles d'état

Nous considérons l'exemple d'un modèle commutant dont les sous-modèles sont décrits par des représentations d'état. Afin d'illustrer la méthode décrite à la section 4.4 du Chapitre 4, considérons l'exemple numérique suivant qui est constitué de quatre sous-modèles linéaires M_1, M_2, M_3, M_4 d'ordres respectifs 2, 2, 4, 3. Les matrices de ces sous-modèles sont données par

$$M_1 : \begin{cases} A_1 = \begin{bmatrix} 0.5386 & -0.5812 \\ -0.5812 & -0.4745 \end{bmatrix}, & B_1 = \begin{bmatrix} 0.7332 & -1.745 \\ 0 & 0.4392 \end{bmatrix}, \\ C_1 = \begin{bmatrix} -0 & 0.5776 \\ 2.102 & 0 \end{bmatrix}, & D_1 = \begin{bmatrix} 1.432 & 0 \\ 0 & 0 \end{bmatrix}, \end{cases}$$

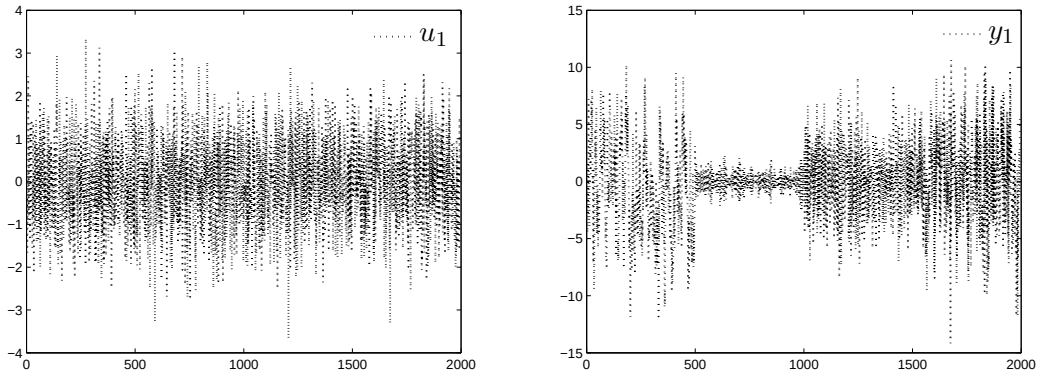
$$M_2 : \begin{cases} A_2 = \begin{bmatrix} 0.08754 & -0.8488 \\ 0.8488 & 0.08754 \end{bmatrix}, & B_2 = \begin{bmatrix} 0 & -0.9219 \\ 0 & 0 \end{bmatrix}, \\ C_2 = \begin{bmatrix} 0.6617 & -0.6294 \\ 0 & 0.01849 \end{bmatrix}, & D_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0.0354 \end{bmatrix}, \end{cases}$$

$$M_3 : \begin{cases} A_3 = \begin{bmatrix} -0.3382 & -0.3114 & 0.598 & -0.1969 \\ -0.465 & -0.5806 & -0.1201 & 0.3968 \\ -0.01017 & -0.3989 & -0.6035 & -0.4613 \\ 0.5263 & -0.3633 & -0.004035 & 0.06349 \end{bmatrix}, & B_3 = \begin{bmatrix} -1.445 & -0.8587 \\ 1.886 & -0.9502 \\ 0.8409 & -0.2533 \\ 0.1275 & 0.9448 \end{bmatrix}, \\ C_3 = \begin{bmatrix} -0.5347 & -0.2918 & -0.02547 & 0 \\ 1.583 & -0.8105 & 0.838 & 0.6313 \end{bmatrix}, & D_3 = \begin{bmatrix} 0 & 0.61 \\ 0.6143 & 0 \end{bmatrix}, \end{cases}$$

$$M_4 : \begin{cases} A_4 = \begin{bmatrix} 0.2397 & -0.522 & -0.4535 \\ -0.522 & -0.06531 & -0.4436 \\ -0.4535 & -0.4436 & 0.3772 \end{bmatrix}, & B_4 = \begin{bmatrix} 0 & 0.5096 \\ 2.143 & 0 \\ -0.8066 & -0.6178 \end{bmatrix}, \\ C_4 = \begin{bmatrix} 0.3859 & 1.495 & 0 \\ 1.202 & 0.4929 & -0.2108 \end{bmatrix}, & D_4 = \begin{bmatrix} 0 & 0.8659 \\ 0.287 & 0.527 \end{bmatrix}. \end{cases}$$

Les instants de commutations qui déclenchent les transitions $M_1 \to M_2$, $M_2 \to M_3$, et $M_3 \to M_4$, sont respectivement 500, 1000, 1500.

Pour générer les données d'identification, nous choisissons le signal d'excitation comme un bruit blanc centré de variance unité. Le bruit de mesure est simulé grâce à l'addition d'un bruit blanc sur la sortie du modèle de façon à réaliser un Rapport Signal sur Bruit (RSB) de 35 dB. Afin d'illustrer les formes de l'entrée et de la sortie, nous montrons à la figure 6.1 la première composante u_1 de l'entrée d'excitation ainsi que la première composante y_1 du vecteur de sortie. Cette figure montre que les changements de sous-modèles sont observables dans l'amplitude de la sortie.



(a) Entrée u_1. (b) Sortie y_1.

FIGURE 6.1 – Première composante du vecteur d'entrée et première composante du vecteur de sortie.

Nous appliquons ensuite l'algorithme d'identification 4.1 du Chapitre 4 avec les paramètres de réglage suivants : $f = 7 > \max_{j=1,\ldots,4}(n_j)$; $\lambda = 0.9$ pour permettre une convergence rapide tout en lissant les paramètres estimés ; $T_o = 0.1$ est supposé être légèrement supérieur à la variance σ_v^2 du bruit de sortie ; le vecteur de poids γ est généré de façon aléatoire puis normalisé de sorte que $\gamma^\top \gamma = 1$.

Avec ces paramètres de réglage, nous effectuons sur le modèle commutant $\{M_1, M_2, M_3, M_4\}$, une simulation de Monte-Carlo de taille 100 avec différentes réalisations du bruit de sortie et de l'entrée d'excitation.

Nous représentons alors sur la figure 6.2, la moyenne sur les 100 simulations, des estimées obtenues pour les ordres ainsi que pour les pôles (modules). Pour la lecture de cette figure, il est important de savoir que le sous-modèle M_1 a deux pôles réels ; le modèle M_2 possède une paire de pôles complexes ; le sous-modèle M_3 possède quant à lui, deux paires de pôles complexes tandis que les trois pôles de M_4

Instant de commutation	Instant de commutation détecté	Retard de détection
500	505	5
1000	1006	6
1500	1503	3

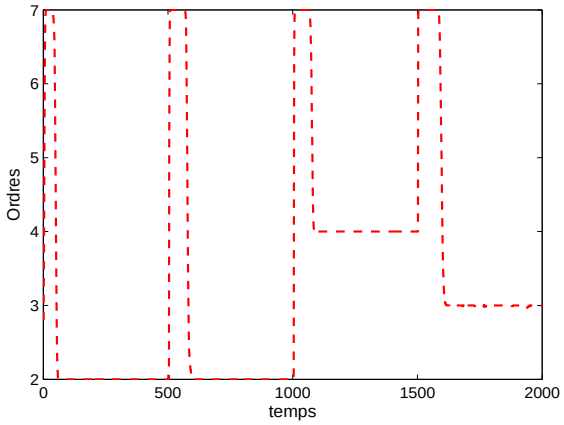
TABLE 6.1 – Moyenne sur 100 simulations des instants de commutation détectés. En fait, ces moyennes peuvent être décimales. Nous en présentons dans ce cas la partie entière.

sont tous réels. C'est pourquoi le nombre de courbes d'amplitude varie entre 1 et 3 selon les intervalles d'activité de chaque sous-modèle. Nous pouvons remarquer que lorsque le système commute d'un sous-modèle à un autre, l'estimée de l'ordre augmente de façon abrupte pour atteindre la valeur maximale f et cela, même quand les deux sous-modèles concernés par la commutation possèdent le même ordre. Comme conséquence d'un changement de mode, ce phénomène peut être attribué à la présence de données mixtes (générées par deux sous-modèles différents) dans la fenêtre d'estimation. Ainsi, c'est comme si au voisinage de la commutation, l'algorithme d'identification tentait d'ajuster à un seul modèle, des données qui ne suivent cependant plus la structure d'un modèle linéaire, d'où l'augmentation de l'ordre. Quant aux instants de commutation, ils semblent être détectés avec un retard relativement faible (de l'ordre de f échantillons environ), ce qui est en concordance avec le résultat de la Proposition 4.3. Cela est illustré par le tableau 6.1. Par contre, après la commutation, il est nécessaire d'accumuler un nombre consistant d'échantillons de données pour pouvoir retrouver l'ordre du sous-modèle suivant.

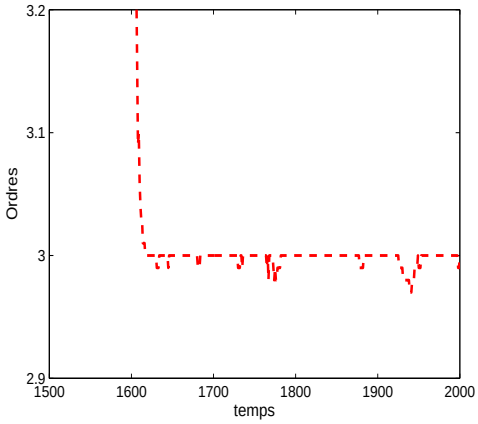
Il est possible d'observer (Sous-figures 6.2-(c) et 6.2-(d)) quelques petits écarts dans la moyenne des pôles sur l'intervalle de temps [1500, 2000]. Cela nous incite à considérer plus attentivement la moyenne de l'ordre donnée par la sous-figure 6.2-(b) sur l'intervalle [1500, 2000]. Il s'avère alors que cette moyenne de l'ordre a quelques fois des valeurs non entières, ce qui suggère que l'ordre du sous-modèle M_3 n'a pas été correctement estimé sur l'ensemble des 100 simulations.

La figure 6.3 montre, à titre illustratif, l'évolution du seuil qui sert à la détection des ordres à partir des quantités h_r définies à la section 3.4. L'estimation de l'ordre est bien sûr très étroitement liée à la valeur de ce seuil.

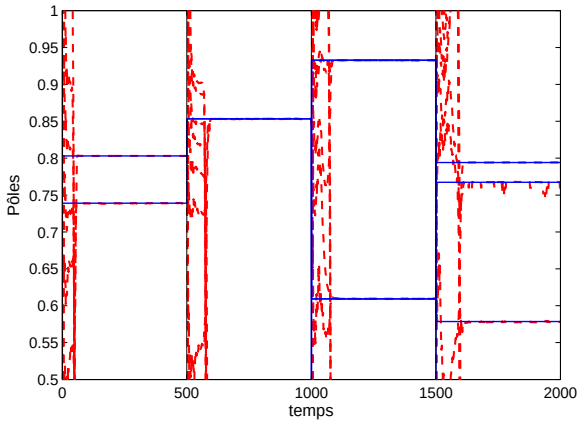
Sur la figure 6.4 sont représentés les quatre premiers paramètres de Markov du modèle estimé en même temps que ceux du système réel. Il s'avère que les paramètres du modèle estimé et ceux du système réel sont bien superposés et que l'algorithme d'identification permet d'obtenir de bons résultats en présence de bruit. Notons encore une fois que l'un des principaux intérêts de cette approche récursive est la possibilité de déterminer ou de reconnaître les différents modes opératoires à mesure que ceux-ci sont visités par le système. Cela constitue un avantage intéressant qui permet d'éviter que certains modes du système ne soient ignorés par le procédé d'identification. En effet, dans le cas des méthodes hors-ligne qui opèrent sur un échantillon fini de données, il peut arriver que tous les états discrets du système n'apparaissent pas intégralement dans les données collectées. De ce fait, ces états discrets ne seront pas visibles par un algorithme d'identification basé sur l'échantillon fini considéré. La méthode de la section 4.4 permet d'éviter ce problème.



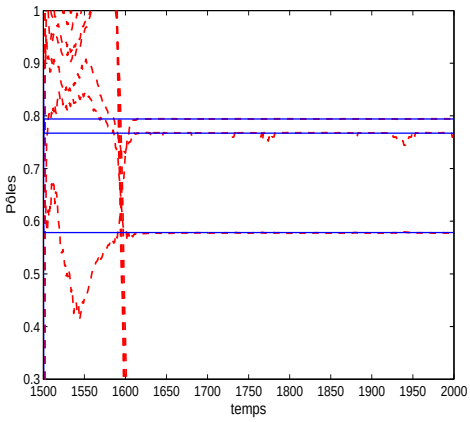
(a) Ordres.



(b) Zoom de (a) entre 1500 et 2000.



(c) Amplitude des pôles.



(d) Zoom de (c) entre 1500 et 2000.

FIGURE 6.2 – Moyenne des estimées (ordres et pôles en pointillés rouge) obtenues par une simulation de Monte-Carlo de taille 100. Les modules des pôles du système réel sont tracés en trait plein bleu.

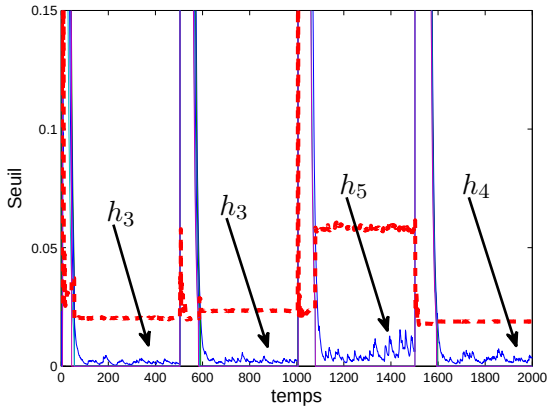


FIGURE 6.3 – Seuil de détection des ordres (pointillés) opérant sur les valeurs des paramètres h_r (trait plein) définis à la section 3.4.

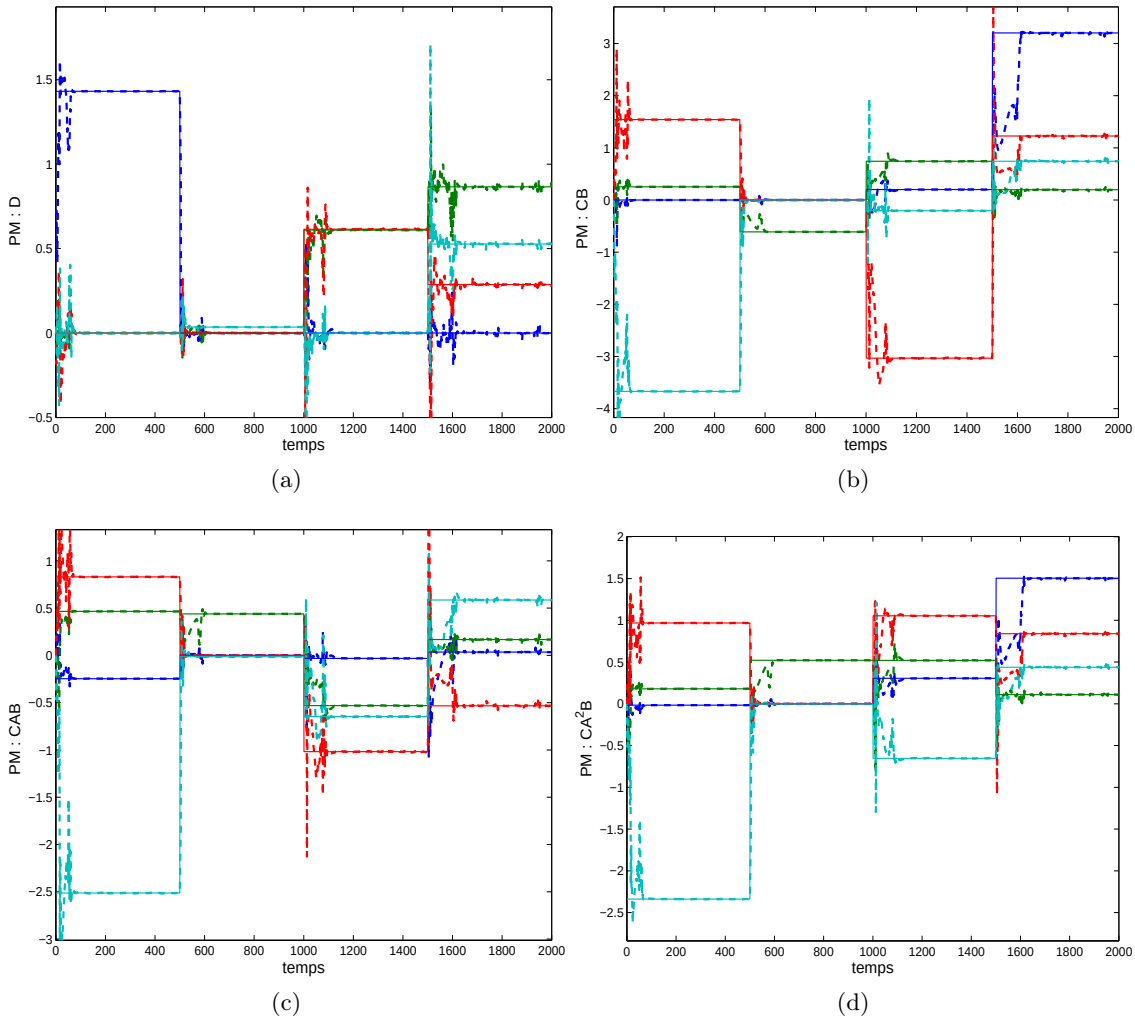


FIGURE 6.4 – Moyennes des quatre premiers Paramètres de Markov estimés sur une simulation de Monte-Carlo de taille 100. (a) : éléments de la matrice D, (b) : éléments de la matrice CB, (c) : éléments de la matrice CAB, (d) : éléments de la matrice CA^2B.

Afin de mieux apprécier les potentialités de notre méthode pour l'identification de modèles commutants, nous l'appliquons maintenant à un système commutant dont les paramètres des différents sous-modèles sont susceptibles de varier lentement au cours du temps. Le système considéré est composé des sous-modèles M_1, M_2 définis ci-dessus dont nous faisons varier maintenant les pôles. Les variations sont créées en multipliant la matrice A du sous-modèle M_1 par $1 - 5.10^{-2}\sqrt{t-\tau_1}$ (pour obtenir une variation rapide) et celle du sous-modèle M_2 par $1 - 10^{-2}\sqrt{t-\tau_2}$ (pour avoir une variation lente), avec $\tau_1 = 0$ et $\tau_2 = 1000$, t faisant ici référence au temps. Les résultats obtenus alors sont représentés sur la figure 6.5 pour les pôles et les ordres, et la figure 6.6 pour les quatre premiers paramètres de Markov. Puisque les variations ne concernent que la matrice A, seules les matrices CAB et CA^2B varient. Par contre les matrices D et CB sont en principe constantes dans notre exemple.
Le sous-modèle M_1 est actif sur [0, 1000] ; ses pôles varient lentement en amplitude en passant par zéro. Pendant ce temps l'ordre décroît de 2 à 1 au voisinage de $t = 230$ et augmente à nouveau dès que la paire de pôles redevient détectable. En réalité, comme le montre la figure 6.6, la matrice de dynamique A s'annule au bruit près autour de $t = 400$. L'estimée de cette matrice est minimale à environ $t = 419$ avec $\left|\hat{A}(t = 419)\right| \simeq 1.97 \times 10^{-4}$, l'ordre estimé étant égal à un dans ce voisinage. Mais à cause du bruit contenu dans les données, le seuillage utilisé ne permet pas de détecter un ordre nul. Cette mauvaise estimation de l'ordre se traduit sur l'ensemble des quatre courbes de la figure 6.6 par quelques pics de valeur sur l'intervalle [0, 1000]. Si nous comparons les figures 6.4 et 6.6, alors nous pouvons faire le constat que la variance des estimées présentées sur la figure 6.6 est plus forte sur l'intervalle [0, 1000] où le taux de variation de paramètres est important. Il existe même un certain biais sur la figure 6.6-(b) entre 250 et 700 environ. Ainsi, les variations de la matrice A perturbent aussi les estimées de D et CB. Cela se justifie par le fait que la procédure d'identification commence par extraire les matrices A (qui varie) et C pour ensuite estimer les matrices B et D.
Le sous-modèle M_2 est actif sur [1000, 2000] avec un taux de variation bien plus lent ; l'ordre est correctement estimé et ne change pas de valeur. Les estimées des paramètres semblent suivre presque parfaitement les vrais paramètres. Certes, notre méthode est théoriquement fondée seulement pour un système dont les paramètres sont constants mais son caractère récursif lui permet, comme le montre cette expérience, de surmonter d'éventuelles variations lentes de paramètres. Cette propriété est bien sûr très importante puisque tout système réel est sujet à des variations de paramètres.

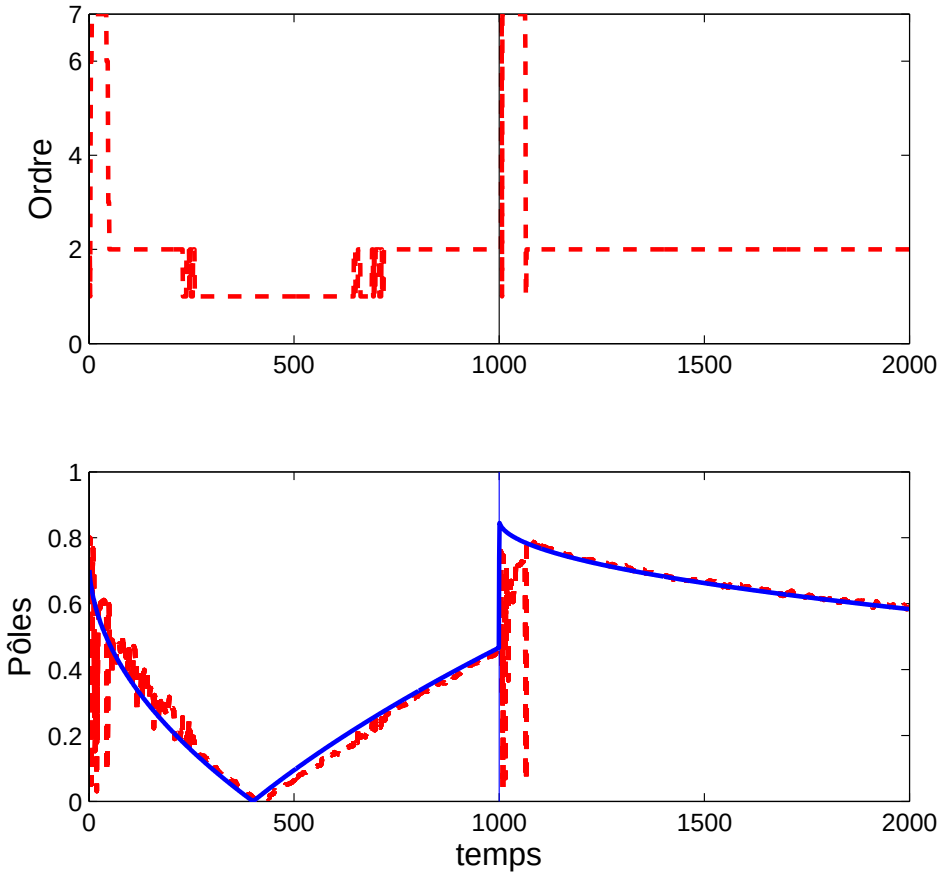


FIGURE 6.5 – Pôles et ordres d'un système commutant dont les paramètres des sous-modèles varient lentement dans le temps.

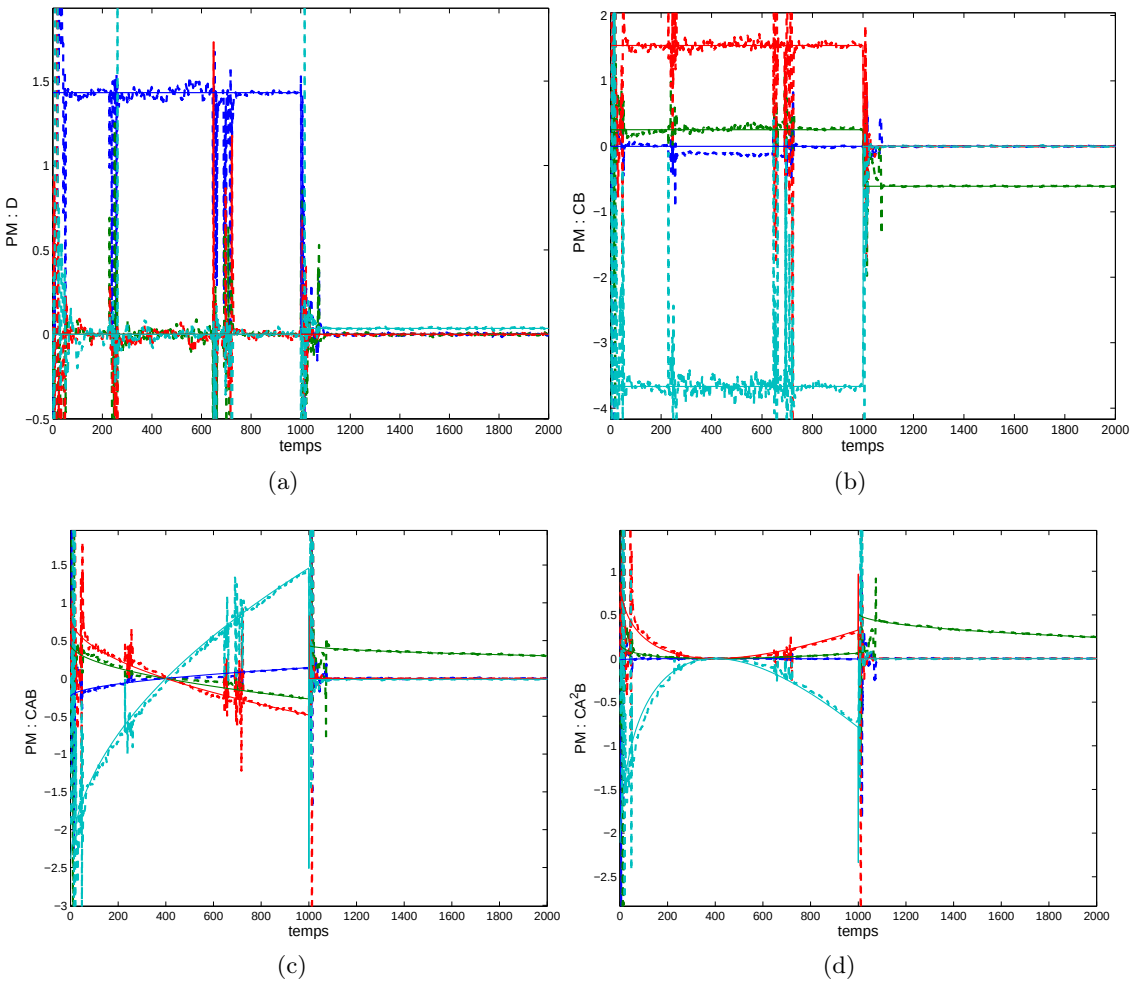


FIGURE 6.6 – Estimées des paramètres de Markov pour un système commutant dont les paramètres des sous-modèles varient lentement dans le temps. (a) : éléments de la matrice D, (b) : éléments de la matrice CB, (c) : éléments de la matrice CAB, (d) : éléments de la matrice CA^2B.

6.1.2 Application de la méthode algébro-géométrique

Nous testons maintenant les performances des deux versions de la méthode algébro-géométrique du chapitre 5 pour l'identification de systèmes à commutations décrits par des modèles SARX. La première méthode consiste en l'extraction des ordres et du nombre de sous-modèles puis en l'application de l'algorithme GPCA [129]. Cette technique implique l'estimation d'un nombre $n_h \geq 2$ (et généralement de l'ordre de $(n_y)^s$, où n_y est le nombre de sorties et s est le nombre de sous-modèles) de polynômes annihilateurs. Dans le cas de la seconde approche, l'espace des polynômes annihilateurs est projeté sur un sous-espace de dimension un. Les données sont alors séparées par sous-modèle générateur grâce à l'estimation d'un seul polynôme annihilateur et les paramètres sont déterminés par la méthode des moindres carrés.

Pour tester ces deux méthodes, considérons un exemple de système commutant constitué de deux sous-modèles d'ordres respectifs 2 et 1, avec $n_u = 1$ entrée et $n_y = 2$ sorties. Les équations du système sont données par

$$y(t) = a^1_{\lambda_t} I_{n_y} y(t-1) + a^2_{\lambda_t} I_{n_y} y(t-2) + b^0_{\lambda_t} u(t) + b^1_{\lambda_t} u(t-1) + b^2_{\lambda_t} u(t-2) + e(t), \tag{6.1}$$

où $a^1_{\lambda_t}$ et $a^2_{\lambda_t}$, avec $\lambda_t \in \{1, 2\}$, sont des coefficients scalaires, $b^0_{\lambda_t}$, $b^1_{\lambda_t}$, $b^2_{\lambda_t}$ sont des vecteurs de dimension $n_y = 2$. Les coefficients $a^2_{\lambda_t}$ et $b^2_{\lambda_t}$ sont nuls pour le sous-modèle d'ordre 1.

Le système est excité en entrée par un bruit blanc Gaussien de moyenne nulle et de variance unité. Bien que la méthode décrite ne requiert aucun temps de séjour minimum, nous ferons, pour des raisons de stabilité, commuter le système périodiquement d'un état discret à un autre tous les 10 échantillons (cf. Figure 6.7). Un bruit blanc additif est rajouté sur la sortie de sorte que le RSB soit de 30 dB.

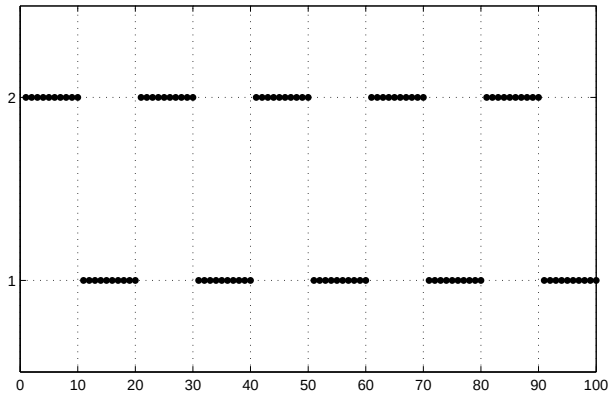


FIGURE 6.7 – Représentation de l'état discret λ_t sur [0, 100].

Les matrices de paramètres des deux sous-modèles considérés à l'équation (6.1) sont explicitement don-

nées par

$$P_1 = \left[\begin{array}{cc|cc|c|c|c} 1.3561 & 0 & 0.6913 & 0 & 0 & 0.3793 & 0.2639 \\ 0 & 1.3561 & 0 & 0.6913 & 1.3001 & 1.8145 & 0.7768 \end{array}\right],$$
$$P_2 = \left[\begin{array}{cc|cc|c|c|c} 0.9485 & 0 & 0 & 0 & 1.7661 & 2.9830 & 0 \\ 0 & 0.9485 & 0 & 0 & 0 & 0.9106 & 0 \end{array}\right], \quad (6.2)$$

qui sont définis par rapport au vecteur de régression

$$\varphi(t) = \left[y(t-1)^\top \;\middle|\; y(t-2)^\top \;\middle|\; u(t) \;\middle|\; u(t-1) \;\middle|\; u(t-2)\right]^\top,$$

de façon à ce que le modèle (6.1) puisse s'écrire $y(t) = P_{\lambda_t}\varphi(t) + e(t)$, où λ_t a la forme représentée à la figure 6.7. Etant donné des mesures entrée-sortie générées par ce système sur une fenêtre temporelle de taille 1500, l'objectif est d'extraire le nombre s de sous-modèles, les ordres n_1 et n_2 de ces sous-modèles ainsi que les matrices de paramètres P_1 et P_2 respectives qui les décrivent. Afin d'illustrer les performances de nos deux algorithmes, nous procédons à une simulation de Monte-Carlo de taille 1000, avec des réalisations différentes de l'entrée et du bruit. Les bornes supérieures sur les ordres et sur le nombre d'états discrets sont respectivement $\bar{n} = 3$ et $\bar{s} = 3$. En prenant alors un seuil $\varepsilon_0 = 0.001$ dans l'algorithme §5.3.2.2 (Chapitre 5, page 154), l'estimation des ordres de tous les deux sous-modèles est réalisée avec 100% de succès. Puisque $\bar{s} = 3$, le vecteur des ordres obtenu est $\hat{\rho} = \begin{bmatrix} 2 & 1 & 0 \end{bmatrix}$. Les moyennes $\hat{P}_1$ et $\hat{P}_2$ des estimées des matrices de paramètres sur l'ensemble des 1000 simulations sont données par

$$\mathrm{Moy}(\hat{P}_1) = \left[\begin{array}{cc|cc|c|c|c} 1.3558 & 0.0043 & 0.6897 & 0.0036 & 0.0056 & 0.3937 & 0.2639 \\ -0.0012 & 1.3558 & -0.0021 & 0.6907 & 1.3031 & 1.8208 & 0.7753 \end{array}\right],$$
$$\mathrm{Moy}(\hat{P}_2) = \left[\begin{array}{cc|cc|c|c|c} 0.9480 & 0.0045 & -0.0005 & 0.0050 & 1.7710 & 2.9869 & 0.0050 \\ -0.0003 & 0.9479 & -0.0001 & -0.0006 & -0.0012 & 0.9081 & -0.0018 \end{array}\right]$$

avec les variances

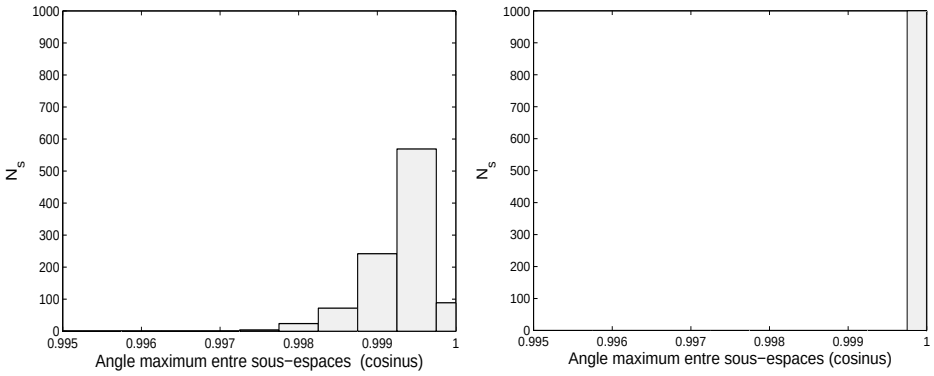
$$\mathrm{Var}(\hat{P}_1) = \left[\begin{array}{cc|cc|c|c|c} 0.0005 & 0.0009 & 0.0014 & 0.0020 & 0.0183 & 0.0246 & 0.0028 \\ 0.0003 & 0.0016 & 0.0007 & 0.0039 & 0.0262 & 0.0245 & 0.0047 \end{array}\right],$$
$$\mathrm{Var}(\hat{P}_2) = \left[\begin{array}{cc|cc|c|c|c} 0.0003 & 0.0015 & 0.0006 & 0.0016 & 0.0061 & 0.0108 & 0.0043 \\ 0.0002 & 0.0006 & 0.0003 & 0.0006 & 0.0021 & 0.0040 & 0.0018 \end{array}\right],$$

où $\mathrm{Var}(\hat{P})$ est une matrice construite avec les variances de chaque élément de la matrice $\hat{P}$.

La figure 6.8 présente un histogramme de l'angle maximum entre l'espace des colonnes de la matrice de paramètres hybrides H et celui de son estimée $\hat{H}$. On peut remarquer que pour l'ensemble des 1000 simulations effectuées, le cosinus de cet angle est plus grand que 0.99, impliquant une correlation forte entre H et $\hat{H}$. Pour la seconde version de la méthode (cf. Section 5.4), ce résultat est bien meilleur puisque H consiste dans ce cas en un seul vecteur h à estimer.

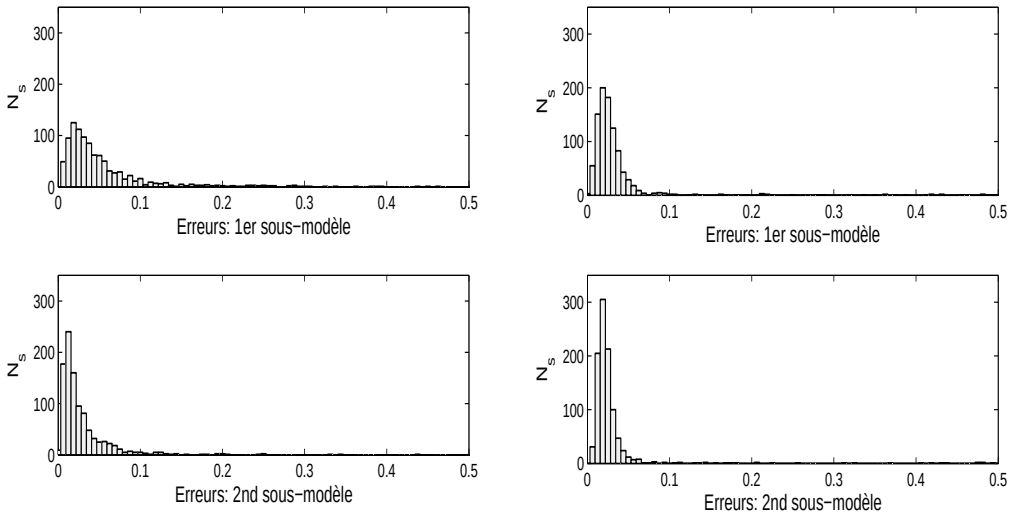
La figure 6.9 présente les erreurs relatives entre les vraies matrices de paramètres P_j, $j = 1, 2$, et leurs

estimées respectives $\hat{P}_j$ obtenues par nos deux algorithmes. Nous pouvons observer que le pourcentage de simulations pour lesquelles les erreurs relatives sont en dessous de 0.05 est d'environ 66% pour le premier sous-modèle et 85% pour le second sous-modèle. Ces pourcentages sont significativement améliorés (86% et 93% respectivement) lorsque nous utilisons l'algorithme GPCA pour la séparation des données par mode puis les moindres carrés (cf. Section 5.4) pour l'estimation des paramètres.



(a) Méthode MIMO GPCA (b) Méthode MISO GPCA

FIGURE 6.8 – Histogrammes sur 1000 simulations du cosinus de l'angle maximum entre les sous-espaces span(H) et span($\hat{H}$). La notation N_s fait référence au nombre de simulations.



(a) Méthode MIMO GPCA (b) Méthode MISO GPCA

FIGURE 6.9 – Histogrammes des erreurs relatives $\left\|P_1 - \hat{P}_1\right\|_2 / \left\|P_1\right\|_2$ et $\left\|P_2 - \hat{P}_2\right\|_2 / \left\|P_2\right\|_2$.

Notons cependant que si nous augmentons considérablement le niveau de bruit dans les données (RSB $<$ 20 dB par exemple), il devient difficile d'extraire les ordres des différents sous-modèles. En fait le réglage du seuil ε_o devient très délicat. Le tableau 6.2 donne le pourcentage de succès (sur 100 simulations) dans la détermination des ordres pour différents niveaux de bruit. Pour $\bar{n} = 3$ par exemple,

nous disons que les ordres sont déterminés avec succès si $\hat{\rho} = \begin{bmatrix} 2 & 1 & 0 \end{bmatrix}$ tandis que $\hat{\rho} = \begin{bmatrix} 2 & 1 & 1 \end{bmatrix}$ par exemple est compté comme un échec bien que les deux ordres soient correctement estimés. Les performances de l'algorithme d'estimation de l'ordre semblent dépendre fortement du seuil ε_o utilisé. Par exemple, dans le tableau 6.2, on peut lire que pour un RSB de 15 dB et un seuil ε_o valant 0.05, on obtient seulement 77% de succès lorsque $(\bar{n}, \bar{s}) = (3, 3)$. Cependant, si nous prenons $\varepsilon_o = 0.01$, nous obtenons un taux de 100% de succès dans les mêmes conditions. En conclusion, il semble que pour améliorer l'estimation des ordres, le seuil ε_o a besoin d'être choisi en fonction de $\bar{n}$, $\bar{s}$, σ_e^2 et de la matrice de données V_ρ, où σ_e^2 est la variance du bruit $e(t)$ dans (6.1). Malheureusement, une telle relation est très compliquée à établir de façon théorique.

RSB, seuil ε_o	$\bar{n} = 3$, $\bar{s} = 3$	$\bar{n} = 4$, $\bar{s} = 3$
10 dB, $\varepsilon_o = 0.05$	98 %	81 %
15 dB, $\varepsilon_o = 0.05$	77 %	100%
20 dB, $\varepsilon_o = 0.01$	100 %	100%
25 dB, $\varepsilon_o = 0.01$	98 %	100%
30 dB, $\varepsilon_o = 0.001$	100 %	99%
35 dB, $\varepsilon_o = 0.001$	100 %	100%

TABLE 6.2 – Performances de la procédure de détermination des ordres en fonction du niveau de bruit.

6.1.3 Application de la méthode de classification-identification

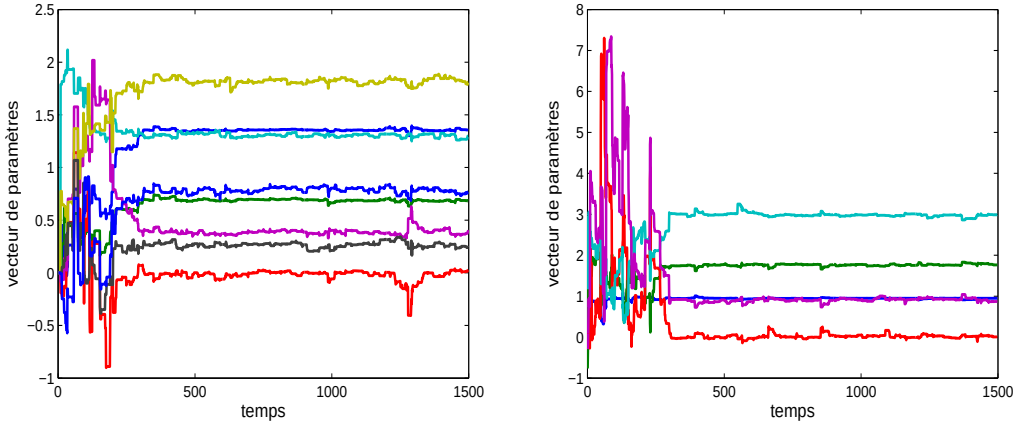
Nous considérons toujours le modèle SARX (6.1) et procédons à son identification en utilisant cette fois la méthode de classification-identification simultanée discutée à la section Section 5.6 du Chapitre 5. Comme précédemment, l'entrée du système est choisie comme un bruit blanc centré de variance unité ; un bruit additif de sortie est choisi également blanc tel que le RSB soit de 30 dB. Par contre, nous supposons que les ordres des deux sous-modèles du modèle SARX (6.1) sont fournis *a priori*.
Puisque la convergence de l'algorithme EM-MCR dépend fortement de son initialisation, celui-ci ne fournit pas nécessairement des résultats satisfaisants chaque fois qu'il est exécuté sur un nombre fini d'observations. Ainsi, avec une initialisation aléatoire, plusieurs tentatives peuvent être nécessaires à l'obtention de bonnes estimées. Pour une tentative réussie, nous obtenons les estimées suivantes pour les matrices P_j définies en (6.2) :

$$\hat{P}_1 = \left[\begin{array}{cc|cc|cc|cc} 1.3534 & 0 & 0.6830 & 0 & 0.0287 & 0.3899 & 0.2418 \\ 0 & 1.3534 & 0 & 0.6830 & 1.2988 & 1.8104 & 0.7712 \end{array}\right],$$

$$\hat{P}_2 = \left[\begin{array}{cc|cc|cc|cc} 0.9490 & 0 & 0 & 0 & 1.7647 & 2.9866 & 0 \\ 0 & 0.9490 & 0 & 0 & 0.0046 & 0.8694 & 0 \end{array}\right].$$

L'évolution de ces paramètres au cours de la procédure de classification-identification récursive est illustrée par la figure 6.10. Cette figure représente en fait l'évolution des éléments des vecteurs (en fait, il s'agit de leurs estimées) $\begin{bmatrix} a_1^1 & a_1^2 & (b_1^0)^\top & (b_1^1)^\top & (b_1^2)^\top \end{bmatrix}^\top \in \mathbb{R}^8$ pour le premier sous-modèle et

$\begin{bmatrix} a_1^1 & (b_1^0)^\top & (b_1^1)^\top \end{bmatrix}^\top \in \mathbb{R}^5$ pour le second sous-modèle, avec a_j^i, b_j^i définis par (6.1). Maintenant, nous ef-



(a) 1er sous-modèle (b) 2ème sous-modèle

FIGURE 6.10 – Evolution des vecteurs de paramètres au cours de leur estimation récursive.

fectuons une simulation de Monte-Carlo de 1000 jeux, chacun des jeux de données portant sur un horizon temporel de 1500 points. Pour chacune des simulations ainsi effectuées, l'initialisation des paramètres à estimer est faite de façon aléatoire. L'histogramme de la figure 6.11 représente alors les erreurs (relatives) entre les vraies matrices de paramètres P_j, $j = 1, 2$, et les matrices de paramètres estimées $\hat{P}_j$, $j = 1, 2$. Cela donne une idée de la performance de la méthode de classification-identification. A en juger par la figure 6.11, les résultats fournis par l'algorithme EM-MCR semblent comparables à ceux obtenus à la section 6.1.2 avec la méthode algébro-géométrique en terme de nombre de simulations aboutissant à une erreur d'estimation raisonnable. Cependant, en regardant ci-dessous la moyenne des estimées obtenues sur les 1000 simulations de Monte-Carlo, il apparaît que la méthode EM-MCR donne nettement de moins bons résultats.

$$\text{Moy}(\hat{P}_1) = \left[\begin{array}{cc|cc|c|c|c} 1.1887, & 0 & 0.3989, & 0 & 0.8279 & 1.5095 & 0.1365 \\ 0, & 1.1887 & 0, & 0.3989 & 0.8445 & 1.5582 & 0.5161 \end{array}\right],$$

$$\text{Moy}(\hat{P}_2) = \left[\begin{array}{cc|cc|c|c|c} 0.8934, & 0 & 0, & 0 & 0.7813 & 1.6706 & 0 \\ 0, & 0.8934 & 0, & 0 & 0.6774 & 1.1338 & 0 \end{array}\right]$$

avec les variances

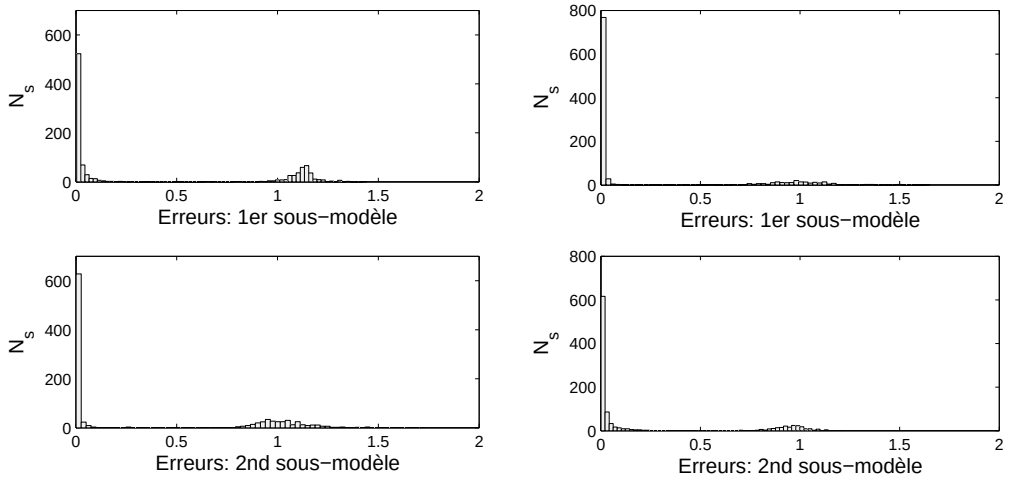
$$\text{Var}(\hat{P}_1) = \left[\begin{array}{cc|cc|c|c|c} 0.0759 & 0 & 0.1351 & 0 & 0.8119 & 1.5716 & 0.3583 \\ 0 & 0.0759 & 0 & 0.1351 & 0.4057 & 0.4497 & 0.4525 \end{array}\right],$$

$$\text{Var}(\hat{P}_2) = \left[\begin{array}{cc|cc|c|c|c} 0.0105 & 0 & 0 & 0 & 1.2009 & 2.0244 & 0 \\ 0 & 0.0105 & 0 & 0 & 0.5139 & 0.1331 & 0 \end{array}\right],$$

où $\text{Var}(\cdot)$ est défini comme précédemment. Ces variances sont assez fortes par rapport à celles des estimées obtenues à la sous-section 6.1.2 par la méthode algébro-géométrique. Ces valeurs numériques

illustrent bien la sensibilité de l'algorithme EM-MCR à l'étape de l'initialisation qui justifie la présence de beaucoup d'estimées aberrantes. Nous pouvons aussi nous reporter à la figure 6.11 pour voir qu'un certain nombre de simulations fournit des estimées bien éloignées des vraies valeurs. Rappelons que notre algorithme EM-MCR n'est pas optimal dans le sens où, en partant de valeurs initiales totalement aléatoires, sa convergence vers un optimum n'est pas garantie (sur un nombre fini d'observations). En outre, plus le nombre de sous-modèles à estimer est élevé, moins les résultats obtenus avec la méthode de EM-MCR sont corrects. Cela est assez logique : le fait de devoir, lors de l'assignation instantanée des données aux sous-modèles, décider entre plusieurs sous-modèles accroît la probabilité de mauvaise classification.

Notons enfin qu'en utilisant un échantillon assez large d'observations pour l'identification, les performances de l'algorithme EM-MCR peuvent être améliorées de façon significative. Cela est illustré sur la figure 6.11-(b). Si chacune des 1000 simulations de Monte-Carlo est réalisée avec 5000 observations au lieu de 1500 comme dans la première expérience (Figure 6.11-(a)), le nombre de mauvaises estimations peut diminuer. De même, en procédant à un certain nombre d'itérations (exécution de l'algorithme plusieurs fois de suite tant qu'une certaine performance n'est pas atteinte) sur l'ensemble des données peut contribuer à réduire significativement les problèmes de convergence. On peut aussi avoir recours à un algorithme spécifique d'initialisation. A cet effet, l'algorithme algébro-géométrique peut être utilisé dans le cas où les dimensions du système considéré ne sont pas très grandes. Ce dernier algorithme n'ayant pas théoriquement une grande aptitude à supporter un niveau de bruit élevé (bien que analytique dans un cas déterministe), il peut servir à initialiser d'autres algorithmes d'identification comme EM-MCR.



(a) Chaque simulation porte sur 1500 points (b) Chaque simulation porte sur 5000 points

FIGURE 6.11 – Méthode de classification-identification : histogramme des erreurs relatives $\left\|P_1 - \hat{P}_1\right\|_2 / \left\|P_1\right\|_2$ et $\left\|P_2 - \hat{P}_2\right\|_2 / \left\|P_2\right\|_2$ entre les paramètres estimés et les paramètres réels sur 1000 simulations avec initialisation aléatoire.

6.2 Modélisation d'une machine de montage de composants sur circuit imprimé

6.2.1 Description de la maquette

Nous procédons dans cette section à une validation de nos méthodes sur un système de montage automatique de composants électroniques sur un circuit imprimé. Ce système est appelé Machine Pickup-and-Place. Une description détaillée du processus peut être trouvée dans [66]. Ici, nous en reproduisons à la figure 6.12 seulement une représentation fonctionnelle. Le système est composé d'une tête de montage qui transporte un certain composant électronique. A chaque cycle de montage, la tête de montage est entraînée jusqu'à la position exacte du circuit imprimé où le composant doit être fixé. Une fois à la position désirée, le composant est poussé délicatement jusqu'à ce qu'il soit solidement en contact avec la surface du circuit, puis il est relâché. Et cette procédure est répétée de manière cyclique chaque fois qu'un composant doit être monté.

Sur la figure 6.12, la tête de montage est représentée par la masse M contrainte à se déplacer seulement selon un axe vertical. Les ressorts c_1 et c_2 simulent une certaine élasticité. Les amortisseurs d_1 et d_2 représentent une friction linéaire tandis que les blocs f_1 et f_2 modélisent une friction sèche. La partie supérieure de la figure 6.12 schématise le mouvement vertical de la tête de montage tandis que la partie inférieure symbolise le support du circuit imprimé. L'entrée du système, représentée par la force $\vec{F}$, est la tension d'alimentation du moteur chargé d'entraîner la tête de montage. Quant à la sortie du système, elle est définie comme étant la position de la tête de montage le long de la verticale. On distingue alors essentiellement deux modes de fonctionnement : le mode *libre* pendant lequel la masse se déplace (de haut en bas) librement le long de l'axe vertical et le mode *contraint* (contact) qui correspond au fonctionnement du système lorsque la tête de montage se trouve en contact avec la surface du banc de montage.

La maquette de montage de composants (Figure 6.12) a été utilisée pour la validation de divers algorithmes d'identification de systèmes hybrides [66], [67], [18], [77]. Nous allons aussi utiliser ce système pour illustrer ci-dessous les potentialités de nos méthodes.[1] Nous disposons pour l'identification de la maquette d'une collection de 60 000 données entrée-sortie $\{u(t), y(t)\}$. A titre d'illustration, la séquence des 10 000 premières données est représentée sur la figure 6.13. Pour la lecture de ce graphique, il est nécessaire de savoir que seule une donnée sur dix est conservée. En réalité, étant donné la déficience fréquentielle du signal d'excitation $u(t)$ donnée par la figure 6.13, il peut être parfois nécessaire, en vue de la procédure d'identification, de l'augmenter d'un signal (de faible amplitude) du type bruit blanc.

6.2.2 Identification d'un modèle d'état

Nous appliquons l'algorithme 4.1 présenté dans la section 4.4 pour l'identification en ligne des sous-modèles linéaires d'un système commutant. Cette méthode permet d'estimer directement des modèles

1. Les données utilisées nous ont été fournies par Dr A. Juloski (Department of Electrical Engineering, Eindhoven University of Technology) avec l'autorisation de la compagnie Assemblon (http ://www.assemblon.com/). Nous tenons à les en remercier.

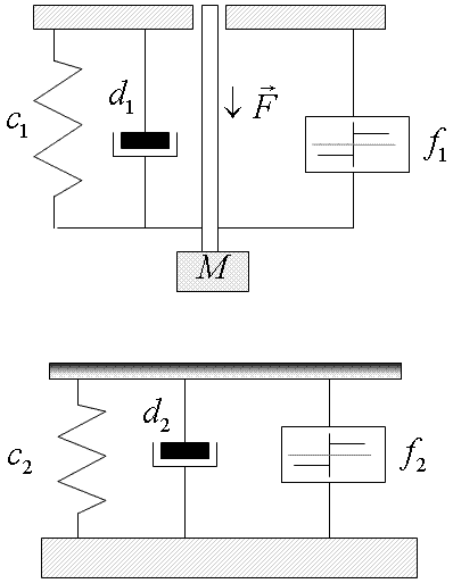


FIGURE 6.12 – Représentation schématique du fonctionnement du système de montage [66].

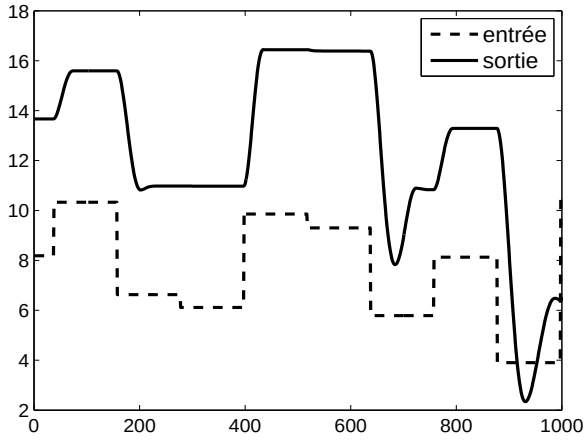


FIGURE 6.13 – Entrée (pointillé) et sortie (trait plein) de la machine Pickup-and-place. Ici, une mise à l'échelle de l'entrée du système est effectuée grâce à une multiplication de celle-ci par un facteur constant de -100.

d'état sous l'hypothèse de l'existence d'un certain temps de séjour suffisamment long entre les occurrences des commutations. Dans un souci de généralité, nous considérons que les sous-modèles sont du type affine, c'est-à-dire que le modèle a la forme

$$\begin{cases} x(t+1) = A_{\lambda_t}x(t) + B_{\lambda_t}u(t) + f_{\lambda_t} \\ \quad y(t) = C_{\lambda_t}x(t) + D_{\lambda_t}u(t) + g_{\lambda_t} \end{cases} \tag{6.3}$$

qui diffère de l'équation (4.1) par l'apparition de constantes additives inconnues f_{λ_t} et g_{λ_t}. C'est donc un modèle affine par morceaux dans lequel le mécanisme de changement de sous-modèle est fonction de $\begin{bmatrix} x(t)^\top & u(t)^\top \end{bmatrix}^\top$. A en juger par les données entrée-sortie représentées à la figure 6.13, il semble qu'il existe un certain temps de séjour entre les changements de modes du système. Cela est conforme à l'intuition qu'on pourrait avoir du fonctionnement de la machine Pickup-and-Place. Le mouvement vertical de la tête de montage pour acheminer une pièce jusqu'au circuit où celle-ci doit être montée requiert une certaine durée. De même, le montage proprement dit de la pièce sur le circuit électronique n'est pas instantané.

Avant de procéder à l'identification du modèle (6.3), il est important de noter que ce dernier modèle est différent de celui (Eq. (4.35)) utilisé dans la section 4.4. Par conséquent, l'application de l'algorithme présenté dans cette dernière section requiert un certain ajustement du modèle (6.3). Dans ce but, considérons un certain intervalle temporel $[\nu,\, \tau-1]$ tel que pour tout $t \in [\nu,\, \tau-1]$, $\lambda_t = i$ pour un certain entier i. Alors, pour tout $t \in [\nu+1,\, \tau-1]$, posons

$$\begin{aligned} \tilde{y}(t) &= y(t) - y(t-1), \\ \tilde{u}(t) &= u(t) - u(t-1), \\ \tilde{x}(t) &= x(t) - x(t-1). \end{aligned}$$

En utilisant ensuite l'équation (6.3), on obtient facilement, pour $t \in [\nu+1,\, \tau-1]$, un nouveau modèle d'état

$$\begin{cases} \tilde{x}(t+1) = A_{\lambda_t}\tilde{x}(t) + B_{\lambda_t}\tilde{u}(t) \\ \quad\;\; \tilde{y}(t) = C_{\lambda_t}\tilde{x}(t) + D_{\lambda_t}\tilde{u}(t) \end{cases} \tag{6.4}$$

qui ne contient plus les constantes f_{λ_t} et g_{λ_t}. Les vecteurs $\tilde{u}(t) \in \mathbb{R}^{n_u}$, $\tilde{x}(t) \in \mathbb{R}^n$, $\tilde{y}(t) \in \mathbb{R}^{n_y}$ sont respectivement l'entrée, l'état et la sortie du modèle linéaire commutant (6.4). Avec un temps de séjour suffisamment long, ce modèle est un modèle linéaire commutant. Etant donné les mesures $\{\tilde{u}(t), \tilde{y}(t)\}$ représentées à la figure 6.14, nous pouvons maintenant appliquer l'algorithme 4.1.

Pour commencer, nous étudions la possibilité d'extraire les ordres des sous-modèles qui composent le système. Dans cet objectif, rappelons que dans l'algorithme 4.1, l'estimation des ordres est basée sur une inspection des paramètres h_r définis à la section 3.4. En utilisant un seuil adéquat $\text{Tresh}(r)$, l'ordre peut être déterminé à travers la détection du premier saut notable dans la série des valeurs h_r, $r = 1, \ldots, r_{\max}$, c'est-à-dire, $n = \min\{r : h_{r+1} < \text{Thres}(r)\}$. Pour des raisons de simplicité, nous choisissons ici un seuil Thres constant. Avec les paramètres de réglage $f = 4$ et $\lambda = 0.98$, notre algorithme d'estimation échoue à rendre une valeur correcte de l'ordre et cela, pour différentes valeurs constantes du seuil Thres. Pour comprendre la raison de ce problème, nous traçons sur la figure 6.15 les paramètres h_r. Dans un souci de comparaison, nous traçons aussi les valeurs singulières σ_r de la matrice de données dont les h_r sont extraits (cf. Section 3.4). Il s'avère que les quantités h_r et σ_r ont une allure très similaire. Les pics de valeurs dans ces derniers paramètres indiquent très vraisemblablement des instants de changements de mode. On peut constater sur le graphique 6.15 que les quantités $h_1, h_2, \cdots$, ont des valeurs très

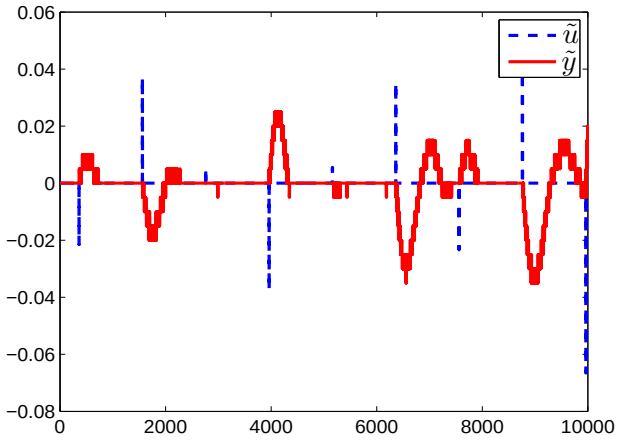


FIGURE 6.14 – Entrée (pointillé) et sortie (trait plein) du modèle.

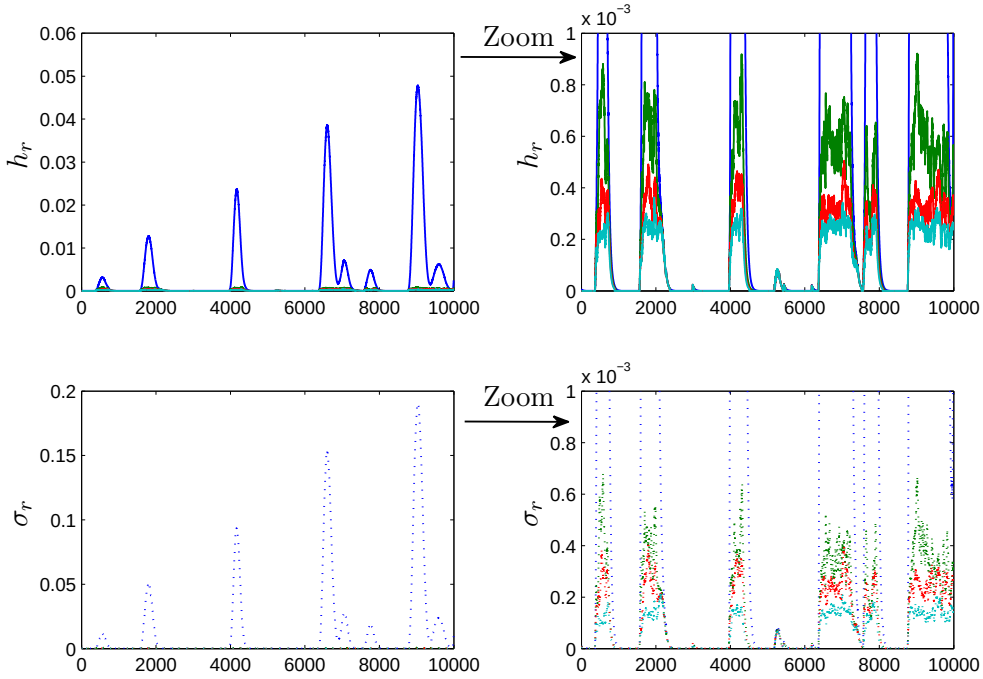


FIGURE 6.15 – Paramètres de seuillage h_r. Plus précisément, la courbe bleue qui est la plus visible ici, représente le paramètre h_1. Les pics de valeurs de cette grandeur indiquent probablement des instants de changement de mode de fonctionnement.

faibles et sont pratiquement confondues excepté dans le voisinage des commutations. Il est donc très ardu de trouver une droite horizontale (seuil constant) qui puisse séparer ces paramètres. Ce problème est vraisemblablement dû à une insuffisance d'excitation des états du système dans les zones où l'entrée est approximativement constante.

Avec les mêmes paramètres de réglage que précédemment, nous fixons désormais l'ordre n à 2. Puis nous réalisons par l'algorithme 4.1, une estimation des paramètres des sous-modèles. Les pôles (en fait il s'agit de leur module) du modèle estimé en ligne sont tracés sur les figures 6.16 et 6.17. Sur ces figures, les intervalles de validité des différents sous-modèles apparaissent d'une manière distincte. Il semble que les transitions entre les sous-modèles s'accompagnent d'un ensemble de phénomènes non-linéaires qu'on pourrait regarder comme étant l'effet des frottements auxquels le système est sujet. Ces perturbations semblent affecter plus le pôle dont l'amplitude est la plus faible. Mais comme le facteur d'oubli λ est très proche de 1, les trajectoires des pôles semblent bien rectilignes après la convergence des paramètres. Notons par ailleurs qu'une petite valeur de λ ne permet pas de réaliser la condition d'excitation persistante car l'échantillon de données expérimentales dont nous disposons ne balaient pas une plage suffisante de fréquences.

Sur la figure 6.16, au moins trois groupes de modes d'allures similaires peuvent être distingués dans la séquence des 10 000 points de mesure utilisés. Une observation attentive de ces résultats laisse penser qu'en raison probablement des éléments élastiques (ressort, amortisseur) qui sont présents dans le système, deux modes correspondant à une même position de la tête de montage (cf. Schéma 6.12) ne sont pas parfaitement identiques. Par exemple, les modes de fonctionnement visités par le système sur les intervalles de temps [1] [3000, 4000] et [5300, 6400] semblent être similaires. On dirait que le système ne revient pas exactement dans une position qu'il a initialement occupée. Cela semble cohérent avec notre intuition de la physique du système. Par exemple, le mode de contact pendant lequel la tête de montage M (cf. Schéma 6.12) est en contact avec le banc de montage, pourrait changer légèrement du fait des éléments élastiques selon l'effort d'appui auquel il est soumis. C'est un phénomène important que notre méthode d'identification en ligne a permis de mettre en évidence contrairement aux méthodes comparées dans [65] qui opèrent hors ligne.

Si nous essayons maintenant de reconstruire la sortie $\tilde{y}(t)$ du modèle (6.4), nous obtenons les résultats de la figure 6.18. Cependant, il faut remarquer qu'il est particulièrement délicat de reconstruire la sortie d'un modèle à commutations pour au moins trois raisons.

- l'état continu $x(t)$ n'est pas connu (de même que l'état initial $x(0)$),
- l'état discret est inconnu et difficile à extraire de façon exacte,
- les bases des matrices estimées pour les sous-modèles individuels ne sont pas nécessairement cohérentes entre elles (cf. Chapitre 4).

Même si les matrices des sous-modèles considérés séparément étaient cohérentes en termes de bases d'état, le fait que la séquence d'états discrets est inconnu rend complexe la procédure de simulation de la sortie. De plus, dans un contexte d'estimation récursive, seules les valeurs de paramètres obtenues après la convergence de l'algorithme d'identification sont valides. Cela signifie que les valeurs intermédiaires atteintes avant la convergence ne sont pas significatives. Ainsi, la sortie simulée (en ligne) $\widehat{\tilde{y}}$ donnée par la figure 6.18 étant basée sur ces valeurs intermédiaires, présente seulement une similarité d'allure avec la sortie mesurée $\tilde{y}$ mais une différence notable dans les valeurs.

1. Ces intervalles de temps sont donnés approximativement par lecture de la figure 6.16.

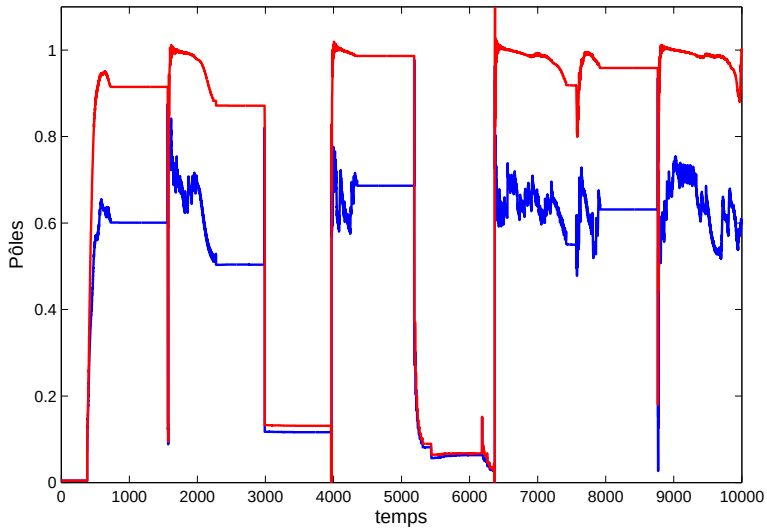


FIGURE 6.16 – Module des deux pôles estimés en ligne.

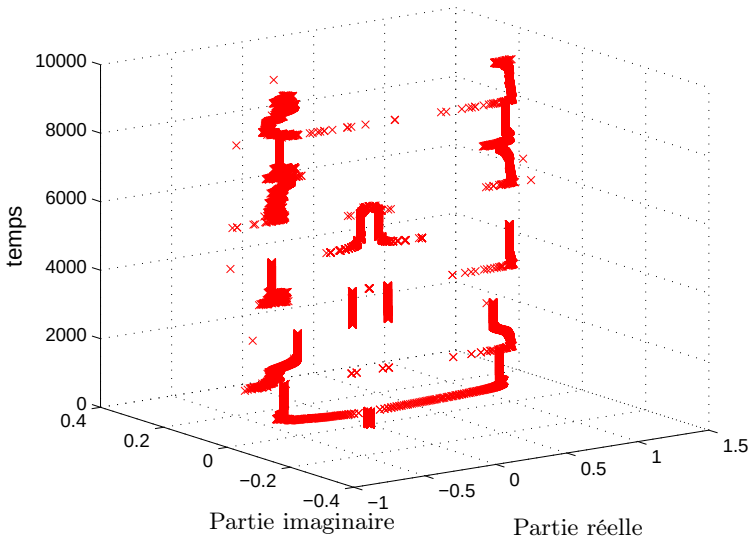


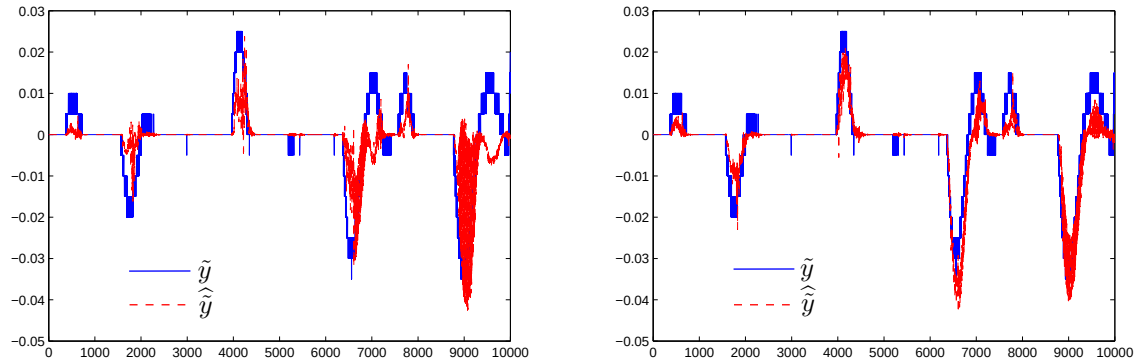
FIGURE 6.17 – Représentation en dimension 3 de l'évolution des deux pôles estimés en ligne. Les pôles sont matérialisés par des croix dans un plan complexe horizontal se mouvant le long de l'axe vertical des indices de temps.

Nous utilisons comme au Chapitre 1, le critère [18]

$$\text{FIT} = \left(1 - \frac{\|\hat{y} - y\|}{\|y - \bar{y}\|}\right) \times 100\%$$

pour mesurer la similarité entre la sortie $y = \begin{bmatrix} y(1) & \cdots & y(N) \end{bmatrix}^\top \in \mathbb{R}^N$ mesurée sur le système et

la sortie $\hat{y} = \begin{bmatrix}\hat{y}(1) & \cdots & \hat{y}(N)\end{bmatrix}^\top \in \mathbb{R}^N$ reconstruite à partir du modèle. Dans cette formule, $\bar{y}$ est la moyenne des mesures $y(t)$, $t = 1, \ldots, N$, où N est le nombre de points de mesure disponibles. Ainsi, dans le cas des courbes de la figure 6.18, nous trouvons FIT= 23% et FIT= 53% en prenant respectivement les ordres $n = 2$ et $n = 3$. De cette étude, nous pouvons conclure que les modèles d'état commutants sont



(a) $n = 2$, FIT = 23%

(b) $n = 3$, FIT = 53%

FIGURE 6.18 – Sortie différentielle $\tilde{y}$ (trait plein bleu) sortie différentielle reconstruite $\widehat{\tilde{y}}$ (pointillé rouge) avec des ordres $n = 2$ et $n = 3$.

peu pratiques pour simuler la sortie à partir d'un modèle estimé. C'est pourquoi, nous nous focalisons dans la suite de cette section sur l'estimation de modèles entrée-sortie commutants pour le système Pickup-and-Place.

6.2.3 Application de la méthode algébrique

L'algorithme algébro-géométrique (ou algorithme GPCA) de R. Vidal a déjà été testé dans [77] et [65] sur la machine Pickup-and-Place et les résultats obtenus ont été comparés avec d'autres méthodes d'identification de systèmes hybrides. En raison de l'extension de cette méthode que nous avons présentée au Chapitre 5 pour l'identification de systèmes MIMO SARX, nous vérifions ici que cet algorithme (Section 5.3) permet d'extraire les paramètres d'un modèle SARX du type (5.1) pour représenter le comportement entrée-sortie de la machine Pickup-and-Place. De cette manière, nous pourrons, bien que la machine Pickup-and-Place ne soit qu'un exemple de système SISO, comparer la technique GPCA avec l'algorithme EM-MCR de la section 5.6.

Le nombre de sous-modèles est fixé à 2 et les valeurs $n_1 = n_2 = 2$ sont assignées aux ordres des deux sous-modèles. Sur un échantillon de 40 000 mesures disponibles, nous utilisons 3000 données pour l'estimation des paramètres et les 37 000 autres points pour la validation du modèle estimé. Le résultat présenté à la figure 6.19 démontre une correspondance presque parfaite entre la sortie réelle y et la sortie estimée. Il convient de noter qu'à moins de disposer de l'état discret ou du mécanisme de commutation, il n'est pas possible de reconstruire la sortie d'un modèle commutant. De ce fait, pour procéder à la simulation de la sortie du modèle estimé à partir de l'entrée et des paramètres, nous devons d'abord

estimer l'état discret en utilisant la sortie mesurée sur le système réel.

Tandis que la sortie $\hat{y}$ reconstruite à partir des paramètres estimés suit parfaitement la sortie réelle du système, l'estimée de l'état discret comme illustré sur la figure 6.20 ne reflète pas la réalité physique de la machine Pickup-and-Place. Car on peut lire sur la courbe de la figure 6.16 qu'il existe un certain délai minimum entre les changements de modes, ce qui n'est pas perceptible sur la figure 6.20. Cela signifie que nous pouvons, avec l'algorithme GPCA, reproduire la sortie du système à travers l'estimation d'un modèle du type SARX sans que les modes estimés ne correspondent vraiment aux modes résultant d'une analyse du fonctionnement physique du processus. En d'autres, les modes estimés par GPCA n'ont pas nécessairement une interprétation significative dans un sens physique.

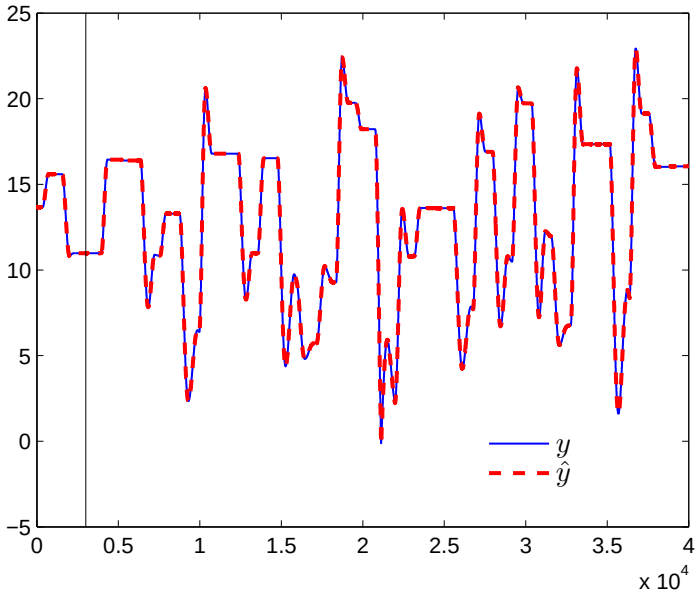


FIGURE 6.19 – Sortie réelle y et sortie reconstruite $\hat{y}$ par la méthode algébrique : FIT = 99%. La ligne verticale sépare les données ayant servi à l'estimation (à gauche) des données de validation (à droite).

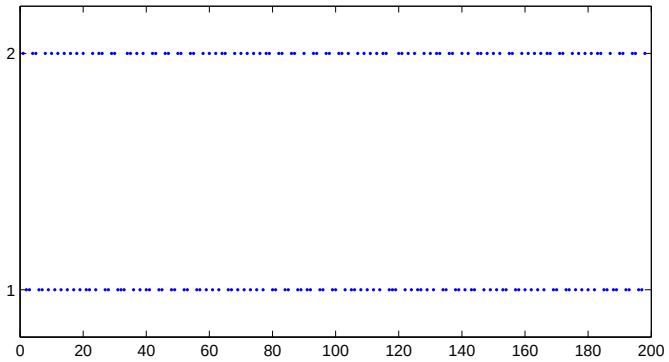
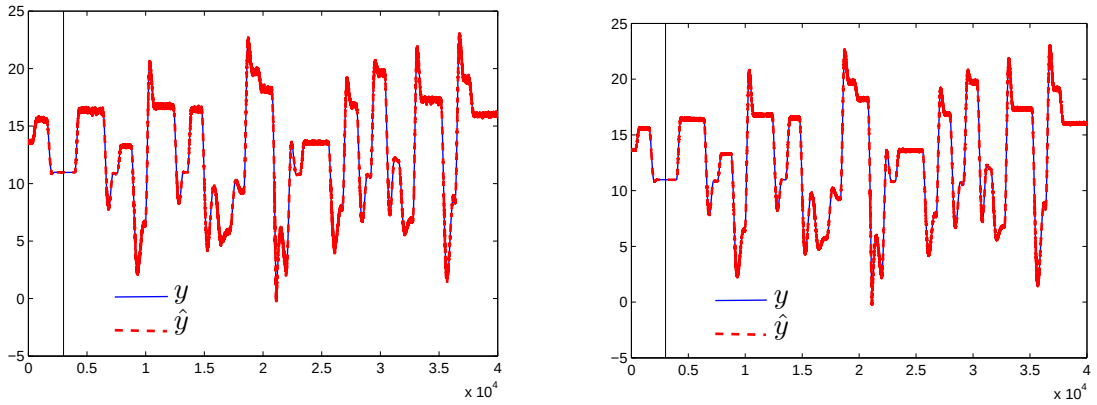


FIGURE 6.20 – Etat discret estimé sur [0, 200].

6.2.4 Application de la méthode EM-MCR

Considérons maintenant la méthode EM-MCR (cf. Section 5.6) pour l'identification de la machine Pickup-and-Place. Pour cela, nous supposons que l'ordre est le même pour tous les sous-modèles et nous supposons successivement que le nombre de sous-modèles vaut $s = 2$ et $s = 3$. Les sorties réelle et reconstruite sont alors fournies par la figure 6.21. Il apparaît que ces deux sorties coincident plus ou moins parfaitement dans tous les deux cas. Mais le cas $s = 3$ semble avoir une performance meilleure par rapport à celui où $s = 2$. Ainsi, il semble que plus nous augmentons le nombre de sous-modèles, plus l'erreur d'approximation est petite. Concernant l'estimation de l'état discret, il convient de préciser que la remarque faite dans la sous-section précédente (cf. Figure 6.20) sur la méthode GPCA est aussi valide dans le cas de la méthode EM-MCR.



(a) 2 sous-modèles : FIT = 96%.

(b) 3 sous-modèles : FIT = 98%.

FIGURE 6.21 – Sortie réelle y et sortie reconstruite $\hat{y}$ par la méthode EM-MCR. La ligne verticale sépare les données ayant servi à l'estimation (à gauche) des données de validation (à droite).

Afin de confirmer les résultats de la sous-section 6.2.2, nous estimons de manière récursive un seul sous-modèle affine pour la machine Pickup-and-Place. Il en résulte la courbe de la figure 6.22 représentant l'évolution du vecteur de paramètres identifiés. Malgré la présence de quelques non-linéarités, les instants de changement de modes (matérialisés par des traits verticaux) semblent détectables sur la courbe. De plus, conformément à notre discussion de la sous-section 6.2.2, il existe plusieurs groupes de modes qui sont similaires sans être exactement identiques. Nous pouvons remarquer une fois encore que la méthode GPCA de la sous-section 6.2.3 a de meilleures performances que la méthode EM-MCR. En effet, la première méthode permet d'obtenir un FIT=99% seulement avec deux sous-modèles tandis que la seconde méthode nécessite un nombre de sous-modèles d'au moins trois pour atteindre un FIT de 98%. De plus, la méthode GPCA est déterministe tandis que la technique EM-MCR donne des résultats variables en fonction de l'initialisation que nous réalisons de façon aléatoire. Mais lorsque les dimensions du système considéré deviennent importantes, l'algorithme GPCA n'est plus applicable car la dimension $M_s(K)$ des coefficients des polynômes de transformation (cf. Chapitre 5) augmente alors considérablement. Dans ces types de situations, la méthode EM-MCR représente une alternative intéressante.

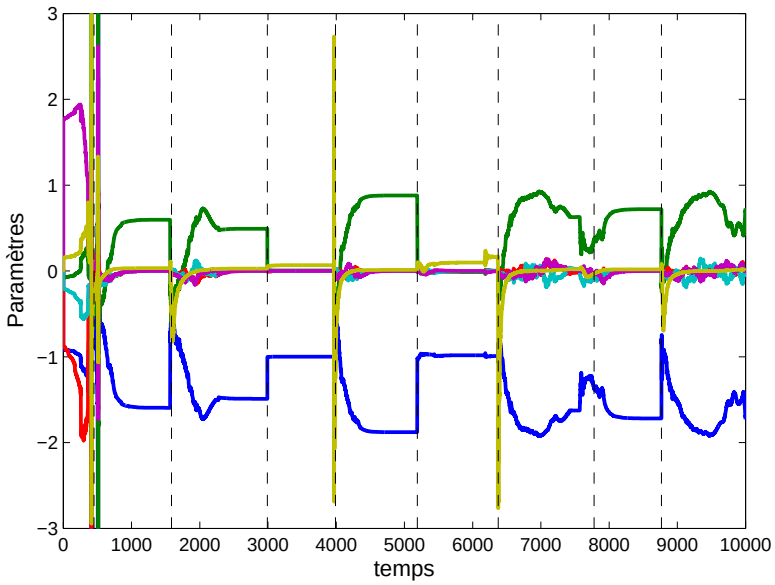


FIGURE 6.22 – Estimation récursive d'un seul sous-modèle affine sur les premiers 10000 points de l'échantillon de données, avec un facteur d'oubli $\lambda = 0.99$. Chaque couleur représente l'évolution d'un élément du vecteur de paramètres identifié.

En conclusion, la méthode EM-MCR permet d'identifier un modèle SARX, simple et précis du système Pickup-and-Place. De même, la méthode d'identification en ligne de modèles d'état commutants (cf. Section 4.4) montre une certaine potentialité à pouvoir capturer les dynamiques du même système. Cependant, un problème général lié à la nature des modèles d'états à commutations est qu'il est fastidieux d'en déduire rigoureusement la sortie (cf. problèmes mentionnés ci-dessus). Ainsi, nous ne pouvons pas comparer dans un sens strict nos méthodes d'estimation de modèles d'état aux techniques entrée-sortie existant dans la littérature. Par contre la méthode EM-MCR est comparable à l'algorithme GPCA de R. Vidal que nous avons déjà étudié ci-dessus. Ces deux méthodes s'appliquent dans le cas général d'un modèle commutant du type SARX, indépendamment du mécanisme de commutation. En comparaison avec certaines méthodes d'identification de modèles plus particuliers comme les PWARX (cf. [65]), il semble que les méthodes telles que EM-MCR et GPCA pourraient ne pas permettre une bonne partition des données de régression (pour un modèle PWARX) contrairement aux méthodes de Ferrari-Trecate *et al.* [46] et de Bemporad *et al.* [18]. Cela s'explique par le fait que l'état discret λ_t est estimé dans EM-MCR et GPCA comme l'indice j du sous-modèle qui minimise l'erreur de prédiction $(y(t) - \theta_j^\top \varphi(t))^2$, où θ_j est le vecteur de paramètres associé au mode j. Dans le cas des autres méthodes qui sont plus spécifiques aux modèles PWARX, la prise en compte dans l'algorithme de la forme de partition recherchée permet d'obtenir effectivement une partition linéaire proche de celle d'un modèle affine par morceaux comme c'est le cas du système Pickup-and-Place.

6.3 Modélisation de systèmes hydrauliques à surface libre

6.3.1 Description du système

Dans cette section, nous nous intéressons à la modélisation de biefs de rivières ou de canaux d'irrigation à surface libre. Cela nous permettra de démontrer qu'un système non linéaire continu peut être représenté par un modèle commutant entre plusieurs sous-modèles affines. Nous considérons deux systèmes dynamiques [42], [41] :

- un système SISO consistant en un bras de rivière de section circulaire avec un rayon R, une longueur X et une inclinaison α par rapport au niveau horizontal. Des schémas d'explication de ce système sont donnés par les figures 6.23 et 6.24.
- un système MIMO représenté à la figure 6.25 et consistant en un réseau de bras de rivières. Dans ce dernier système, la section des différents bras est trapézoïdale avec des paramètres X, b, β, α dont les valeurs respectives sont fournies dans le tableau 6.3.

Les entrées et les sorties de ces deux systèmes (cf. Figures 6.24 et 6.25) sont définies respectivement comme le débit volumique d'eau en amont et en aval du canal ou de l'ensemble de canaux, exprimés tous en $\mathrm{m}^3\,\mathrm{s}^{-1}$.

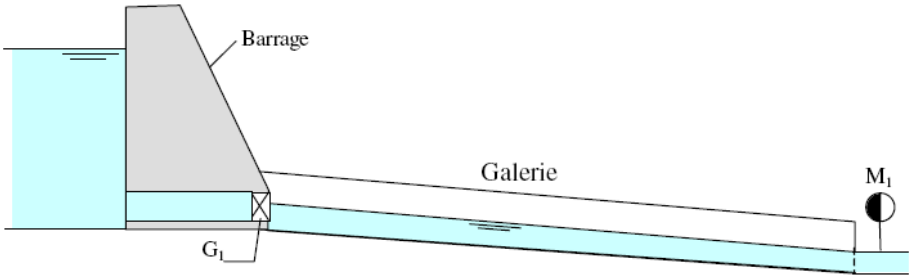


FIGURE 6.23 – Représentation schématique du barrage et de la galerie.

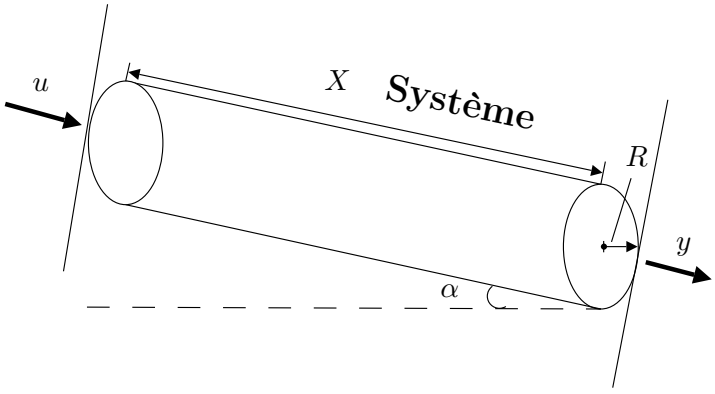


FIGURE 6.24 – Caractéristiques géométriques de la galerie.

Nous commençons ici par donner une intuition physique du fonctionnement du système. Pour cela, désignons par $q(x,t)$ le champ de débit volumique dans le canal, où x est une coordonnée spatiale

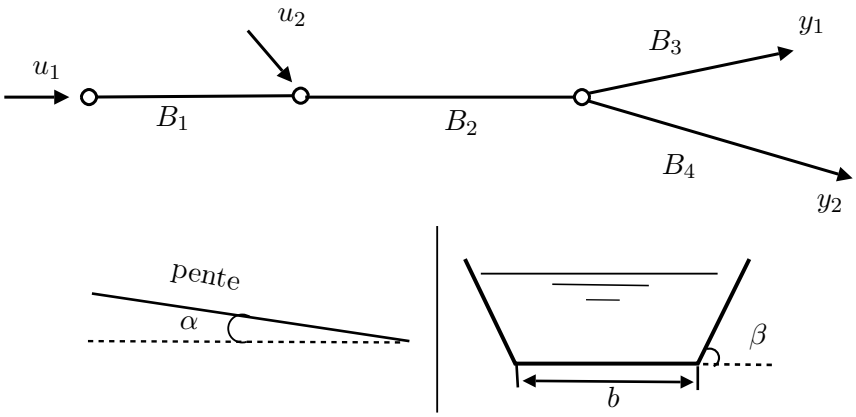


FIGURE 6.25 – Caractéristiques géométriques de la galerie définissant le système MIMO. La sous-figure supérieure indique les tronçons B_1, B_2, B_3 et B_4 du réseau considéré. La sous-figure inférieure illustre la forme trapezoïdale des sections des différents tronçons (cf. Tableau 6.3 pour les caractéristiques géométriques des bras).

unidimensionnelle mesurée selon l'axe du canal et t fait référence au temps. Ce champ de débit (flux d'eau) satisfait en tout point intérieur au niveau d'eau, l'équation aux dérivées partielles suivante, dite equation (simplifiée) de Saint Venant [30]

$$\frac{\partial q(x,t)}{\partial t} + \delta_c(q,x,t)\frac{\partial q(x,t)}{\partial x} - \delta_d(q,x,t)\frac{\partial^2 q(x,t)}{\partial x^2} = 0, \tag{6.5}$$

où $\delta_c(q,x,t)$ en $\mathrm{m\,s^{-1}}$ et $\delta_d(q,x,t)$ en $\mathrm{m^2\,s^{-1}}$ désignent respectivement les coefficients de célérité et de diffusion. C'est une équation du type diffusif semblable par exemple à celle de la propagation de chaleur. Selon cette équation, le champ de débit est soit constant, soit variable à la fois en fonction du temps et de l'espace. Mais en un point fixe de position x_o, le débit q ne dépend que du temps, c'est-à-dire $q(x,t)$ peut être noté simplement $q(x_o,t)$. Ainsi, les débits aux deux extrémités du canal représentent deux grandeurs évoluant seulement en fonction du temps. Notre étude s'intéresse à l'expression dynamique du débit à l'extrémité d'entrée notée $u(t)$ en fonction de celui à l'extrémité de sortie notée $y(t)$. Pour ce faire, nous avons besoin de disposer d'une séquence significative de mesures $\{u(t), y(t)\}_{t=1}^{N}$. En raison de l'étendue des systèmes hydrauliques, de telles données sont souvent difficiles à collecter sur un système réel essentiellement pour des raisons d'instrumentation. Cependant, il existe des outils numériques qui permettent de reproduire le comportement des canaux d'irrigation. L'un de ces outils est le logiciel SIC (Simulation de Canaux d'Irrigation) développé par le CEMAGREF [1] qui permet de résoudre numériquement l'équation (6.5) de Saint Venant avec une relative bonne précision. Il s'agit d'un logiciel de simulation hydraulique qui est très utilisé par les gestionnaires de réseaux hydrographiques pour le calcul des écoulements dans des canaux d'irrigation et des rivières. Les résultats obtenus par une telle résolution numérique de l'équation de Saint Venant sont connus dans la communauté des hydrauliciens comme étant très proches de ceux qu'on pourrait obtenir sur un système réel.

1. SIC user guide and and theoretical concepts, 1992. http ://canari.montpellier.cemagref.fr/

Etant donné les caractéristiques géométriques (forme, dimensions, etc.) du canal et une entrée d'excitation $u(t)$, le logiciel SIC permet de produire une sortie $y(t)$ correspondant au débit en aval du canal défini. En faisant usage de ce logiciel, nous générons, pour chacune des deux configurations (SISO et MIMO), une séquence d'observations entrée-sortie constituée de 6000 et 2500 points de mesure. Les paramètres du système SISO sont donnés par $X = 946\,\text{m}$, $R = 0.9\,\text{m}$ et $\alpha = 0.26\%$ (Figure 6.24) tandis que ceux du système MIMO sont définis dans le tableau 6.3 (cf. Figure 6.25).

Tronçon	X(m)	b(m)	β	α
B_1	500	5	0.9	4×10^{-4}
B_2	1000	5	0.9	4×10^{-4}
B_3	500	2	0.95	5×10^{-4}
B_4	1500	4	0.9	5×10^{-4}

TABLE 6.3 – Caractéristiques géométriques du système MIMO.

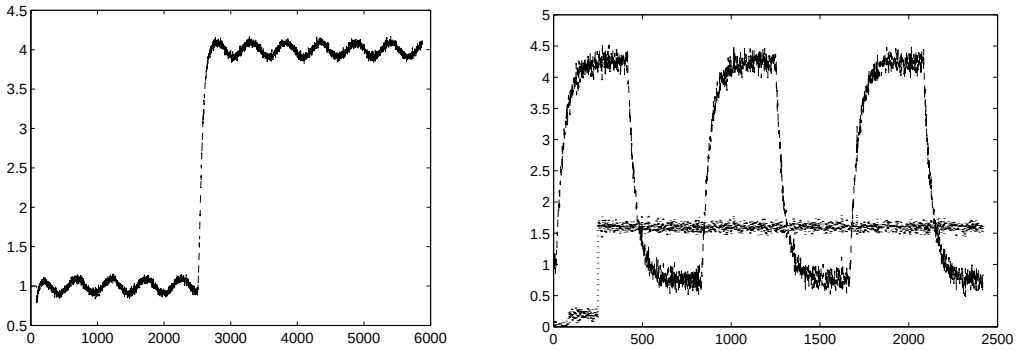
6.3.2 Identification d'un modèle de la galerie

Les deux systèmes hydrauliques décrits dans la sous-section précédente ont clairement un comportement non-linéaire et continu. Mais, pour procéder à leur identification, nous les traitons comme des systèmes hybrides commutant entre un certain nombre de sous-systèmes linéaires. Cela s'inspire directement des travaux de E. Duviella [41] dans lesquels une approche multi-modèles d'inspiration physique a été présentée pour la modélisation de ces types de systèmes. Nous employons alors l'algorithme de classification-identification EM-MCR (cf. Section 5.6) pour identifier un modèle entrée-sortie. [1]
Sur la figure 6.26, nous représentons les vecteurs d'entrées des deux systèmes. Ces signaux sont essentiellement situés dans la plage de fréquences d'exploitation du système hydraulique. Notons qu'une caractéristique importante des systèmes de diffusion comme les systèmes hydrauliques considérés, est la présence de retards pouvant varier en fonction du débit. Cette remarque confirme l'intuition selon laquelle une variation de l'entrée n'est pas instantanément reflétée sur la sortie puisqu'elle doit se propager le long du canal avant d'avoir une répercussion perceptible sur la sortie (cf. [42]). Cependant, nous faisons abstraction de ces retards dans notre approche de modélisation. Ayant choisi l'ordre du modèle égal à 2, nous appliquons l'algorithme EM-MCR avec une initialisation aléatoire et un facteur d'oubli de 0.99. Dans chaque configuration de système, une moitié des données est utilisée pour l'identification des paramètres et l'autre moitié pour la validation du modèle. Nous obtenons alors les résultats fournis par les figures 6.27 et 6.28 pour les systèmes SISO et MIMO respectivement. Il s'agit des sorties réelles (en bleu) superposées aux sorties reconstruites (en rouge) pour différentes valeurs (entre 1 et 4) du nombre s de sous-modèles. Nous pouvons constater que la précision du modèle estimé semble augmenter avec le nombre de sous-modèles. En revanche, lorsque le nombre de sous-modèles considérés est assez important, la convergence de l'algorithme d'identification est plus problématique. Il semble exister un certain seuil de saturation au-delà duquel, une augmentation du nombre de sous-modèles ne permet plus d'améliorer

1. Nous aurions pu utiliser aussi la méthode algébrique. Mais cette méthode ayant déjà été validée sur diverses applications par R. Vidal, il semble plus intéressant de se focaliser ici sur la validation notre algorithme EM-MCR.

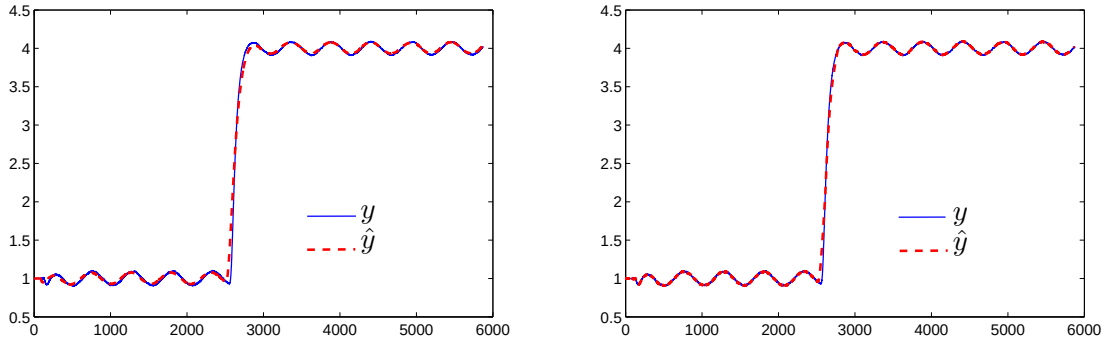
la précision du modèle global. Par exemple, dans le cas de la figure 6.28, il n'y a pratiquement pas de différence entre les courbes obtenues avec $s = 3$ et $s = 4$ puisqu'on a FIT=97% dans les deux cas. Nous pouvons donc en déduire que trois sous-modèles suffisent à bien représenter les dynamiques des deux systèmes considérés.

Nous rappelons que dans notre démarche de modélisation de systèmes hydrauliques, les retards naturels du système n'ont pas été pris en compte. Malgré cette approximation, les sorties reconstruites par identification sont très proches des sorties réelles et cela, pour des entrées non constantes comme le montre la figure 6.26. Les résultats présentés sont satisfaisants dans l'ensemble et ainsi démontrent les potentialités expérimentales de notre méthode.



(a) entrée du modèle SISO. (b) entrées du modèle MIMO.

FIGURE 6.26 – Signaux d'entrée utilisés pour la génération des données.



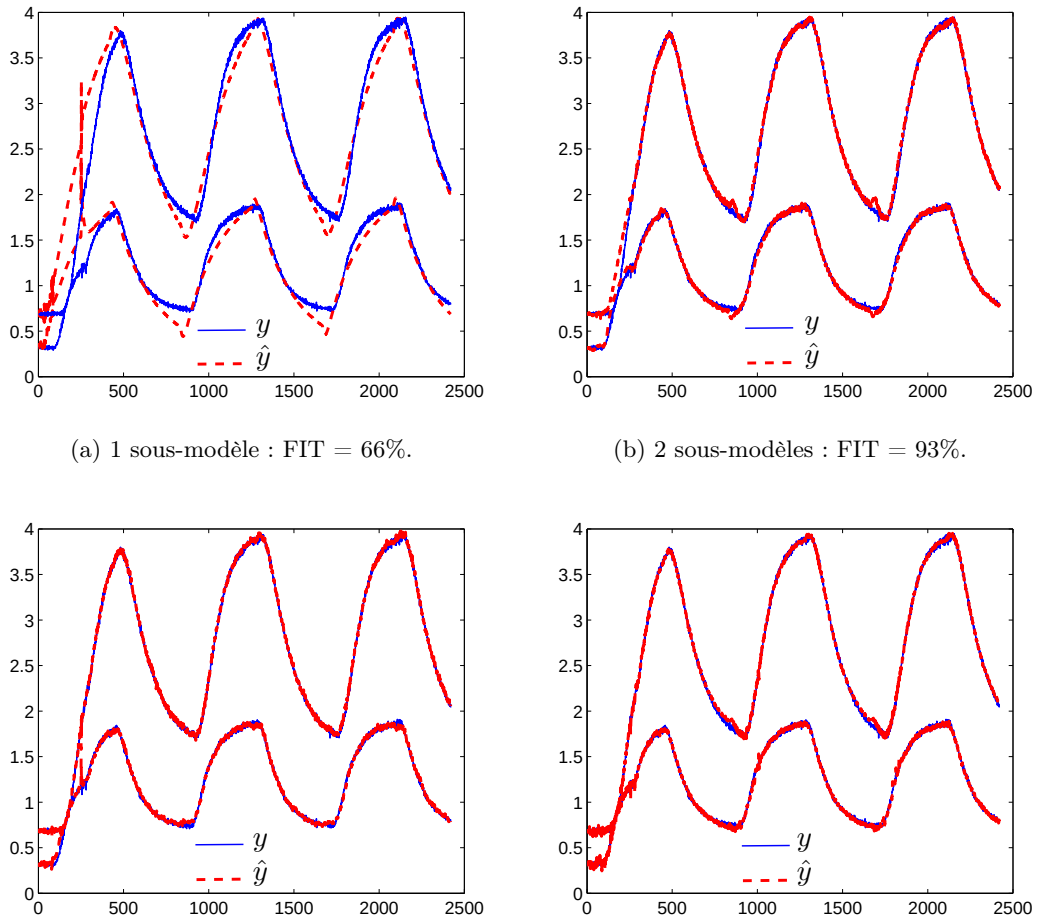
(a) 1 sous-modèle : FIT = 95%.

(b) 2 sous-modèles : FIT = 97%.

(c) 3 sous-modèles : FIT = 98%.

(d) 4 sous-modèles : FIT = 99%.

FIGURE 6.27 – Sortie réelle et sortie reconstruite du modèle SISO représentées, pour différentes valeurs du nombre de sous-modèles, en fonction du temps. L'estimation du modèle est réalisée avec l'algorithme EM-MCR.



(a) 1 sous-modèle : FIT = 66%.

(b) 2 sous-modèles : FIT = 93%.

(c) 3 sous-modèles : FIT = 97%.

(d) 4 sous-modèles : FIT = 97%.

FIGURE 6.28 – Sortie réelle et sortie reconstruite du modèle MIMO représentées, pour différentes valeurs du nombre de sous-modèles, en fonction du temps. L'estimation du modèle est réalisée avec l'algorithme EM-MCR.

6.4 Conclusion

Dans ce chapitre, nous avons, à travers la présentation de quelques résultats, illustré les potentialités d'une partie des algorithmes développés dans ce manuscrit pour l'identification de systèmes hybrides : l'algorithme d'identification de modèles d'état (Section 4.4), la méthode algébro-géométrique (Section 5.3) et la méthode EM-MCR (Section 5.6). Ces différentes techniques ont été validées aussi bien en simulation que sur des données expérimentales collectées sur un système industriel de montage automatique de composants électroniques. Nous avons aussi montré sur un système d'écoulement d'eau que la méthode EM-MCR permet d'obtenir une modélisation satisfaisante.

Toutes ces techniques fournissent, sous leurs hypothèses d'application respectives et sur les cas étudiés, des résultats satisfaisants. Cependant, nous devons mentionner que l'extraction des ordres des sous-modèles à partir des données est en général ardu sur un système réel. Cette difficulté est surtout liée au fait que la plupart des processus réels ne sont pas rigoureusement modélisables avec des structures linéaires commutants (présence d'effets non-linéaires) et que la condition de persistance d'excitation est difficile à réaliser en pratique. De plus, les changements de modes ne sont pas parfaits (instantanés) car accompagnés de régimes transitoires (mixte) plus ou moins longs.

Conclusions et perspectives

1 Résumé du mémoire

Dans ce mémoire, nous avons présenté quelques contributions à l'identification de systèmes multivariables commutants, représentés aussi bien par des structures d'état que par des modèles entrée-sortie ARX. Ce travail d'identification a été réalisé sous les hypothèses très générales où le nombre de sous-modèles constituant le modèle commutant, les ordres et les paramètres de ces sous-modèles peuvent être simultanément inconnus.

Nous avons commencé, dans le chapitre 2, par faire le point sur les méthodes disponibles dans la littérature récente pour l'estimation de systèmes hybrides. Nous avons ainsi pu constater que la plupart de ces méthodes traitent de l'identification de modèles du type PWARX, généralement SISO, quelque fois aussi MISO mais rarement MIMO. De plus, ces méthodes sont, pour la plupart, fondées sur l'hypothèse que le nombre de sous-modèles et/ou les ordres des différents sous-modèles sont connus *a priori*. Les sous-modèles sont supposés avoir des ordres qui sont tous égaux. Cependant quelques autres travaux dont l'approche algébro-géométrique (qui permet l'extraction du nombre de sous-modèles ou des ordres) s'intéressent aussi à l'identification de modèles commutants. Les rares travaux qui existent pour l'identification de systèmes MIMO découlent d'une application du concept des méthodes des sous-espaces. Malheureusement, on a besoin dans ce cas de recourir à l'hypothèse restrictive que les instants de commutation sont séparés par un certain temps de séjour minimum. Notre étude bibliographique nous a ainsi permis de mettre en évidence la nécessité de développer des méthodes d'identification de systèmes hybrides multivariables dans le contexte très délicat où ni le nombre de sous-modèles constitutifs du système hybride, ni les ordres de ces sous-modèles, ni leurs paramètres ne sont connus *a priori*.

Nous avons considéré d'abord des modèles d'état commutants. Pour estimer ces modèles par les méthodes des sous-espaces, il est indispensable de contrôler les bases de l'espace d'état dans lesquelles les matrices des sous-modèles doivent être estimées. Cela nous a conduit au développement de nouvelles techniques d'identification de modèles linéaires d'état qui possèdent la propriété intéressante de permettre la fixation de la base de coordonnées de l'état (cf. Chapitre 3). Cette nouvelle classe de méthodes d'identification structurée de modèles d'état ne requiert aucune Décomposition en Valeurs Singulières. Dans le même temps, elles permettent une extraction de l'ordre du système, même dans un contexte récursif.

Ensuite nous avons généralisé ces méthodes à l'estimation de systèmes commutants, décrits par des

modèles d'état (cf. Chapitre 4). Un algorithme très générique a été alors développé pour l'estimation de modèles d'état commutants. Cet algorithme ne nécessite ni la connaissance du mécanisme de commutation ni celle de l'état discret ni l'hypothèse bien classique que les instants de changement de mode sont séparés par un certain temps de séjour minimum. Cependant, dans le cas général, l'identification de modèles d'état hybrides est limitée par de sévères problèmes de complexité. De ce fait, nous nous sommes focalisés sur le cas particulier où les instants de commutation sont séparés par un certain temps de séjour minimum dans les différents modes du système. Nous avons alors proposé une méthode récursive pour estimer simultanément les paramètres, les ordres (ceux-ci pouvant différer d'un sous-modèle à un autre) et le nombre de sous-modèles. Puisque cette méthode estime continûment en ligne les modes du système, un même mode peut être identifié deux fois, d'où l'utilité d'une procédure conjointe de classification des modes au fur et à mesure que ceux-ci sont estimés.

Afin de nous affranchir de la contrainte de temps de séjour, une attention particulière a été accordée dans le chapitre 5, à l'identification de modèles MIMO commutants du type ARX. Dans ce cadre, nous avons généralisé la méthode algébro-géométrique de R. Vidal à l'identification de systèmes multivariables dans le contexte critique où les ordres, les paramètres ainsi que le nombre de sous-modèles sont simultanément inconnus. Une technique a été proposée pour l'extraction à la fois des ordres et du nombre de sous-modèles. Puis, les paramètres sont identifiés grâce à l'algorithme GPCA. Nous avons ensuite discuté quelques problèmes de complexité numérique et suggéré des alternatives possibles.

La dernière partie (Chapitre 6) du travail est consacrée à la validation de nos méthodes sur des exemples de simulation numérique et aussi sur des données expérimentales collectées sur une machine de montage de composants électroniques.

2 Perspectives

Il demeure plusieurs questions ouvertes qui méritent d'être explorées plus profondément dans le cadre de futures recherches. Ces questions se rapportent à la théorie de la réalisation de systèmes hybrides et concernent particulièrement la réalisation du comportement entrée-sortie des systèmes commutants au moyen de modèles d'états. L'importance de ce problème tient au fait que les modèles d'état sont très utilisés pour la mise en oeuvre de diverses méthodes d'analyse et d'interprétation de systèmes dynamiques. Ainsi, en amont du problème d'identification, il serait intéressant d'investiguer les points suivants :

– La formulation claire d'une notion de minimalité pour les modèles d'états commutants. Etant donné le comportement entrée-sortie associé à un processus commutant, ce problème revient à se demander s'il existe une dimension minimale d'espace d'état et un nombre d'état discrets minimum (dans un certain sens à définir) pour représenter ce comportement. Une difficulté évidente de ce problème est liée à la définition même de la notion de minimalité du couple (n, s) de l'ordre et du nombre d'états discrets. Comme nous l'avons montré au chapitre 4, on peut simplement choisir s le plus petit possible ($s = 1$) à condition de prendre un ordre n suffisamment grand. De même, avec $n = 1$, le nombre s a besoin d'être très important.

- Le développement de méthodes de réduction de modèles commutants. Etant donné la caractérisation d'un modèle minimal, des méthodes de réduction de modèles permettraient de convertir des modèles commutants complexes dont l'ordre ou le nombre de sous-modèles est important en un modèle plus simple. Bien évidemment, cette procédure nécessite que l'on établisse au préalable les relations mathématiques qui existent entre des modèles commutants équivalents en terme d'expression entrée-sortie.
- La construction d'un modèle d'état commutant à partir de représentations individuelles de ses sous-modèles. Disposant des modèles des sous-systèmes relativement à des bases de coordonnées quelconques de l'espace d'état (ce qui est fréquemment le cas), il s'agit dans ce problème (dans le cas bien sûr où cela est possible), de transformer ces sous-modèles de façon à les rendre « cohérents » par rapport à toute séquence d'états discrets qui peut être prescrite au système. Plus précisément, la connaissance (séparément) des sous-modèles dans des bases arbitraires n'implique pas une disponibilité du modèle global.

L'estimation proprement dite de modèles d'états commutant directement à partir d'observations entrée-sortie est un problème très complexe comme nous l'avons mis en évidence au chapitre 4. En effet, en plus des matrices et de l'état discret à estimer (et éventuellement des ordres et du nombre de sous-modèles), l'état continu est inconnu et déterminable seulement à une transformation similaire (généralisée) près. Le problème est d'autant plus délicat qu'il existe de multiples bases de représentation possibles des matrices d'état du modèle. Ainsi, même si de nombreuses applications s'appuient sur des modèles d'état, il semble en revanche bien moins difficile d'estimer des modèles entrée-sortie qui sont par ailleurs plus naturels, parce que résultant directement de la discrétisation d'équations différentielles. En résolvant les problèmes mentionnés ci-dessus, on pourra alors passer facilement d'une forme de modèle à une autre.

Dans cette optique, un effort de recherche pourrait être dirigé dans le sens de l'amélioration de certains algorithmes d'identification de modèles entrée-sortie (SARX, SARMAX, etc.). Nous avons vu par exemple que l'algorithme GPCA qui permet de séparer des données distribuées sur de multiples sous-espaces est une méthode analytique dans un contexte déterministe. Cependant, un facteur restrictif de l'applicabilité de cette méthode est le problème de la dimensionalité de l'espace des polynômes homogènes associés à l'ensemble algébrique formé par l'union des sous-espaces considérés. Ainsi, on est amené à se demander s'il est possible de proposer une alternative à cet algorithme qui permette de réduire le problème de dimensionalité tout en conservant les performances essentielles de la méthode.

Une autre piste intéressante à explorer se rapporte à la dérivation de conditions théoriques qui garantissent la convergence d'un algorithme du type classification-identification (EM-MCR) conjuguées. En général, l'étude de la convergence des méthodes de cette catégorie (alternant entre assignation des données à des sous-modèles et l'estimation des paramètres associés à ces sous-modèles) est un problème ardu à cause du couplage inextricable entre les tâches de classification et d'identification.

Notons maintenant que les méthodes existantes sont essentiellement relatives à l'identification de systèmes hybrides représentés par des modèles à temps discret. Ainsi, en s'inspirant des méthodes d'estimation de modèles linéaires à temps continus [48], il pourrait être intéressant d'envisager l'identification de modèles hybrides à temps continu. Cela introduit un problème connexe relatif à l'échantillonnage de

systèmes hybrides en fonction du temps qui sépare deux commutations consécutives.

Aussi, les méthodes d'identification présentées dans ce mémoire ainsi que la majeure partie de celles développées dans la littérature traitent de modèles hybrides dans lesquels interagissent des sous-modèles linéaires ou affines. Partant de ce constat, un prochain problème de recherche pourrait se construire autour de l'hypothèse plus générique que les différents sous-modèles sont non-linéaires (cf. par exemple [73]) ou sont de structures mixtes (sous-modèles non-linéaires mélangés à des sous-modèles linéaires).

Enfin, de futures recherches pourraient être menées dans un objectif d'application des algorithmes développés dans ce mémoire, à des procédés réels. Cela pourrait nécessiter l'ajustement des hypothèses formulées et un travail supplémentaire d'amélioration des algorithmes d'identification notamment dans des conditions réalistes où les données d'excitation ne sont pas suffisamment riches. Des applications sont envisageables par exemple à la surveillance ou au diagnostic de procédés, au traitement de signaux d'antennes, à la segmentation de séquences vidéo [130], etc.

A

Preuves

Sommaire

A.1 Preuve de la proposition 3.1

Preuve. Soient $\alpha \in \mathbb{R}^n$ et $\beta \in \mathbb{R}^{fn_u}$ vérifiant

$$\alpha^\top X_{f+1,N} + \beta^\top U_{f+1,f,N} = 0. \tag{A.1}$$

Nous avons besoin de montrer que α et β sont alors nécessairement nuls. Puisque $N \geq N_0 + n$, pour tout t tel que $f + 1 \leq t \leq f + 1 + N - N_0$, on a

$$\alpha^\top X_{t,N_0} + \beta^\top U_{t,f,N_0} = 0. \tag{A.2}$$

Par ailleurs, à partir des équations (3.1) du système, on peut écrire

$$x(t) = A^n x(t-n) + \Delta_n u_n(t-n),\ t > n, \tag{A.3}$$

où $u_n(t-n)$ est défini comme en (3.2) et

$$\Delta_n = \begin{bmatrix} A^{n-1}B & \cdots & AB & B \end{bmatrix} \in \mathbb{R}^{n \times nn_u}.$$

En utilisant le théorème de Cayley-Hamilton, l'équation (A.3) peut s'écrire

$$\begin{aligned} x(t) &= -\left(a_0 I_n + a_1 A + \cdots + a_{n-1}A^{n-1}\right) x(t-n) + \Delta_n u_n(t-n) \\ &= -(a^\top \otimes I_n)\Gamma_n(A, I_n)x(t-n) + \Delta_n u_n(t-n), \end{aligned}$$

où $\otimes$ fait référence au produit de Kronecker [26], $a = \begin{bmatrix} a_0 & \cdots & a_{n-1} \end{bmatrix}^\top$ est un vecteur formé avec les coefficients a_j, $j = 0, \cdots, n-1$, du polynôme caractéristique de A, $p_A(z) \doteq \det(zI - A) = a_0 + a_1 z + \cdots + a_n z^{n-1} + z^n$. En s'inspirant de l'équation (3.4) (c'est-à-dire en considérant le système d'entrée $u(t)$ et de sortie $x(t)$), on peut en rappelant que $w(t) = 0$, écrire

$$\Gamma_n(A, I_n)x(t-n) = x_n(t-n) - H_n(A, B, I_n, 0)u_n(t-n).$$

En portant maintenant cette expression dans l'équation précédente, on obtient après quelques simplifications algébriques,

$$x(t) = -(a^\top \otimes I_n)x_n(t-n) + \Delta_n(\mathcal{K} \otimes I_{n_u})u_n(t-n), \tag{A.4}$$

où $\mathcal{K}$ est la matrice définie par

$$\mathcal{K} = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} & 1 \\ a_2 & a_3 & \cdots & a_{n-1} & 1 & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\ a_{n-1} & 1 & \cdots & 0 & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

Pour tout $t > n$, l'équation (A.4) se lit de la façon suivante

$$x(t) + a_{n-1}x(t-1) + \cdots + a_0 x(t-n) - \Delta_n(\mathcal{K} \otimes I_{n_u})u_n(t-n) = 0. \tag{A.5}$$

Soit $\tau = f + n + 1 > n$. Alors, Eq. (A.5) implique

$$X_{\tau,N_0} + a_{n-1}X_{\tau-1,N_0} + \cdots + a_0 X_{\tau-n,N_0} - \Delta_n(\mathcal{K} \otimes I_{n_u})U_{\tau-n,n,N_0} = 0. \tag{A.6}$$

Une multiplication de l'équation (A.6) à gauche par $\alpha^\top$, produit

$$\alpha^\top X_{\tau,N_0} + a_{n-1}\alpha^\top X_{\tau-1,N_0} + \cdots + a_0\alpha^\top X_{\tau-n,N_0} - \bar{\alpha}^\top U_{\tau-n,n,N_0} = 0, \tag{A.7}$$

avec $\bar{\alpha}^\top = \alpha^\top \Delta_n(\mathcal{K} \otimes I_{n_u})$. En combinant maintenant cette dernière équation avec (A.2), il vient

$$-\beta^\top U_{\tau,f,N_0} - a_{n-1}\beta^\top U_{\tau-1,f,N_0} - \cdots - a_0\beta^\top U_{\tau-n,f,N_0} - \bar{\alpha}^\top U_{\tau-n,n,N_0} = 0. \tag{A.8}$$

Remarquons que toutes les matrices de la forme U_{k,f,N_0}, $\tau - n \leq k \leq \tau$ peuvent être exprimées comme des combinaisons linéaires des lignes de $U_{\tau-n,f+n,N_0}$. Ainsi, on a $U_{k,f,N_0} = P_k U_{\tau-n,f+n,N_0}$ pour $\tau - n \leq k \leq \tau$, et $U_{k,n,N_0} = Q_k U_{\tau-n,f+n,N_0}$, avec

$$\begin{aligned} P_{\tau-j} &= \left[0_{fn_u\times(n-j)n_u},\ I_{fn_u},\ 0_{fn_u\times jn_u}\right] \in \mathbb{R}^{fn_u\times(f+n)n_u} \\ Q_{\tau-n} &= \left[I_{nn_u},\ 0_{nn_u\times fn_u}\right] \in \mathbb{R}^{nn_u\times(f+n)n_u}, \end{aligned}$$

pour $j = 0, \ldots, n$. En faisant usage de ces notations on peut écrire

$$\Big(\beta^\top P_\tau + a_{n-1}\beta^\top P_{\tau-1} + \cdots + a_0\beta^\top P_{\tau-n} + \bar{\alpha}^\top Q_{\tau-n}\Big)U_{\tau-n,f+n,N_0} = 0. \tag{A.9}$$

Or, l'entrée $\{u(t)\}$ étant suffisamment excitante d'ordre au moins $n + f$, la matrice $U_{\tau-n,f+n,N_0} = U_{f+1,f+n,N_0}$ est de rang plein ligne. Par suite, le terme qui se trouve entre parenthèses dans (A.9), est nul. Si nous partitionnons $\beta^\top$ et $\bar{\alpha}^\top$ comme

$$\begin{aligned} \bar{\alpha}^\top &= \left[\bar{\alpha}_1^\top \quad \cdots \quad \bar{\alpha}_n^\top\right],\ \bar{\alpha}_j^\top \in \mathbb{R}^{1\times n_u}, \\ \beta^\top &= \left[\beta_1^\top \quad \cdots \quad \beta_f^\top\right],\ \beta_j^\top \in \mathbb{R}^{1\times n_u}, \end{aligned}$$

alors, des développements algébriques directs conduisent à

$$\begin{bmatrix} 1 & a_{n-1} & \cdots & \cdots & a_0 & & \\ & 1 & a_{n-1} & \cdots & \cdots & a_0 & 0 \\ & 0 & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & 1 & a_{n-1} & \cdots & \cdots & a_0 \end{bmatrix} \begin{bmatrix} \beta_1^\top \\ \vdots \\ \vdots \\ \beta_f^\top \end{bmatrix} = 0,$$

et

$$\begin{bmatrix} a_0 & & & \\ a_1 & a_0 & 0 & \\ \vdots & \vdots & \ddots & \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix} \begin{bmatrix} \beta_1^\top \\ \vdots \\ \beta_n^\top \end{bmatrix} + \begin{bmatrix} \bar{\alpha}_1^\top \\ \vdots \\ \bar{\alpha}_n^\top \end{bmatrix} = 0.$$

Il en résulte immédiatement que $\beta = 0$ et $\bar{\alpha} = \alpha^\top \Delta_n(\mathcal{K} \otimes I_{n_u}) = 0$. Cependant, le système (3.1) est atteignable ; ce qui signifie que $\text{rang}(\Delta_n) = n$ et donc, $\Delta_n(\mathcal{K} \otimes I_{n_u})$ est clairement de rang plein n (ligne). Par conséquent, on a aussi $\alpha = 0$ et ainsi, les lignes de $\begin{bmatrix} X_{f+1,N}^\top & U_{f+1,f,N}^\top \end{bmatrix}^\top$ sont linéairement indépendantes. ■

A.2 Preuve du théorème 3.2

Preuve. Pour des raisons de lisibilité, nous omettrons les indices des matrices dans l'équation (3.25). Alors, on a

$$\tilde{Y} = \tilde{\Gamma} X + \tilde{H}_f U + \tilde{V}.$$

De la factorisation RQ en (3.32), il vient

$$\tilde{Y} = R_{21}Q_1 + R_{22}Q_2 = \tilde{\Gamma}_f X + \tilde{H} R_{11} Q_1 + \tilde{V}.$$

En multipliant cette équation à gauche par S et à droite par Q_2^T, on obtient

$$S\tilde{Y}Q_2^\top = SR_{22} = \begin{bmatrix} I_n \\ \mathscr{P} \end{bmatrix} TXQ_2^\top + S\tilde{V}Q_2^\top.$$

puisque $Q_1 Q_2^\top = 0$.
Nous allons calculer maintenant le rapport du carré de cette dernière équation par N

$$\begin{aligned} \frac{1}{N} SR_{22}R_{22}^\top S^\top = \begin{bmatrix} I_n \\ \mathscr{P} \end{bmatrix} \underbrace{\frac{1}{N}(TX)\,\mathcal{Q}\,(TX)^\top}_{(I)} \begin{bmatrix} I_n \\ \mathscr{P} \end{bmatrix}^\top \\ + \underbrace{\frac{1}{N} S\tilde{V}\mathcal{Q}\tilde{V}^T S^\top}_{(II)} + S \underbrace{\frac{1}{N}\left(\tilde{\Gamma}_f X \mathcal{Q} \tilde{V}^\top + \left(\tilde{\Gamma}_f X \mathcal{Q} \tilde{V}^\top\right)^\top\right)}_{(III)} S^\top, \end{aligned} \tag{A.10}$$

où $\mathcal{Q} = Q_2^\top Q_2 = I - Q_1^\top Q_1 = I_N - U^\top \left(UU^\top\right)^{-1} U$. Notons au passage que l'existence de $\mathcal{Q}$ est subordonnée à la condition d'excitation persistante qui implique en fait que $UU^\top$ est de rang plein.

Le reste de la preuve consiste à simplifier l'équation (A.10) :

1) Commençons par introduire l'expression de $\mathcal{Q}$ dans (II). Il vient

$$(II) = S\left(\frac{1}{N}\tilde{V}\tilde{V}^\top - \frac{1}{N}\tilde{V}U^\top\left(\frac{1}{N}UU^\top\right)^{-1}\frac{1}{N}U\tilde{V}^\top\right)S^\top.$$

En exploitant les propriétés d'indépendance et d'ergodicité de la séquence $\{\tilde{v}(t)\}$ et en se rappelant que cette même séquence de bruit est statistiquement non correlée avec $\{u(t)\}$ (hypothèses A1-A2), on peut voir que le second terme dans les parenthèses tend vers 0 quand $N \to \infty$. Il reste donc $(II) \to \lim_{N\to\infty}\frac{1}{N}S\left(\tilde{V}\tilde{V}^\top\right)S^\top = \sigma_v^2\tilde{R}_v$ avec $\tilde{R}_v$ défini par (3.35).

2) Nous allons montrer maintenant que le terme (III) dans (A.10) est asymptotiquement nul. Pour ce faire, notons qu'on peut écrire

$$\frac{1}{N}\tilde{\Gamma}_f X\mathcal{Q}\tilde{V}^\top = \frac{1}{N}\tilde{\Gamma}_f X\tilde{V}^\top - \tilde{\Gamma}_f\frac{1}{N}XU^\top\left(\frac{1}{N}UU^\top\right)^{-1}\frac{1}{N}U\tilde{V}^\top. \tag{A.11}$$

En vertu de l'hypothèse A2, puisque $\tilde{V} = D_K V$, avec D_K une matrice bloc-diagonale définie par $D_K = \text{diag}(K(\gamma),\cdots,K(\gamma)) \in \mathbb{R}^{fn_y\times fn_y}$, avec $K(\gamma)$ comme en (3.24), on a $\lim_{N\to\infty}\left(\frac{1}{N}U\tilde{V}^\top\right) = 0$ et donc le second terme de (A.11) s'annule.

Il nous faut prouver maintenant que le premier terme dans la partie droite de (A.11) s'annule asymptotiquement. Pour cela, nous écrivons

$$\frac{1}{N}X\tilde{V}^\top = \frac{1}{N}\sum_{t=1}^{f}x(t)\tilde{v}_f(t)^\top + \frac{1}{N}\sum_{t=f+1}^{N}x(t)\tilde{v}_f(t)^\top. \tag{A.12}$$

Clairement, $\lim_{N\to\infty}\left(\frac{1}{N}\sum_{t=1}^{f}x(t)\tilde{v}_f(t)^\top\right) = 0$ puisque $\sum_{t=1}^{f}x(t)\tilde{v}_f(t)^\top$ est une quantité finie fixe. Pour voir que $\lim_{N\to\infty}\left(\frac{1}{N}\sum_{t=f+1}^{N}x(t)\tilde{v}_f(t)^\top\right) = 0$, on peut utiliser l'équation (3.21) pour écrire

$$\begin{cases}\tilde{y}_f(t-f) = \tilde{\Gamma}_f x(t-f) + \tilde{H}_f u_f(t-f) + D_K v_f(t-f)\\ x(t) = A^f x(t-f) + \Omega_f u_f(t-f),\end{cases} \tag{A.13}$$

où $\Omega_f = \begin{bmatrix}A^{f-1}B & \cdots & AB & B\end{bmatrix}$. De la première équation de (A.13) on tire

$$x(t-f) = \tilde{\Gamma}_f^\dagger\big(\tilde{y}_f(t-f) - \tilde{H}_f u_f(t-f) - D_K v_f(t-f)\big).$$

Une substitution dans la seconde équation de (A.13) conduit alors à

$$x(t) = A^f\tilde{\Gamma}_f^\dagger\tilde{y}_f(t-f) + \left(\Omega_f - A^f\tilde{\Gamma}_f^\dagger\tilde{H}_f\right)u_f(t-f) - A^f\tilde{\Gamma}_f^\dagger D_K v_f(t-f).$$

Développé, le second terme de (A.12) devient

$$\begin{aligned}\frac{1}{N}\sum_{t=f+1}^{N} x(t)\tilde{v}_f(t)^\top =& A^f\tilde{\Gamma}_f^\dagger D_K\left(\frac{1}{N}\sum_{t=f+1}^{N} y_f(t-f)v_f(t)^\top\right)D_K^\top \\ &+\left(\Omega_f - A^f\tilde{\Gamma}_f^\dagger\tilde{H}_f\right)\left(\frac{1}{N}\sum_{t=f+1}^{N} u_f(t-f)v_f(t)^\top\right)D_K^\top \\ &- A^f\tilde{\Gamma}_f^\dagger D_K\left(\frac{1}{N}\sum_{t=f+1}^{N} v_f(t-f)v_f(t)^\top\right)D_K^\top.\end{aligned} \tag{A.14}$$

Avec les hypothèses A2-A3, il est bien connu [121] que le vecteur de sorties *passées* $y_f(t-f)$ n'est pas corrélé avec le bruit *future* $v_f(t)$. Par conséquent, le premier terme de Eq. (A.14) s'annule asymptotiquement. De plus, par l'hypothèse A2, les deux autres termes de l'équation (A.14) tendent respectivement vers zéro quand $N \to \infty$. Ainsi,

$$\lim_{N\to\infty}\left(\frac{1}{N}X\tilde{V}^\top\right) = 0 \quad \text{et donc} \quad \lim_{N\to\infty}(II) = 0.$$

3) Finalement, en s'aidant de l'expression $\mathcal{Q} = \Pi_U^\perp$, on peut facilement reécrire la quantité (I) comme

$$(I) = \frac{1}{N}\mathsf{Z}\Pi_U^\perp\mathsf{Z}^\top, \text{ avec } \mathsf{Z} = TX.$$

Ces trois points considérés ensemble prouvent le résultat. ■

A.3 Preuve du théorème 3.3

Afin de démontrer le théorème 3.3, nous avons besoin du résultat intermédiaire suivant.

Lemme A.1. *Soient $A \in \mathbb{R}^{n\times n}$ une matrice non dérogatoire et $B = \begin{bmatrix} b_1 & \cdots & b_m \end{bmatrix} \in \mathbb{R}^{n\times m}$ une matrice quelconque. Définissons $\Omega_n(A,B) = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$,*

$$\bar{A} = \begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix} \in \mathbb{R}^{nm\times nm} \quad et \quad \bar{b} = vec(B) = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \in \mathbb{R}^{nm}.$$

Alors, on a le résultat suivant. Si $\Omega_n(A,B)$ est de rang n, alors $\Omega_n(\bar{A},\bar{b})$ est aussi de rang n.

Preuve. Supposons rang $\big(\Omega_n(A,B)\big) = n$. Il nous faut prouver que $\Omega_n(\bar{A},\bar{b})$ est alors de rang n. Pour cela, supposons qu'il existe $j < n$ tel que $\bar{A}^j\bar{b}$ soit une combinaison linéaire des $\bar{A}^i\bar{b}$, $i = 0,\ldots,j-1$. Alors, il existe un ensemble de scalaires $\alpha_0,\cdots,\alpha_{j-1}$ non tous nuls tel que $\bar{A}^j\bar{b} = \alpha_{j-1}\bar{A}^{j-1}\bar{b} + \cdots + \alpha_0\bar{b}$. Soit p le polynôme de degré j défini par $p(z) = z^j - \alpha_{j-1}z^{j-1} - \cdots - \alpha_0$. Alors,

$$p(\bar{A})\bar{b} = 0.$$

Cela équivaut à $p(A)b_i = 0$, $i = 1, \ldots, m$. Par conséquent $p(A)B = 0$ et $p(A)\Omega_n(A, B) = 0$ parce qu'en tant que polynôme de A, $p(A)$ commute avec toute puissance de A. Mais comme $\Omega_n(A, B)$ est de rang plein ligne, il s'ensuit que $p(A) = 0$. Ainsi, le polynôme minimal $m_A(z)$ de A, divise $p(z)$. Cependant, A étant non dérogatoire, son polynôme caractéristique $p_A(z)$ est égal à son polynôme minimal $m_A(z)$. Par suite, $p_A(z) = m_A(z)$ divise $p(z)$, ce qui est incompatible avec le fait que $\deg\{p(z)\} = j < \deg\{p_A(z)\} = n$. En conclusion, $\text{rang}\left(\Omega_n(\bar{A}, \bar{b})\right) = n$. ■

Preuve. [du théorème 3.3] La matrice de transfert fréquentiel entrée-sortie est donnée par

$$C(zI - A)^{-1}B + D = \frac{C\text{Adj}(zI - A)B + p_A(z)D}{p_A(z)}, \tag{A.15}$$

où $\text{Adj}(\cdot)$ est la matrice des cofacteurs de $(\cdot)$ et $p_A(z) = \det(zI - A)$ est le polynôme caractéristique de A. Notons

$$C\text{Adj}(zI - A)B = [l_{ij}(z)]_{i=1,\ldots,n_y}^{j=1,\ldots,n_u}, \tag{A.16}$$

avec $l_{ij}(z)$, le polynôme numérateur correspondant au transfert entre la i-ème sortie et la j-ème entrée. Nous devons montrer que $n_{\text{SS}} = n_{\text{ES}}$, c'est-à-dire, les polynômes $p_A(z)$ et $l_{ij}(z)$, $i = 1, \ldots, n_y$, $j = 1, \ldots, n_u$, sont premiers entre eux, si et seulement si A est non dérogatoire. Pour ce faire, nous commençons par remarquer que la matrice D peut être supposée nulle sans perte de généralité en ce qui concerne la recherche de facteurs communs entre les polynômes numérateur et dénominateur de (A.15).

Maintenant nous calculons les coefficients des polynômes $l_{ij}(z)$. Ceux-ci peuvent être obtenus directement à partir du terme $(a^\top \otimes I_{n_y})H_n + C\Delta_n$ de l'équation (3.51) (matrice qui multiplie $u_n(t-n)$). A cette étape, il est important de remarquer que

$$(a^\top \otimes I_{n_y})H_n + C\Delta_n = C\Delta_n(\mathcal{K}_o \otimes I_{n_u}),$$

avec

$$\mathcal{K}_o = \begin{bmatrix} 1 & & & \\ a_{n-1} & 1 & & 0 \\ \vdots & \vdots & \ddots & \\ a_1 & \cdots & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{A.17}$$

Soient b_j la j-ème colonne de B et $c_i^\top$ la i-ème ligne de C. Alors les n coefficients de $l_{ij}(z)$ (car ces polynômes sont tous de degré $n-1$) sont donnés par le vecteur ligne

$$l_{ij} = c_i^\top \underbrace{\begin{bmatrix} A^{n-1}b_j & \cdots & Ab_j & b_j \end{bmatrix}}_{\Delta_n(A, b_j)} \mathcal{K}_o.$$

Définissons

$$\mathscr{L}_j = \begin{bmatrix} l_{1j} \\ \vdots \\ l_{n_y j} \end{bmatrix} = C\Delta_n(A, b_j)\mathcal{K}_o$$

puis

$$\mathscr{L} = \begin{bmatrix} \mathscr{L}_1 \\ \vdots \\ \mathscr{L}_{n_u} \end{bmatrix} = \begin{bmatrix} C\Delta_n(A, b_1) \\ \vdots \\ C\Delta_n(A, b_{n_u}) \end{bmatrix} \mathcal{K}_o \in \mathbb{R}^{nn_y \times n}$$

comme une concaténation de tous les coefficients des polynômes numérateurs. Alors, selon le théorème 3.1 de Barnett [13], $n_{\text{ES}} = n_{\text{SS}}$ ou, de manière équivalente, les polynômes $l_{ij}(z)$ et $p_A(z)$ n'ont aucune racine commune si et seulement si la matrice

$$\mathscr{O} = \begin{bmatrix} \mathscr{L} \\ \mathscr{L}A_c \\ \vdots \\ \mathscr{L}A_c^{n-1} \end{bmatrix},$$

avec

$$A_c = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix},$$

est de rang plein colonne. Il nous suffit donc de montrer que $\mathscr{O}$ est de rang plein si et seulement si A est non dérogatoire. Commençons par rappeler [57] que A est non dérogatoire équivaut à : il existe une matrice non singulière S telle que $A = SA_cS^{-1}$, où A_c est la matrice compagne horizontale définie ci-dessus. Posons $\beta_i = S^{-1}b_i$. Alors on a $\Delta_n(A, b_i) = S\Delta_n(A_c, \beta_i)$. A une permutation de lignes près, $\mathscr{O}$ est égale à une matrice

$$\mathscr{O}_1 = \left[\begin{array}{c} CS\Delta^1 \\ CS\Delta^1 A_c \\ \vdots \\ CS\Delta^1 A_c^{n-1} \\ \hline \vdots \\ \hline CS\Delta^{n_u} \\ CS\Delta^{n_u} A_c \\ \vdots \\ CS\Delta^{n_u} A_c^{n-1} \end{array}\right],$$

où $\Delta^i = \Delta_n(A_c, \beta_i)\mathcal{K}_o$. Maintenant il est important de remarquer que pour tout i, Δ^i commute avec A_c et donc avec toute puissance de A_c. On peut donc écrire

$$\mathscr{O}_1 = \begin{bmatrix} \Gamma_n & & \\ & \ddots & \\ & & \Gamma_n \end{bmatrix} \begin{bmatrix} S\Delta^1 \\ \vdots \\ S\Delta^{n_u} \end{bmatrix} = \begin{bmatrix} \Gamma_n & & \\ & \ddots & \\ & & \Gamma_n \end{bmatrix} \Omega_n(\bar{A}, \bar{b})\mathcal{K}_o,$$

avec $\Gamma_n = \Gamma_n(A, C)$. Comme le système est observable, la matrice contenant Γ_n dans l'équation précédente est de rang plein colonne. Ainsi, en utilisant le Lemme A.1, et en rappelant que $\mathcal{K}_o$ est non singulière, il vient rang $\left(\mathscr{O}\right) = \text{rang}\left(\mathscr{O}_1\right) = \text{rang}\left(\Omega_n(\bar{A}, \bar{b})\right) = n$.

Réciproque : Supposons maintenant A dérogatoire. Il nous reste à montrer que cette hypothèse implique que les polynômes $l_{ij}(z)$, $i = 1, \ldots, n_y$, $j = 1, \ldots, n_u$ et $p_A(z)$ ne sont pas relativement premiers. Pour cela, commençons par remarquer le fait suivant. Pour tout polynôme $p(z) = \alpha_0 + \alpha_1 z + \cdots + \alpha_{r-1}z^{r-1} + z^r$, on peut écrire

$$\begin{aligned} & p(z)I_n - p(A) \\ & = \alpha_1(zI - A) + \cdots + \alpha_{r-1}(z^{r-1}I - A^{r-1}) + (z^r I - A^r) \\ & = (zI - A)Q(z, A), \end{aligned}$$

où $Q(z, A)$ est une matrice polynomiale qui s'exprime en fonction de z et A. En prenant en particulier $p(z)$ égal au polynôme minimum $m_A(z)$ de A, on peut écrire

$$m_A(z)I_n - m_A(A) = (zI - A)Q_o(z, A)$$

pour une certaine matrice polynomiale $Q_o(z, A)$. Or $m_A(A) = 0$ par définition de $m_A(z)$, d'où en multipliant à gauche par $\text{Adj}(zI - A)$, on obtient

$$\begin{aligned} m_A(z)\text{Adj}(zI - A) &= \text{Adj}(zI - A)(zI - A)Q_o(z, A) \\ &= p_A(z)Q_o(z, A). \end{aligned} \tag{A.18}$$

Par ailleurs, on sait que $m_A(z)$ divise $p_A(z)$. Il existe donc un polynôme $h(z)$ tel que $p_A(z) = m_A(z)h(z)$. En remplaçant dans (A.18), on a $m_A(z)\text{Adj}(zI - A) = m_A(z)h(z)Q_o(z, A)$. Divisons maintenant par $m_A(z)$ puis multiplions par C et B respectivement à gauche et à droite. Il vient

$$C\text{Adj}(zI - A)B = h(z)CQ_o(z, A)B.$$

Ainsi, $h(z)$ divise $C\text{Adj}(zI - A)B$, c'est-à-dire que $h(z)$ divise chacun des polynômes $l_{ij}(z)$ (cf. (A.16)). En se servant maintenant du fait que A est dérogatoire, on a $\deg\{h(z)\} \geq 1$. Alors, comme $h(z)$ divise aussi $p_A(z)$, on peut conclure que les polynômes $l_{ij}(z)$ et $p_A(z)$ ne sont pas relativement premiers.

En conclusion $n_{\text{ES}} = n_{\text{SS}}$ si et seulement si A est non dérogatoire selon l'énoncé du théorème. ■

A.4 Quelques résultats intermédiaires

Dans cette section, nous donnons quelques résultats intermédiaires qui seront utilisés pour démontrer les propositions 5.1 et 5.2 respectivement dans les sections A.5 et A.6.

Lemme A.2. *Soient* $\{V_i\}_{i=1}^{p}$ *et* $\{W_j\}_{j=1}^{q}$ *deux ensembles de sous-espaces de* $\mathbb{R}^K$ *tels que*

$$V_1 \cup \cdots \cup V_p \subset W_1 \cup \cdots \cup W_q.$$

Alors, pour tout $i \in \{1, \cdots, p\}$ *il existe au moins un* $j \in \{1, \cdots, q\}$ *tel que* $V_i \subset W_j$.

Preuve. Puisque pour tout i, $V_i \subset \cup_i (V_i) \subset \cup_j (W_j)$, il suffit de prouver le résultat pour $p = 1$. Ainsi, nous procédons par récurrence sur q seulement. Le cas $q = 1$ est trivialement vrai. Supposons que le résultat soit vrai pour une union de q sous-espaces W_j et supposons $V_1 \subset W_1 \cup \cdots \cup W_q \cup W_{q+1}$.

Notons $\bar{W} = W_1 \cup \cdots \cup W_q$. Pour montrer que V_1 est alors nécessairement inclus dans W_j pour un certain $j \in \{1, \cdots, q+1\}$, nous allons procéder par contradiction. Supposons que V_1 n'est inclus ni dans W_{q+1} ni dans $\bar{W}$. Alors, on peut trouver des vecteurs x_o et x_1 non nuls tels que $x_o \in V_1 \setminus \bar{W}$ and $x_1 \in V_1 \setminus W_{q+1}$. Mais alors, pour tout $\lambda \in \mathbb{R}$, $\lambda \neq 0$, on a $x_o + \lambda x_1 \in V_1 \subset \bar{W} \cup W_{q+1}$ car V_1 est un sous-espace. Clairement, $x_o + \lambda x_1$ ne peut être dans W_{q+1} parce que dans ce cas, x_1 serait aussi dans W_{q+1}, x_o étant un élément de W_{q+1}. Par conséquent, la droite affine $(D) : x_o + \lambda x_1$, $\lambda \neq 0$, doit être entièrement contenue dans $\bar{W}$. Remarquons aussi que pour un sous-espace W_j, $j \in \{1, \cdots, q\}$, la droite $x_o + \lambda x_1$ ne peut intersecter W_j en deux points distincts. Pour s'en convaincre admettons que $x_o + \lambda_1 x_1$ et $x_o + \lambda_2 x_1$, $\lambda_1 \neq \lambda_2$, appartiennent à W_j. Alors $\lambda_2 (x_o + \lambda_1 x_1) - \lambda_1 (x_o + \lambda_2 x_1) = (\lambda_1 - \lambda_2) x_o$ est aussi dans W_j puisque W_j est un sous-espace. Cela est impossible parce que $x_o \notin W_j$. Ainsi, la droite (D) coupe $\bar{W}$ en au plus q points et donc (D) ne peut pas être entièrement incluse dans $\bar{W}$, ce qui constitue une contradiction.
Nous pouvons maintenant conclure que $V_1 \subset \bar{W}$ ou $V_1 \subset W_{q+1}$, laquelle conclusion induit immédiatement l'assertion formulée par le Lemme A.2. ■

Lemme A.3. *Soient* $\mathcal{H} = \mathrm{Sym}\left(P_1^\top \otimes \cdots \otimes P_s^\top\right)$ *et* $h = \mathrm{Sym}\left(\theta_1 \otimes \cdots \otimes \theta_s\right)$ *avec* $P_j^\top \in \mathbb{R}^{K \times m}$, *et* $\theta_j \in \mathbb{R}^K$, $j = 1, \ldots, s$.
Si (1) h *est linéairement indépendant des colonnes de* $\mathcal{H}$, *alors (2) Il n'existe aucune permutation* $\sigma \in \mathfrak{S}_s$ *telle que pour tout* $j \in \{1, \ldots, s\}$, θ_j *est linéairement dépendant des colonnes de* $P_{\sigma(j)}^\top$.

Preuve. En fait, nous allons prouver la contraposée de cette assertion. Supposons qu'il existe une permutation $\sigma \in \mathfrak{S}_s$ vérifiant

$$\forall j \in \{1, \ldots, s\}, \exists \lambda_j \in \mathbb{R}^m / \theta_j = P_{\sigma(j)}^\top \lambda_j.$$

Alors, on a $h = \mathrm{Sym}\left(\theta_1 \otimes \cdots \otimes \theta_s\right) = \mathrm{Sym}\left(P_{\sigma(1)}^\top \lambda_1 \otimes \cdots \otimes P_{\sigma(s)}^\top \lambda_s\right)$. En utilisant la remarque faite dans l'exemple de la section 5.3 (avant l'équation (5.7)), il existe une matrice $S(s, K) \in \mathbb{R}^{M_s(K) \times K^s}$ remplie

avec des éléments 0 et 1 telle que le produit tensoriel symétrique ci-dessus puisse se réécrire

$$\begin{aligned} h &= S(s,K)\left(P_{\sigma(1)}^\top \lambda_1 \otimes \cdots \otimes P_{\sigma(s)}^\top \lambda_s\right) \\ &= S(s,K)\left(P_{\sigma(1)}^\top \otimes \cdots \otimes P_{\sigma(s)}^\top\right)(\lambda_1 \otimes \cdots \otimes \lambda_s) \\ &= \underbrace{\mathrm{Sym}\left(P_{\sigma(1)}^\top \otimes \cdots \otimes P_{\sigma(s)}^\top\right)}_{\mathcal{H}_o}\underbrace{(\lambda_1 \otimes \cdots \otimes \lambda_s)}_{\bar{\lambda}} \\ &= \mathcal{H}_o\bar{\lambda}, \end{aligned}$$

où la propriété du produit mixte de Kronecker [26] a été appliquée. σ étant une permutation, il n' y a pas de redondance dans l'ensemble d'indices $\{\sigma(1),\cdots,\sigma(s)\}$. Par conséquent $\mathcal{H}_o$ est égal à $\mathcal{H}$ à une permutation de colonnes près. D'où on obtient que $h = \mathcal{H}_o\bar{\lambda}$ est linéairement dépendant des colonnes de $\mathcal{H}$. ■

A.5 Preuve de la proposition 5.1

Preuve. Soit r le rang de la matrice $\mathcal{H}$. Alors, il est clair que $n_h \geq r$.
Nous allons montrer que si les hypothèses (5.14) et (5.15) sont simultanément satisfaites, alors $n_h = r$. Nous procédons par contradiction. Pour cela, supposons $n_h > r$. Alors, il existe $h \in \mathbb{R}^{M_s(K)}$, h linéairement indépendant des colonnes de $\mathcal{H}$ et $h^\top \nu_s(x) = 0$ pour tout $x \in \mathscr{A}$. De ce fait, on peut noter que h est factorisable de la façon suivante : $h = \mathrm{Sym}\,(b_1 \otimes \cdots \otimes b_s)$, avec $b_j \in \mathbb{R}^K$, $j = 1,\ldots,s$ non nuls. Puisque pour tout $x \in \mathscr{A}$, $(b_1^\top x)\cdots(b_s^\top x) = \mathrm{Sym}\,(b_1 \otimes \cdots \otimes b_s)^\top \nu_s(x) = 0$, on a $\mathscr{A} = \mathrm{null}(P_1) \cup \cdots \cup \mathrm{null}(P_s) \subset (b_1)^\perp \cup \cdots \cup (b_s)^\perp$, où $(b_j)^\perp$ fait référence à l'hyperplan homogène [1] qui est orthogonal à b_j. Selon le lemme A.2, pour tout $i \in \{1,\ldots,s\}$ il existe au moins un $j \in \{1,\ldots,s\}$ tel que $\mathrm{null}(P_i) \subset (b_j)^\perp$. Cela entraîne que $\mathrm{null}(P_i)^\perp = \mathrm{span}(P_i^\top) \supset \mathrm{span}(b_j)$. Autrement dit, il existe j tel que b_j est linéairement dépendant des colonnes de $P_i^\top$.
Cependant, le lemme A.3 exclut l'existence de toute bijection σ telle que $\sigma(i) = j$. On peut donc trouver deux indices p et q avec $p \neq q$, tels que $\mathrm{null}(P_p)$ et $\mathrm{null}(P_q)$ soient inclus dans le même hyperplan $(b_k)^\perp$ pour un certain k, c'est-à-dire, $\mathrm{null}(P_p) \subset (b_k)^\perp$ et $\mathrm{null}(P_q) \subset (b_k)^\perp$. Cela entraîne que $b_k \in \mathrm{span}(P_p^\top) \cap \mathrm{span}(P_q^\top)$. Par suite, $\mathrm{span}(P_p^\top) \cap \mathrm{span}(P_q^\top) \neq \{0\}$. La condition (5.15) de la proposition impose alors $\mathrm{rang}\left(\begin{bmatrix} P_p^\top & P_q^\top \end{bmatrix}\right) = K$, d'où $\mathrm{rang}(P_p^\top) + \mathrm{rang}(P_q^\top) \geq K+1$. Par conséquent on a $\sum_{i=1}^s \mathrm{rang}(P_i^\top) \geq K + s - 1$, ce qui contredit la condition (5.14).
En conclusion, le nombre n_h est exactement égal à $\mathrm{rang}(\mathcal{H})$. ■

A.6 Preuve de la proposition 5.2

Preuve. Commençons par noter que $\mathrm{rang}(\mathcal{H})$ ne peut excéder $\prod_{j=1}^s r_j$, où $r_j = \mathrm{rang}(P_j^\top)$. En effet, comme nous l'avons indiqué précédemment (cf. Section 5.3), il existe une matrice $S(s,K) \in \mathbb{R}^{M_s(K)\times K^s}$

1. hyperplan passant par l'origine.

remplie avec des 0 et des 1 telle que $\mathcal{H} = S(s,K)\left(P_1^\top \otimes \cdots \otimes P_s^\top\right)$ où $\otimes$ dénote le produit de Kronecker. Un résultat basique d'algèbre linéaire nous permet alors de conclure que

$$\text{rang}(\mathcal{H}) \leq \text{rang}\left(P_1^\top \otimes \cdots \otimes P_s^\top\right) = \prod_{j=1}^{s} r_j.$$

Excluons maintenant le cas trivial où l'un des $P_j^\top$ est nul. Alors, une consequence de la condition (5.15) est qu'on peut, en utilisant r_j colonnes linéairement indépendantes de $P_j^\top$, $j = 1, \ldots, s$, construire des matrices $L_j \in \mathbb{R}^{K\times r_j}$ telles qu'en définissant $L = \left[L_1, \cdots, L_s\right] \in \mathbb{R}^{K\times r}$, on ait $\text{rang}\,(L) = \min(r, K)$, avec $r = \sum_{j=1}^{s} r_j$. Considérons la sous-matrice $\mathcal{H}_L$ de colonnes de $\mathcal{H}$ définie par $\mathcal{H}_L = \text{Sym}\,(L_1 \otimes \cdots \otimes L_s)$. Nous allons prouver par contradiction que $\text{rang}\,(\mathcal{H}_L)$ vaut en fait $\prod_{j=1}^{s} r_j$.
Si cela n'était pas vrai, alors en notant $h_{i_1\cdots i_s}$ les colonnes de $\mathcal{H}_L$, il existerait un ensemble de scalaires $\lambda_{i_1\cdots i_s}$ non tous nuls tel que $\sum_{i_1\cdots i_s} \lambda_{i_1\cdots i_s} h_{i_1\cdots i_s} = 0$. Chaque $h_{i_1\cdots i_s}$ peut être regardé comme le vecteur de coefficients du polynôme $p_{i_1\cdots i_s}(x) = \left(b_{i_1}^\top x\right)\cdots\left(b_{i_s}^\top x\right) = h_{i_1\cdots i_s}^\top \nu_s(x)$, où $x \in \mathbb{R}^K$. Ainsi, on a

$$Q(x) = \sum_{i_1\cdots i_s} \lambda_{i_1\cdots i_s} p_{i_1\cdots i_s}(x) = 0 \quad \forall x \in \mathbb{R}^K. \tag{A.19}$$

Soit $(i_1^o, \cdots, i_s^o)$ un vecteur d'indices tel que $\lambda_{i_1^o\cdots i_s^o} \neq 0$. L'équation (A.19) peut être reécrite comme

$$Q(x) = p_{i_1^o\cdots i_s^o}(x) + \sum_{i_1\cdots i_s} \lambda_{i_1^o\cdots i_s^o}^{-1} \lambda_{i_1\cdots i_s} p_{i_1\cdots i_s}(x) = 0 \quad \forall\, x. \tag{A.20}$$

Définissons $L^o = \begin{bmatrix} L_1^o & \cdots & L_s^o \end{bmatrix}$ comme étant la matrice L mais dans laquelle les colonnes $b_{i_1^o}, \cdots, b_{i_s^o}$ ont été omises. Comme $\text{rang}(L^o) = r - s < K$, $(L^o)^\perp$, le complément orthogonal à l'espace colonne de L^o contient au moins un vecteur non nul x. Pour un tel x, il est facile de vérifier que le second terme de (A.20) est nul de sorte qu'il reste seulement $p_{i_1^o\cdots i_s^o}(x) = \left(b_{i_1^o}^\top x\right)\cdots\left(b_{i_s^o}^\top x\right) = 0$. En d'autres termes,

$$(L^o)^\perp \subset \left(b_{i_1^o}\right)^\perp \cup \cdots \cup \left(b_{i_s^o}\right)^\perp. \tag{A.21}$$

Selon le lemme A.2, il existe un indice k tel que $(L^o)^\perp \subset \left(b_{i_k^o}\right)^\perp$. Mais cela contredit le fait que les colonnes de L^o sont linéairement indépendantes de $b_{i_k^o}$. Pour le voir, considérons $T = \begin{bmatrix} L^o & b_{i_k^o} & G \end{bmatrix} \in \mathbb{R}^{K\times K}$, où G est une matrice de dimensions appropriées avec des colonnes linéairement indépendantes de $\begin{bmatrix} L^o & b_{i_k^o} \end{bmatrix}$. Puisque T engendre alors $\mathbb{R}^K$, on peut trouver un $x \in \mathbb{R}^K$ vérifiant $T^\top x = \begin{bmatrix} 0 & 1 & c^\top \end{bmatrix}^\top$ avec c un vecteur quelconque. Ainsi, $x \in (L^o)^\perp$ mais $x \notin \left(b_{i_1^o}\right)^\perp \cup \cdots \cup \left(b_{i_s^o}\right)^\perp$, ce qui constitue une contradiction avec (A.21). ■

Liste de publications

Revues internationales

- L. Bako, G. Mercère, S. Lecoeuche and M. Lovera. *Recursive Subspace Identification of Hammerstein Models Based on Least Squares Support Vector Machines.* IET Control Theory & Applications, 2008 (accepté).
- L. Bako, G. Mercère, and S. Lecoeuche. *Online structured subspace identification with application to switched linear systems.* International Journal of Control, 2008 (à paraître).
- G. Mercère, L. Bako, and S. Lecoeuche. *Propagator-based methods for recursive subspace model identification.* Signal Processing, vol. 88, pp. 468-491, 2008.

Conférences internationales

- L. Bako, G. Mercère, R. Vidal, and S. Lecoeuche. *Identification of switched linear state space models without minimum dwell time.* IFAC Symposium on System Identification, 2009 (soumis).
- L. Bako, G. Mercère and S. Lecoeuche. *Identification of Multivariable Canonical State-Space Representations.* IFAC Symposium on System Identification, 2009 (soumis).
- L. Bako, G. Mercère and S. Lecoeuche. *A Least Squares Approach to the Subspace Identification Problem.* IEEE Conference on Decision and Control, Cancun, Mexico, 2008.
- L. Bako, G. Mercère and S. Lecoeuche. *Identification structurée de modèles d'état de systèmes multivariables.* Conférence Internationale Francophone d'Automatique, Bucarest, Romania, 2008.
- L. Bako and R. Vidal. *Algebraic Identification of Switched MIMO ARX Systems.* Hybrid Systems : Computation and Control, Springer-Verlag, St. Louis, USA, 2008.
- A. Akhenak, L. Bako, E. Duviella, K. M. Pekpe and S. Lecoeuche. *Fault diagnosis for switching system using observer Kalman filter identification.* IFAC World Congress, Seoul, Korea, 2008.
- E. Duviella, L. Bako and P. Charbonnaud. *Gaussian and boolean weighted models to represent va-*

riable dynamics of open channel systems. IEEE Conference on Decision and Control, New Orleans, USA, 2007.

– L. Bako, G. Mercère, S. Lecoeuche, and M. Lovera. *Recursive subspace identification of Hammerstein models using LS-SVM.* IFAC Workshop on Adaptation and Learning in Control and Signal Processing, Saint Petersburg, Russia, 2007.

– L. Bako, G. Mercère, and S. Lecoeuche. *Online structured identification of switching systems with possibly varying orders.* European Control Conference, Kos, Greece, 2007.

– H. A. Boubacar, G. Mercère, L. Bako, and S. Lecoeuche. *Suivi de systèmes non stationnaires basé sur une approche de reconnaissance des formes.* Conférence Internationale Francophone d'Automatique, Bordeaux, France, 2006.

Conférences nationales françaises

– L. Bako, and S. Lecoeuche. *Identification récursive de systèmes à commutations.* 2èmes Journées Doctorales/ Journées Nationales MACS, Reims, France, 2007.

– L. Bako, G. Mercère, and S. Lecoeuche. *Identification en ligne de systèmes commutants à structure variable.* Journées Identification et Modélisation Expérimentale, Poitiers, France, 2006.

Bibliographie

[1] J. C. Aguero and G. C. Goodwin. A recursive algorithm for MIMO stochastic model estimation. In *Asian Control Conference*, 2004.

[2] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19 :716–723, 1974.

[3] E. Amaldi and M. Mattavelli. The MIN PFS problem and piecewise linear model estimation. *Discrete Applied Mathematics*, 118 :115–143, 2002.

[4] B. D. O. Anderson and C. J. Richard Johnson. Exponential convergence of adaptive identification and control algorithms. *Automatica*, 18 :1–13, 1982.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50 :174–188, 2002.

[6] M. Babaali and M. Egerstedt. Observability for switched linear systems. In *Hybrid Systems : Computation and Control, Philadelphia PA, USA*, 2004.

[7] E. W. Bai and Y. Liu. Recursive direct weight optimization in nonlinear system identification : a minimal probability approach. *IEEE Transaction on Automatic Control*, 52 :1218–1231, 2007.

[8] L. Bako and S. Lecoeuche. Identification récursive de systèmes à commutation. In *2èmes Journées Doctorales/ Journées Nationales MACS, Reims, France*, 2007.

[9] L. Bako, G. Mercère, and S. Lecoeuche. Online subspace identification of switching systems with possibly varying orders. In *European Control Conference, Kos, Greece*, 2007.

[10] L. Bako, G. Mercère, and S. Lecoeuche. A least squares approach to the subspace identification problem. In *Conference on Decision and Control, Cancun, Mexico*, 2008.

[11] L. Bako, G. Mercère, and S. Lecoeuche. Online structured subspace identification with application to switched linear systems. *International Journal of Control (Submitted)*, 2008.

[12] L. Bako and R. Vidal. Algebraic identification of switched MIMO ARX models. In *Hybrid Systems : Control and Computation, St Louis MO, USA*, 2008.

[13] S. Barnett. Matrices, polynomials, and linear time-invariant systems. *IEEE Transactions on Automatic Control*, Ac-18 :1–10, 1973.

[14] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes : Theory and Application*. Prentice-Hall, Inc., Simon & Schuster Company, Englewood Cliffs, NJ, 1993.

[15] D. Bauer. Order estimation for subspace methods. *Automatica*, 37 :1561–1573, 2001.

[16] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45 :1864–1876, 2000.

[17] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. Data classification and parameter estimation for the identification of piecewise affine models. In *Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

[18] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50 :1567–1580, 2005.

[19] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35 :407–427, 1999.

[20] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods Software*, 1 :23–34, 1992.

[21] K. P. Bennett and O. L. Mangasarian. Multicategory discrimination via linear programming. *Optimization Methods and Software*, 4 :27–39, 1994.

[22] J. Borges, V. Verdult, M. Verhaegen, and M. A. Botto. A switching detection method based on projected subspace classification. In *Conference on Decision and Control-European Control Conference, Seville, Spain*, 2005.

[23] A. Boukhris, G. Mourot, and J. Ragot. Non-linear dynamic system identification : a multi-model approach. *International Journal of Control*, 72 :591–604, 1999.

[24] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004, 2004.

[25] L. Breiman. Hinging hyperplanes for regression, classification and function approximation. *IEEE Transactions on Information Theory*, 39 :999–1013, 1993.

[26] J. W. Brewer. Kronecker product and matrix calculus in system theory. *IEEE Transactions on Circuits and Systems*, 25 :772–781, 1978.

[27] P. E. Caines. On the asymptotic normality of instrumental variable and least squares estimators. *IEEE Transactions on Automatic Control*, 21 :598–600, 1976.

[28] S. G. Cao, N. W. Rees, and G. Feng. Analysis and design for a class of complex control systems part i : Fuzzy modelling and identification. *Automatica*, 33 :1017–1028, 1997.

[29] C. T. Chou and M. Verhaegen. Subspace algorithms for the identification multivariables dynamic error-in-variables models. *Automatica*, 33 :1857–1869, 1997.

[30] V. T. Chow, D. R. Maidment, and L. W. Mays. *Applied Hydrology*. McGraw-Hill, New York, Paris, 1988.

[31] L. O. Chua and A. C. Deng. Canonical piecewise-linear representation. *IEEE Transactions on Circuits and Systems*, 35 :101–111, 1998.

[32] L. O. Chua and S. M. Kang. Sectionwise piecewise-linear functions : Canonical representation, properties and applications. *Proceedings of the IEEE*, 65 :915–929, 1977.

[33] Dash-Associates. X-press-mp user guide. Technical report, URL http ://www.dashopt.com, 1999.

[34] K. De Cock and B. De Moor. Subspace angles between arma models. *Systems and Control Letters*, 46 :265– 270, 2002.

[35] K. De Cock, G. Mercère, and B. De Moor. Recursive subspace identification for in flight modal analysis of airplanes. In *ISMA International Conference on Noise and Vibration Engineering, Leuven, Belgium*, 2006.

[36] B. De Schutter. Optimal control of a class of linear hybrid systems with saturation. *SIAM Journal on Control and Optimization*, 39 :835–851, 2000.

[37] B. De Schutter and T. Van den Boom. On model predictive control for max-min-plus-scaling discrete event systems. Technical report, Technical Report bds :00-04, Control Lab, Fac. ITS, Delft Univ. Techn., Delft, The Netherlands, 2000.

[38] H. Derksen. Hilbert series of subspaces arrangements. *Journal of Pure and Applied Algebra*, 209 :91–98, 2007.

[39] E. A. Domlan. *Diagnostic des systèmes à changement de régime de fonctionnement*. PhD thesis, Institut National Polytechnique de Lorraine, 2006.

[40] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. New York : Wiley, 1973.

[41] E. Duviella. *Conduite réactive de systèmes dynamiques étendus à retards variable : cas des réseaux hydrographiques*. PhD thesis, Institut National Polytechnique de Toulouse, 2005.

[42] E. Duviella, L. Bako, and P. Charbonnaud. Gaussian and boolean weighted models to represent variable dynamics of open channel systems. In *Conference on Decision and Control, New Orleans LA, USA*, 2007.

[43] M. Egerstedt and B. M. (Eds). *Hybrid Systems : Computation and Control (HSCC)*. Proceedings of HSCC 2008, Springer Verlag, St. Louis MO, USA, 2008.

[44] E. Elhamifar, M. Petreczky, and R. Vidal. Observability of jump linear systems with inputs. In *Technical report, The Johns Hopkins University*, 2008.

[45] G. Ferrari-Trecate. Hybrid identification toolbox (HIT). Technical report, http ://sisdin.unipv.it/lab/personale/pers_hp/ferrari/HIT_toolbox.html, 2005.

[46] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39 :205–217, 2003.

[47] M. Fliess, C. Join, and W. Perruquetti. Real-time estimation for switched linear systems. In *Conference on Decision and Control, Mexico, Cancun*, 2008.

[48] H. Garnier and L. W. (Eds). *Identification of continuous-time models from sampled data*. Springer-Verlag, London, 2008.

[49] G. Golub and C. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1996.

[50] C. Guzelis and I. Goknar. A canonical representation for piecewise-affine maps and its applications to circuit analysis. *IEEE Transactions on Circuits and Systems*, 38 :1342 – 1354, 1991.

[51] P. R. Halmos. *Measure Theory*. New York : Springer-Verlag, 1974.

[52] J. Harris. *Algebraic Geometry : A First Course*. Springer-Verlag, 1992.

[53] Y. Hashambhoy and R. Vidal. Recursive identification of switched ARX models with unknown number of models and unknown orders. In *Conference on Decision and Control, Seville, Spain*, 2005.

[54] B. Haverkamp. *State space identification : theory and practice*. PhD thesis, Delft University of Technology, 2001.

[55] W. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37 :1085–1091, 2001.

[56] W. P. M. H. Heemels, J. M. Schumacher, and S. Weiland. Linear complementarity systems. *SIAM Journal of Applied Mathematics*, 60 :1234–1269, 2000.

[57] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1985.

[58] K. Huang, A. Wagner, and Y. Ma. Identification of hybrid linear time-invariant systems via subspace embedding and segmentation. In *Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

[59] M. Jansson and B. Wahlberg. A linear regression approach to state space subspace system identification. *Signal Processing*, 52 :103–129, 1996.

[60] M. Jansson and B. Wahlberg. On the consistency of subspace methods for system identification. *Automatica*, 34 :1507–1519, 1998.

[61] T. A. Johansen and B. A. Foss. Identification of non-linear system structure and parameters using regime decomposition. *Automatica*, 31 :321–326, 1995.

[62] T. A. Johansen, R. Shorten, and R. Murray-Smith. On the interpretation and identification of dynamic takagi-sugeno fuzzy models. *IEEE Transactions on Fuzzy systems*, 8 :297–313, 2000.

[63] R. M. Johnstone, C. J. R. Johnson, R. R. Bitmead, and B. D. Anderson. Exponential convergence of recursive least squares with exponential forgetting factor. In *Conference on Decison and Control*, 1982.

[64] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjoberg, , and Q. Zhang. Nonlinear black box models in system identification : Mathematical foundations. *Automatica*, 31 :1725–1751, 1995.

[65] A. L. Juloski, W. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J. Niessen. Comparison of four procedures for the identification of hybrid systems. In *Hybrid systems : computation and control, Zurich, Switzerland*, 2005.

[66] A. L. Juloski, W. P. M. H. Heemels, and G. Ferrari-Trecate. Data-based hybrid modelling of the component placement process in pick-and-place machines. *Control Engineering Practice*, 12 :1241–1252, 2004.

[67] A. L. Juloski, S. Weiland, and W. Heemels. A bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50 :1520–1533, 2005.

[68] B. C. Juriceka, D. E. Seborg, and W. E. Larimore. Identification of the tennessee eastman challenge process with subspace methods. *Control Engineering Practice*, 9 :1337–1351, 2001.

[69] T. Kailath. *Linear Systems*. Prentice Hall Inc., Englewood Cliffs, NJ 07632, 1980.

[70] T. Kailath and A. H. Sayed. *Fast Reliable Algorithms for Matrices with Structure*. SIAM, PA, 1999.

[71] S. M. Kang and L. O. Chua. A global representation of multidimensional piecewise-linear functions with linear partitions. *IEEE Transactions on Circuits and Systems*, CAS-25 :938–940, 1978.

[72] T. Katayama. *Subspace methods for system identification*. Springer-Verlag, 2005.

[73] F. Lauer and G. Bloch. Switched and piecewise nonlinear hybrid system identification. In *11th International Conference on Hybrid Systems : Computation and Control*, 2008.

[74] L. Ljung. *System Identification : Theory for the user (2nd Ed.)*. PTR Prentice Hall., Upper Saddle River, USA, 1999.

[75] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, 1983.

[76] M. Lovera, T. Gustafsson, and M. Verhaegen. Recursive subspace identification of linear and non-linear wiener state-space models. *Automatica*, 36 :1639–1650, 2000.

[77] Y. Ma and R. Vidal. Identification of deterministic switched arx systems via identification of algebraic varieties. In *Hybrid systems computation and control, Zurich, Switzerland*, 2005.

[78] Y. Ma, A. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50, 2008.

[79] S. Marcos, A. Marsal, and M. Benidir. The propagator method for source bearing estimation. *Signal Processing*, 42 :121–138, 1995.

[80] G. Mercère. *Contribution à l'identification récursive des systèmes par l'approche des sous espaces*. PhD thesis, Université Scientifique et Technologique de Lille, 2004.

[81] G. Mercère, L. Bako, and S. Lecoeuche. Propagator-based methods for recursive subspace model identification. *Signal Processing*, 2008 :468–491, 88.

[82] G. Mercère, S. Lecoeuche, and C. Vasseur. A new recursive method for subspace identification of noisy systems : EIVPM. In *IFAC Symposium on System Identification*, 2003.

[83] C. D. Meyer. *Matrix analysis and applied linear algebra*. Cambridge University Press, 2000.

[84] 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.

[85] E. Münz and V. Krebs. Identification of hybrid systems using a priori knowledge. In *IFAC World Congress*, 2002.

[86] R. Murray-Smith and T. Johansen. Local learning in local model networks. In *4th International Conference on Artificial Neural Networks*, 1995.

[87] H. Nakada, K. Takaba, and T. Katayama. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41 :905–913, 2005.

[88] A. Nazin, J. Roll, L. Ljung, and I. Grama. Direct weight optimization in statistical estimation and system identification. In *International Conference on System Identification and Control Problems, Moscow, Russia*, 2008.

[89] H. Oku and H. Kimura. Recursive 4SID algorithms using gradient type subspace tracking. *Automatica*, 38 :1035–1043, 2002.

[90] S. Paoletti, A. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems : A tutorial. *European Journal of Control*, 13 :242–260, 2007.

[91] S. Paoletti, J. Roll, A. Garulli, and A. Vicino. Input-output realization of piecewise affine state space models. In *Conference on Decision and Control, New Orleans LA, USA*, 2007.

[92] K. M. Pekpe and S. Lecoeuche. Online classification of switching models based on subspace framework. In *IFAC Conference on Analysis and Design of Hybrid Systems, Alghero, Sardinia*, 2006.

[93] K. M. Pekpe, G. Mourot, K. Gasso, and J. Ragot. Identification of switching systems using change detection technique in the subspace framework. In *Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

[94] M. Petreczky. *Realization theory of hybrid systems*. PhD thesis, Vrije Universiteit, Amsterdam, 2006.

[95] M. Petreczky. Realization theory for linear switched systems : a formal power series approach. *Systems and Control Letters*, 56 :588–595, 2007.

[96] M. Petreczky and R. Vidal. Realization theory of stochastic jump-markov linear systems. In *Conference on Decision and Control, New Orleans LA, USA*, 2007.

[97] P. Pucar and J. Sjöberg. On the hinge finding algorithm for hinging hyperplanes. *IEEE Transactions on Information Theory*, 44 :1310–1319, 1998.

[98] S. J. Qin. An overview of subspace identification. *Computers and Chemical Engineering*, 30 :1502–1513, 2006.

[99] J. Ragot, G. Mourot, and D. Maquin. Parameter estimation of switching piecewise linear systems. In *Conference on Decision and Control, Maui, Hawaii, USA*, 2003.

[100] J. Roll. *Local and piecewise affine approaches to system identification*. PhD thesis, Departement of Electrical Engineering, Linköping University, 2003.

[101] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40 :37–50, 2004.

[102] J. Roll, A. Nazin, and L. Ljung. Nonlinear system identification via direct weight optimization. *Automatica*, 41 :475–490, 2005.

[103] F. Rosenqvist and A. Karlstrom. Realisation and estimation of piecewise-linear output-error models. *Automatica*, 41 :545–551, 2005.

[104] N. V. Sahinidis. Baron branch and reduce optimization navigator. Technical report, University of Illinoisat Urbana-Champaign, Departement of Chemical Engineering, Urbana, USA http ://archimedes.scs.uiuc.edu/baron/manuse.pdf, 2000.

[105] B. Schölkopf and A. J. Smola. *Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

[106] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Non linear black box modeling in system identification : an unified overview. *Automatica*, 33 :1691–1724, 1997.

[107] A. Skeppstedt, L. Ljung, and M. Millnert. Construction of composite models from observed data. *International Journal of Control*, 55 :141–152, 1992.

[108] E. D. Sontag. Nonlinear regulation : The piecewise linear approach. *IEEE Transaction on Automatic Control*, 26 :346–357, 1981.

[109] P. Stoica and M. Jansson. MIMO system identification : State-space and subspace approximations versus transfer function and instrumental variables. *IEEE Transaction on Signal Processing*, 48 :3087–3099, 2000.

[110] W. Sun and Y.-X. Yuan. *Optimization theory and methods : Nonlinear programming*. Springer, 2006.

[111] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, 2002.

[112] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15 :116–132, 1985.

[113] R. Toth, F. Felici, P. S. C. Heuberger, and P. M. J. Van den Hof. Discrete time LPV I/O and state space representations : differences of behavior and pitfalls of interpolation. In *European Control Conference, Kos, Greece*, 2007.

[114] A. J. Van der Schaft and J. M. Schumacher. Complementarity modelling of hybrid systems. *IEEE Transactions on Automatic Control*, 43 :483–490, 1998.

[115] P. Van Overschee and B. De Moor. *Subspace identification for linear systems. theory, implementation, applications*. Kluwer Academic Publishers, 1996.

[116] V. Vapnik. *Satistical Learning theory*. New-York : Wiley, 1998.

[117] V. Verdult. *Nonlinear System Identification : A State-Space Approach*. PhD thesis, University of Twente, 2002.

[118] V. Verdult, L. Ljung, and M. Verhaegen. Identification of composite local linear state-space models using a projected gradient search. *International Journal Control*, 75 :1385–1398, 2002.

[119] V. Verdult and M. Verhaegen. Subspace identification of piecewise linear systems. In *Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

[120] M. Verhaegen. Subspace model identification part 3 : Analysis of the ordinary output-error state space model identification algorithm. *International Journal of Control*, 58 :555–586, 1993.

[121] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 30, no. 1 :61–74, 1994.

[122] M. Verhaegen and P. Dewilde. Subspace model identification part 1 : The output-error state space model identification class of algorithms. *International Journal of Control*, 56 :1187–1210, 1992.

[123] M. Verhaegen and V. Verdult. *Filtering and System Identification : A Least Squares Approach*. Cambridge University Press, 2007.

[124] R. Vidal. *Generalized Principal Component Analysis (GPCA) : an Algebraic Geometric Approach to Subspace Clustering and Motion Segmentation*. PhD thesis, University of California at Berkeley, 2003.

[125] R. Vidal. Identification of PWARX hybrid models with unknown and possibly different orders. In *American Control Conference, Boston MA, USA*, 2004.

[126] R. Vidal. Recursive identification of switched ARX systems. *Automatica*, 44 :2274–2287, 2008.

[127] R. Vidal and B. D. O. Anderson. Recursive identification of switched ARX hybrid models : exponential convergence and persistence of excitation. In *Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

[128] R. Vidal, A. Chiuso, and S. Soatto. Observability and identifiability of jump linear systems. In *Conference on Decision and Control, Las Vegas NV, 2002*, 2002.

[129] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 :1–15, 2005.

[130] R. Vidal, S. Soatto, and A. Chiuso. Applications of hybrid system identification in computer vision. In *European Control Conference, Kos, Greece*, 2007.

[131] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Conference on Decision and Control, Maui, Hawaii, USA*, 2003.

[132] S. Weiland, A. L. Juloski, and B. Vet. On the equivalence of switched affine models and switched ARX models. In *Conference on Decision and Control, San Diego CA, USA*, 2006.

[133] J. Yen, L. Wang, and C. Gillespie. Improving the interpretability of tsk fuzzy models by combining global learning and local learning. *IEEE Transactions on Fuzzy Systems*, 6 :530–537, 1998.

Résumé

Dans de nombreuses applications modernes, l'interaction de plus en plus importante entre les systèmes numériques (ordinateurs, logiciels, composants logiques, etc.) et les processus physiques (relations entre signaux continus) a conduit, en Automatique, à l'émergence et à la formalisation des systèmes dits hybrides. Formellement, les systèmes hybrides peuvent être définis comme des systèmes mixtes où interagissent des phénomènes de nature à la fois continue et événementielle. L'analyse et la conduite de tels systèmes comme de tout autre type de système dynamique nécessitent bien souvent que l'on dispose d'un modèle mathématique de ces systèmes. Ainsi, nous nous intéressons dans ce travail, à l'identification de systèmes hybrides linéaires à partir de mesures entrée-sortie. Après avoir fait le point sur les méthodes disponibles dans la littérature récente en relation avec ce sujet, nous mettons en évidence la nécessité de développer des méthodes d'identification de systèmes hybrides multivariables dans le contexte très délicat où ni le nombre de sous-modèles constitutifs du système hybride, ni les ordres de ces sous-modèles, ni leurs paramètres ne sont connus a priori. Nous considérons d'abord des modèles d'état à commutations. Pour estimer ces modèles par les méthodes des sous-espaces, il est indispensable de contrôler dans l'espace d'état, les bases de représentation des matrices de paramètres associées aux différents sous-modèles à estimer. Cela nous a conduit au développement de nouvelles techniques d'identification structurée de modèles linéaires d'état qui possèdent cette propriété. Nous généralisons ensuite les techniques ainsi développées à l'identification de systèmes multivariables commutants, représentés par des modèles d'état. Cependant, dans le cas général, l'identification de modèles d'état hybrides est limitée par de sévères problèmes de complexité numérique. De ce fait, nous étudions le cas particulier où les instants de commutation sont séparés par un certain temps de séjour minimum dans les différents modes du système. Afin de nous affranchir de cette contrainte, nous investiguons l'identification de modèles MIMO commutants de type Auto-Regressif à entrée eXogène (ARX). Nous généralisons alors la méthode algébro-géométrique (GPCA) à l'identification de systèmes multivariables, discutons quelques problèmes de complexité numérique et suggérons des alternatives. La dernière partie du travail est consacrée à la validation de nos méthodes sur des exemples de simulation ainsi que sur un procédé de montage automatique de composants électroniques sur circuit imprimé.

Abstract

In many modern applications, the increasing interaction between digital systems (computers, software, logical components, etc.) and continuous physical processes (relations between continuous signals) have given rise to the formalisation of the so-called hybrid systems. Formally, hybrid systems can be regarded as models that exhibit both continuous and discrete-event phenomena. The analysis and the control of such systems requires generally that one has a mathematical model of them. To this purpose, we consider in this work the identification of linear hybrid systems from input-output data. First, we briefly review the recent literature on the subject and then pointed out the need of developing new methods for hybrid multivariable systems identification in the very challenging context where neither the number of submodels of the hybrid system, nor the respective orders of these submodels, nor their parameters are known a priori. We first focus on the case of switched linear state space models. In order to estimate these models using subspace methods, it is necessary to control the state space basis in which the model matrices will be estimated. This aspect of the problem leads us to the development of new structured subspace identification methods for linear state space models. We then extend these methods to the estimation of switched linear state space models. However, in a general framework, the identification of such models is severely restricted by the issues of numerical complexity. Consequently, we turn to the particular case where the switching times are separated by a certain minimum dwell time. We then propose an algorithm that estimates online the orders and the parameters of the differents submodels. Finally, to escape the constraint of dwell time we consider the identification of MIMO switched ARX models. We hence generalized the GPCA algorithm to the identification to multivariable systems, discussed some issues of numerical complexity and suggested some alternatives.