Théorie de l'apprentissage et identification des systèmes dynamiques hybrides

Louis Massucci

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Théorie de l'apprentissage et identification des systèmes dynamiques hybrides

THÈSE

présentée et soutenue publiquement le 8 décembre 2022

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention Automatique, Traitement du signal et des images, Génie informatique)

par

Louis Massucci

Composition du jury

Présidente :	Marianne Clausel	Professeure, Université de Lorraine
Rapporteurs :	Massih-Reza Amini	Professeur, Université de Grenoble Alpes
	Laurent Bako	Maître de Conférence HDR, Université de Lyon
Examinateurs :	Marianne Clausel	Professeure, Université de Lorraine
	Mihaly Petreczky	Chargé de recherche, CNRS
Directeurs de thèse :	Marion Gilson	Professeure, Université de Lorraine
	Fabien Lauer	Maître de conférence HDR, Université de Lorraine

CRAN — UMR 7039, LORIA — UMR 7503

Mis en page avec la classe thesul.

Remerciements

La rédaction des remerciements est étonnement difficile par l'engagement que cela demande. Il se dit qu'un peu d'humour y est le bienvenu, mais je ne pense pas pouvoir m'y risquer, mon humour est trop complexe à identifier... Je dirai même que je ne suis pas en mesure, sur ce sujet, d'offrir des garanties finies. Il s'agit d'une théorie qui nécessite un apprentissage approfondi pour moi.

Je vais donc me concentrer sur l'essentiel, en commençant par remercier Marion et Fabien, mes directeurs de thèse, qui ont si bien su m'accompagner pendant ces 3 années. Je ne saurai pas trouver les mots suffisants pour témoigner de la reconnaissance que j'ai envers vous deux. Chacun à votre manière, vous avez su m'aider et me faire grandir autant professionnellement que personnellement. Merci à tous les deux.

Je tiens également à remercier tous les membres de mon jury pour avoir rapporté et examiné ma thèse. C'est un grand plaisir pour moi de pouvoir vous présenter mes travaux.

Mes remerciements vont également aux membres de l'équipe iModel, pour les nombreuses discussions professionnelles, ou non, qui m'ont permis de tenir le cap pendant la thèse. Pour la plupart, vous avez d'abord été mes enseignants d'école, et vous êtes maintenant d'anciens collègues dont je garderai de très bons souvenirs. Je remercie également le CRAN, le LORIA, la fédération Charles Hermite ainsi que la région Grand Est.

Je remercie chaleureusement mes amis, et tout particulièrement Axelle, Quentin, Loïc et Arthur, pour leur présence et leur soutien pendant ces 3 années.

Merci à ma famille de toujours être à mes côtés, de me suivre et d'avoir confiance en mes choix. Merci à ma Maman pour tous les petits plats que tu es venue me déposer, et merci à mon Papa pour avoir tenté, pendant 3 ans et sans relâche, de comprendre le sujet de mes recherches.

Enfin, merci à toi Pauline, ma moitié, de m'avoir soutenu pendant ces 3 années et de partager mon quotidien.

Je dédie cette thèse

à ma Oma.

qui a toujours veillé sur moi,

et qui, je sais, continue de le faire.

Table des matières

Glossaire

k : Indice de temps

j : Indice de mode

$u_k,\ y_k$: Entrée et sortie du système à l'instant k

e_k : Perturbations à l'instant k

ϵ_k : Erreur d'estimation à l'instant k

ν_k : Terme de bruit

q : Opérateur d'avance

$G(\mathrm{q}),\ H(\mathrm{q})$: Fonctions de transfert

$\boldsymbol{\theta}$: Vecteur de paramètres

$\boldsymbol{\Theta}$: Matrice de paramètres

$\boldsymbol{x}_k$: Vecteur de regression à l'instant k

$\mathbf{X}$: Matrice de régression

$\mathcal{D}$: Séquence de données

ℓ_p : Fonction de perte d'ordre p

f : Fonction de sortie, modèle

f_j : j-ème sous-modèle

$\mathcal{F}$: Classe de fonctions

$\boldsymbol{r}$: Séquence de commutation dans $[C]^N$

C : Nombre de sous-modèles

$Z = (X, Y)$: Couple Z de variables aléatoires $(X, Y) \in \mathcal{X} \times \mathcal{Y}$

$\mathcal{X}, \mathcal{Y}$: Espaces d'entrée et de sortie

N : Nombre de points de l'échantillon d'apprentissage

$L(f)$: Erreur de généralisation, risque

$\hat{L}_N(f)$: Erreur d'apprentissage, risque empirique

ϵ : Intervalle de confiance

δ : Indice de confiance, $\delta \in (0, 1)$

$\hat{\mathcal{R}}_{\mathbf{Z}_N}(\mathcal{F})$: Complexité de Rademacher empirique d'une classe de fonctions $\mathcal{F}$ étant donné l'échantillon $\mathbf{Z}_N$

σ_N : Séquence de N variables de Rademacher indépendantes

$\mathcal{L}$: Espace de fonctions de perte

M : Borne sur les sorties tronquées

Λ : Borne sur la complexité de la classe de fonctions $\mathcal{F}$

$\mathcal{E}$: Erreur d'apprentissage

$\Gamma(f)$: Régularisation

$\Omega(f)$: Vecteur de complexité de chacun des sous-modèles

λ : hyperparamètre de régularisation

$\beta(a)$: coefficient de mélange

a : Nombre de points par bloc

μ : Nombre de blocs

Table des figures

Liste des tableaux

Introduction

L'automatique s'intéresse à l'analyse et la commande des systèmes dynamiques. Pour pouvoir mener à bien l'analyse et la commande de ces systèmes, une première étape consiste à les modéliser, c'est-à-dire à créer un modèle mathématique caractérisant le fonctionnement d'un système. Dans la plupart des cas, et dès que les principes physiques en jeu sont méconnus ou trop complexes, la modélisation du système est réalisée à partir d'observations expérimentales de son comportement. L'identification des systèmes dynamiques est le champ de l'automatique qui se concentre sur l'estimation de modèles à partir de telles données. Un des points clés de cette démarche réside dans l'obtention de garanties sur la précision du modèle. La théorie de l'identification des systèmes dynamiques, en grande partie fondée sur la statistique paramétrique, permet d'obtenir des garanties asymptotiques sous des hypothèses assez contraignantes (spécification précise du bruit, du modèle et de la méthode d'estimation notamment). Ces garanties ne sont donc, la plupart du temps, pas adaptées pour quantifier précisément l'erreur d'un modèle estimé à partir d'un nombre fini d'observations.

Dans le domaine de l'intelligence artificielle, la construction de modèles de prédiction à partir de données expérimentales est étudiée dans le cadre de l'apprentissage machine. Ici aussi, garantir les performances des modèles estimés est une question centrale, étudiée notamment dans le cadre de la théorie statistique de l'apprentissage. À l'inverse de la statistique paramétrique classique, cette théorie permet d'obtenir des garanties dans un cadre beaucoup moins contraint (modèles non paramétriques, cadre agnostique sans *a priori* sur la forme du bruit...) et fournit en particulier des bornes non asymptotiques et uniformes sur l'erreur de prédiction des modèles estimés à partir d'un nombre fini d'observations. Cependant, la plupart des résultats de ce type sont établis sous des hypothèses d'indépendance des observations, tout à fait classiques dans de nombreux

contextes, mais non adaptées au cas de l'identification des systèmes dynamiques.

Cette thèse vise à combler le fossé entre ces deux disciplines : étendre la théorie de l'apprentissage au cas de données non indépendantes pour obtenir des garanties les plus précises possibles et applicables en pratique à l'identification des systèmes dynamiques. Dans ce contexte, on a fait le choix de s'intéresser plus particulièrement aux systèmes dynamiques hybrides (Liberzon, 2003). Ces systèmes mêlent des comportements continus et des événements discrets, impliquant une représentation à base de modèles commutant entre plusieurs modes de fonctionnement. On les retrouve dans de nombreux domaines tels que les systèmes électrotechniques, les réseaux de communication, les systèmes de transport, etc. Il existe 2 classes de systèmes hybrides : les systèmes affines par morceaux, dont l'état discret dépend de l'état du système, et les systèmes à commutation arbitraire, dont l'état discret est indépendant de l'état. Dans cette thèse, on s'intéresse à la classe des systèmes à commutations arbitraires. Il s'agit de systèmes dont le mécanisme de commutation entre différents modes se fait de manière arbitraire ou en fonction de variables non observées. Pour ces derniers, leur identification à partir de données entrée-sortie revient à un problème d'apprentissage qui consiste à estimer d'une part un ensemble de sous-modèles de régression associés aux différents modes de fonctionnement, et d'autre part l'affectation des points de données aux différents sous-modèles (Lauer and Bloch, 2019).

Dans ce contexte, un des apports majeurs de cette thèse est l'obtention de garanties statistiques sur l'erreur de généralisation des modèles à commutations. Ces bornes fondées sur les travaux de Yu (1994); Mohri and Rostamizadeh (2009), dont une version préliminaire est publiée dans [C3] [1], ont l'avantage d'être valides malgré les liens de dépendance entre les données entrée-sortie, d'être non asymptotiques, et d'être également valables pour différentes classes de modèles régularisés. Ces différentes formes de régularisation ont par ailleurs donné lieu à la proposition de nouveaux algorithmes d'optimisation adaptés à l'apprentissage de modèles à commutations conduisant à de bonnes performances en généralisation. Ces algorithmes sont présentés dans [C1].

Par ailleurs, en identification des systèmes hybrides, un des problèmes majeurs réside dans l'estimation du nombre de sous-modèles associés à chacun des modes de fonctionnement. Il s'agit d'un hyperparamètre à maîtriser pour l'identification de ces systèmes. Les bornes obtenues au

1. Les références au format lettre suivie d'un chiffre correspondent aux publications de l'auteur listées ci-dessous.

cours de cette thèse permettent de traiter ce problème, et de proposer une nouvelle méthode de sélection de modèle, à base de minimisation du risque structurel (Vapnik, 1998). Cette contribution est également introduite dans [C3], [R1]. Cette méthode d'estimation du nombre de modes est confrontée dans [C2] à deux autres méthodes existantes issues de la littérature (Vidal et al., 2003; Ozay et al., 2012), afin d'en mesurer les performances et de proposer un guide pratique pour déterminer la méthode la plus adaptée à une situation donnée.

Dans les deux premiers chapitres de la thèse, on présente le cadre général de l'identification des systèmes dynamiques, en s'inspirant notamment de Ljung (1999); Soderstrom and Stoica (1983); Schoukens and Pintelon (2014), ainsi que celui de la théorie de l'apprentissage, en suivant la présentation de Mohri et al. (2018). Dans le troisième chapitre, des bornes valides pour le cas de l'identification de systèmes hybrides sont développées, et la nouvelle méthode d'estimation du nombre de sous-modèles y est présentée, telle que publiée dans [R1]. Pour ces bornes, on considère différentes formes de régularisation, correspondant à des contraintes que l'on peut fixer sur la classe de modèles étudiée. Le chapitre quatre présente ensuite les nouveaux algorithmes d'apprentissage inspirés des différentes formes de régularisation considérées par ces bornes et proposés dans [C1]. Le cinquième chapitre présente les expériences numériques traitant de l'estimation du nombre de modes des systèmes à commutations arbitraires ainsi que de la comparaison de leur efficacité face aux méthodes algébriques en approfondissant les résultats déjà communiqués dans [C2]. Enfin, le sixième et dernier chapitre expose les conclusions de la thèse et ouvre la discussion sur les limites des méthodes proposées et les perspectives envisagées.

Liste des publications

Article de revue internationale

R1 Massucci, L., Lauer F., and Gilson, M. A Statistical Learning Perspective on Switched Linear System Identification. *Automatica*, Volume 142, 110532, 2022

Articles de conférences internationales avec actes

C1 Massucci, L., Lauer F., and Gilson, M. Regularized switched system identification : A statistical learning perspective. In : *Proceedings of the 7th IFAC Conference on Analysis*

and Design of Hybrid Systems (ADHS), Bruxelles, Belgium, Virtual Event, 2021

C2 Massucci, L., Lauer F., and Gilson, M. How Statistical Learning Can Help to Estimate the Number of Modes in Switched System Identification ? In : *Proceedings of the 19th IFAC Symposium on System Identification (SYSID)*, Padoue, Italy, Virtual Event, 2021

C3 Massucci, L., Lauer F., and Gilson, M. Structural risk minimization for switched system identification. In : *Proceedings of the 59th IEEE Conference on Decision and Control (CDC)*, Jeju Island, Republic of Korea, Virtual Event, 2020

Chapitre 1

Identification de systèmes dynamiques

Dans ce chapitre, l'objectif est de présenter le cadre général de l'identification des systèmes, ainsi que celui plus spécifique des systèmes hybrides. La présentation de ces notions se fait en suivant le cadre défini par Ljung (1999); Soderstrom and Stoica (1983); Schoukens and Pintelon (2014) pour les principes généraux, et celui défini par Lauer and Bloch (2019) pour les systèmes hybrides. Dans la Section 1.1, nous définissons la procédure générale d'identification des systèmes. Les Sections 1.2 et 1.3 décrivent les étapes internes de la procédure plus en détails. La Section 1.4 présente l'identification des systèmes non-linéaires. Finalement, la Section 1.5 introduit le problème d'identification des systèmes hybrides ainsi que les méthodes existantes.

1.1 Procédure d'identification

L'automatique s'intéresse à l'analyse et la commande des systèmes dynamiques, tels que l'évolution d'une réaction chimique au cours du temps, le fonctionnement d'un système électrique, la trajectoire d'un avion, etc. Pour pouvoir mener à bien l'analyse et la commande de ces systèmes, une première étape consiste à les modéliser, c'est-à-dire à créer un modèle mathématique caractérisant le fonctionnement d'un système. Si de nombreux systèmes pourront être modélisés par la physique, ce ne sera pas le cas pour d'autres dont les principes physiques sont méconnus ou trop complexes. La modélisation de ces systèmes se fait alors à partir d'observations expérimentales de leur comportement en suivant la procédure d'identification décrite dans la Table 1.1. Dans un premier temps, il faut alors recueillir un jeu de données, suffisamment riche, décrivant la

TABLE 1.1 – Procédure d'identification

Procédure d'identification
1. Choisir et enregistrer un jeu de données d'entrées-sorties (u_k, y_k)
2. Choisir une classe de modèle, ou une structure de modèle
3. Estimer le modèle au sein de sa classe, en fonction des données et du critère d'identification
4. Évaluer le modèle
5. Suivant la qualité du modèle, re-boucler sur les étapes 1,2 et/ou 3.

dynamique du système. Ensuite, il est nécessaire de définir sur quelle classe de modèles l'identification est réalisée, afin d'avoir une idée de la structure et du nombre paramètres à estimer. La troisième étape consiste alors à estimer le modèle à partir d'un critère d'identification choisi. Enfin, il faudra évaluer le modèle et éventuellement re-boucler sur l'une ou l'autre des étapes précédentes afin d'affiner le résultat.

Comme présenté dans Ljung (1999); Soderstrom and Stoica (1983); Schoukens and Pintelon (2014), pour identifier un système dynamique, on cherche à construire un modèle G_0, ainsi qu'un modèle de perturbation H_0, à partir des données d'entrée u et sortie y, où y est soumis à des perturbations non mesurables e. On s'intéresse, dans cette thèse, exclusivement au cas de l'identification en boucle ouverte. Étant données ces perturbations, ou "bruit", le modèle G_0 présente en sortie une erreur ϵ d'estimation.

Dans le cadre des systèmes linéaires définis par des modèles à temps discret, on définit le système $\mathcal{S}$ ayant généré les données par

$$\mathcal{S} : \quad y_k = G_0(\mathrm{q})u_k + H_0(\mathrm{q})e_k, \tag{1.1}$$

où q décrit un opérateur d'avance, tel que $\mathrm{q}u_k = u_{k+1}$ et $\mathrm{q}^{-1}u_k = u_{k-1}$.

On considère la version paramétrique des modèles $\mathcal{G}$ et $\mathcal{H}$:

$$\mathcal{G}, \mathcal{H} : \quad y_k(\boldsymbol{\theta}) = G(\mathrm{q}, \boldsymbol{\theta})u_k + H(\mathrm{q}, \boldsymbol{\theta})e_k, \tag{1.2}$$

avec de façon générale

$$G(\mathrm{q}, \boldsymbol{\theta}) = \frac{B(\mathrm{q}, \boldsymbol{\theta})}{A(\mathrm{q}, \boldsymbol{\theta})} = \frac{b_0 + b_1\mathrm{q}^{-1} + \cdots + b_{n_b}\mathrm{q}^{-n_b}}{1 + a_1\mathrm{q}^{-1} + \cdots + a_{n_a}\mathrm{q}^{-n_a}} \tag{1.3}$$

et

$$H(\mathrm{q},\boldsymbol{\theta}) = \frac{D(\mathrm{q},\boldsymbol{\theta})}{C(\mathrm{q},\boldsymbol{\theta})} = \frac{d_1\mathrm{q}^{-1} + \cdots + d_{n_d}\mathrm{q}^{-n_b}}{c_1\mathrm{q}^{-1} + \cdots + c_{n_c}\mathrm{q}^{-n_c}}, \tag{1.4}$$

où

$$\boldsymbol{\theta} = [a_1 \ \ldots \ a_{n_a} \ b_0 \ \ldots \ b_{n_b} \ c_1 \ \ldots \ c_{n_c} \ d_1 \ \ldots \ d_{n_d}]^T \tag{1.5}$$

représente un vecteur de paramètres de dimension $d = n_a + n_b + n_c + n_d$. Ici, la forme de $\boldsymbol{\theta}$ correspond à une structure de modèle particulière, les paramètres c_i et d_i dépendent de la structure choisie. Nous discutons des structures de modèle dans la section suivante.

1.2 Structure du modèle

Il existe des structures de modèles classiques qui permettent de décrire avec précision le comportement des systèmes dynamiques. Dans cette thèse, nous nous intéressons aux systèmes composés d'une seule entrée et d'une seule sortie (SISO), et invariants dans le temps.

Systèmes entrée/sortie

La réponse impulsionnelle d'un système entrée/sortie prend la forme

$$y_k = \sum_{i=1}^{N} \tilde{g}_i u_{k-i} + \nu_k, \tag{1.6}$$

et les perturbations ν_k agissant sur le système se notent

$$\nu_k = \sum_{i=1}^{N} \tilde{h}_i e_{k-i}, \tag{1.7}$$

avec $\tilde{g}_i$ et $\tilde{h}_i$ des paramètres du modèle, et e_k un bruit blanc gaussien, i.e., une séquence de variables aléatoires indépendantes et identiquement distribuées (i.i.d) selon une loi gaussienne, de moyenne nulle et de variance unitaire. On définit

$$y_k = G(\mathrm{q},\boldsymbol{\theta})u_k + H(\mathrm{q},\boldsymbol{\theta})e_k, \tag{1.8}$$

où $G(\mathrm{q},\boldsymbol{\theta}) = \sum_{i=1}^{n} \tilde{g}_i \mathrm{q}^{-i}$ est une fonction de transfert et $H(\mathrm{q},\boldsymbol{\theta}) = 1 + \sum_{i=1}^{n} \tilde{h}_i \mathrm{q}^{-i}$ est une fonction monique supposée stable et inversible : $e_k = H^{-1}(\mathrm{q},\boldsymbol{\theta})\nu_k$. Suivant la classe de modèles choisie, les séquences $\{\tilde{g}_i\}$ et $\{\tilde{h}_i\}$ prendront différentes formes que nous verrons par la suite. On redéfinit dans un premier temps l'erreur de prédiction.

L'erreur de prédiction

D'après (1.7), on a $\nu_k = \sum_{i=1}^{n} \tilde{h}_i e_{k-i} = H(\mathrm{q},\boldsymbol{\theta})e_k$. Pour une mesure ν_l, telle que $l \leq k-1$, on retrouve aussi e_l tel que $l \leq k-1$ en raison de la relation d'inversion entre ν_k et e_k. Comme la moyenne de e_k est nulle, on a

$$\hat{\nu}_{k|k-1} = (H(\mathrm{q},\boldsymbol{\theta}) - 1)e_k = (1 - H^{-1}(\mathrm{q},\boldsymbol{\theta}))\nu_k. \tag{1.9}$$

Ainsi, la prédiction du prochain pas de notre modèle défini en (1.8), sachant u_l, y_l et ν_l peut être écrite comme

$$\hat{y}_{k|k-1} = G(\mathrm{q},\hat{\boldsymbol{\theta}})u_k + (1 - H^{-1}(\mathrm{q},\hat{\boldsymbol{\theta}}))(y_k - G(\mathrm{q},\hat{\boldsymbol{\theta}})u_k), \tag{1.10}$$

ou

$$\hat{y}_{k|k-1} = H^{-1}(\mathrm{q},\hat{\boldsymbol{\theta}})G(\mathrm{q},\hat{\boldsymbol{\theta}})u_k + (1 - H^{-1}(\mathrm{q},\hat{\boldsymbol{\theta}}))y_k. \tag{1.11}$$

Les fonctions de transfert $G(\mathrm{q},\boldsymbol{\theta})$ et $H(\mathrm{q},\boldsymbol{\theta})$ seront représentées par des fonctions rationnelles dont les coefficients aux numérateur et dénominateur correspondent aux paramètres à identifier. Ces représentations mènent à des structures de modèles variées, un certain nombre d'entre elles sont brièvement présentées ci-dessous.

La structure de modèle

Il existe différentes structures de modèle. Nous présentons certaines d'entre elles dans la Table 1.2, où $A(\mathrm{q},\boldsymbol{\theta})$, $B(\mathrm{q},\boldsymbol{\theta})$, $C(\mathrm{q},\boldsymbol{\theta})$, $D(\mathrm{q},\boldsymbol{\theta})$, $F(\mathrm{q},\boldsymbol{\theta})$ correspondent à des polynômes en q^{-1} respectivement de degré $n_a, n_b, \ldots, n_f$, avec $A(\mathrm{q},\boldsymbol{\theta})$, $C(\mathrm{q},\boldsymbol{\theta})$, $D(\mathrm{q},\boldsymbol{\theta})$, $F(\mathrm{q},\boldsymbol{\theta})$ des fonctions moniques, c'est-à-dire des polynômes non nuls dont le coefficient dominant est fixé à 1.

TABLE 1.2 – Structures de modèles

Modèle	$G(\mathrm{q},\boldsymbol{\theta})$	$H(\mathrm{q},\boldsymbol{\theta})$
ARX	$\frac{B(\mathrm{q},\boldsymbol{\theta})}{A(\mathrm{q},\boldsymbol{\theta})}$	$\frac{1}{A(\mathrm{q},\boldsymbol{\theta})}$
ARMAX	$\frac{B(\mathrm{q},\boldsymbol{\theta})}{A(\mathrm{q},\boldsymbol{\theta})}$	$\frac{C(\mathrm{q},\boldsymbol{\theta})}{A(\mathrm{q},\boldsymbol{\theta})}$
OE	$\frac{B(\mathrm{q},\boldsymbol{\theta})}{F(\mathrm{q},\boldsymbol{\theta})}$	1
FIR	$B(\mathrm{q},\boldsymbol{\theta})$	1
BJ	$\frac{B(\mathrm{q},\boldsymbol{\theta})}{F(\mathrm{q},\boldsymbol{\theta})}$	$\frac{D(\mathrm{q},\boldsymbol{\theta})}{C(\mathrm{q},\boldsymbol{\theta})}$

Cette thèse se concentrera sur les modèles ARX, pour *AutoRegressive with eXternal input*. Un modèle ARX est défini par 2 polynômes :

$$A(\mathrm{q},\boldsymbol{\theta}) = 1 + a_1\mathrm{q}^{-1} + \cdots + a_{n_a}\mathrm{q}^{-n_a}$$
$$B(\mathrm{q},\boldsymbol{\theta}) = b_1\mathrm{q}^{-1} + \cdots + b_{n_b}\mathrm{q}^{-n_b}$$

avec

$$\boldsymbol{\theta} = [a_1 \dots a_{n_a}\ b_1 \dots b_{n_b}]^T \tag{1.12}$$

tel que

$$G(\mathrm{q},\boldsymbol{\theta}) = \frac{B(\mathrm{q},\boldsymbol{\theta})}{A(\mathrm{q},\boldsymbol{\theta})} \quad \text{et} \quad H(\mathrm{q},\boldsymbol{\theta}) = \frac{1}{A(\mathrm{q},\boldsymbol{\theta})}. \tag{1.13}$$

Avec n_a et n_b le degré des polynômes $A(\mathrm{q},\boldsymbol{\theta})$ et $B(\mathrm{q},\boldsymbol{\theta})$, on obtient donc un modèle linéaire en les paramètres de la forme suivante :

$$y_k = f(\boldsymbol{x}_k,\boldsymbol{\theta}) = (1 - A(\mathrm{q},\boldsymbol{\theta}))y_k + B(\mathrm{q},\boldsymbol{\theta})u_k = \boldsymbol{x}_k^T\boldsymbol{\theta} \tag{1.14}$$

avec le vecteur des régresseurs

$$\boldsymbol{x}_k = [-y_{k-1} \cdots - y_{k-n_a}\ u_{k-1} \dots u_{k-n_b}]^T. \tag{1.15}$$

L'acronyme ARX signifie donc AR pour la partie AutoRegressive, $A(\mathrm{q},\boldsymbol{\theta})y_k$, du modèle tandis

que X renvoie à l'entrée externe, $B(\mathrm{q}, \boldsymbol{\theta})u_k$. La structure du modèle ARX se fonde ainsi sur la régression linéaire (1.14) et de ce fait est connue pour être simple d'utilisation pour l'identification des systèmes dynamiques, tout en restant suffisament générique.

1.3 Régression linéaire

Une fois la classe de modèle choisie et l'ordre du modèle fixé, la troisième étape de la procédure d'identification consiste à estimer les paramètres de notre modèle. On considère tout d'abord un jeu de données

$$\mathcal{D} = \{(\boldsymbol{x}_k, y_k)\}_{k=1}^{N}, \tag{1.16}$$

où $\boldsymbol{x}_k$ est construit à partir des données d'entrée-sortie comme en (1.15) pour les modèles ARX. Ensuite, sachant que l'erreur de prédiction correspond à la différence entre la sortie du système et la prédiction du modèle $\hat{y}_k = f(\boldsymbol{x}_k, \boldsymbol{\theta})$, l'approche la plus commune pour estimer les paramètres est la méthode de l'erreur de prédiction (PEM pour *Prediction Error Method*), qui consiste à résoudre

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \frac{1}{N} \sum_{k=1}^{N} \ell(\epsilon_k), \tag{1.17}$$

où $\ell(\epsilon_k)$ est une fonction de perte qui mesure l'erreur d'estimation et prend comme argument l'erreur de prédiction

$$\epsilon_k(\boldsymbol{\theta}) = y_k + a_1 y_{k-1} + \cdots + a_{n_a} y_{k-n_a} - b_1 u_{k-1} - \cdots - b_{n_b} u_{k-n_b}. \tag{1.18}$$

Dans cette thèse, on se concentrera sur les pertes ℓ_p,

$$\ell_p(\epsilon_k) = |\epsilon_k|^p, \quad p \in \{1, 2\}. \tag{1.19}$$

Celles-ci incluent la mesure de l'erreur la plus communément utilisée, i.e., l'erreur quadratique (obtenue pour $p = 2$)

$$\ell_2(\epsilon) = \epsilon^2. \tag{1.20}$$

Un autre exemple de mesure de l'erreur couramment utilisée est l'erreur absolue (obtenue avec $p = 1$)

$$\ell_1(\epsilon) = |\epsilon|. \tag{1.21}$$

Ainsi, considérant l'erreur quadratique ($p = 2$), l'estimation des paramètres $\boldsymbol{\theta}$ d'un modèle ARX revient à calculer

$$\hat{\boldsymbol{\theta}} = \underset{\hat{\boldsymbol{\theta}} \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{N} \sum_{k=1}^{N} \epsilon_k^2(\boldsymbol{\theta}) \tag{1.22}$$

$$= \underset{\hat{\boldsymbol{\theta}} \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{N} \sum_{k=1}^{N} (y_k - \boldsymbol{x}_k^T \boldsymbol{\theta})^2, \tag{1.23}$$

dont la solution explicite prend la forme

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \boldsymbol{y}, \tag{1.24}$$

où $\boldsymbol{X} = [\boldsymbol{x}_1 \dots \boldsymbol{x}_N]^T$ est la matrice de régression avec $\boldsymbol{x}_k$ correspondant à (1.15) et $\boldsymbol{y} = [y_1 \dots y_N]^T$ est le vecteur des sorties mesurées.

1.4 Systèmes non linéaires

Pour le cas des systèmes non linéaires, deux sous-classes se distinguent : les modèles paramétriques et non paramétriques. Dans un cadre général, on les définit sous la forme

$$y_k = f(\boldsymbol{x}_k), \tag{1.25}$$

avec f une fonction non linéaire.

— Les modèles paramétriques sont des modèles à structure fixe, le nombre de paramètres est donc prédéterminé et fini. Ici, les valeurs des paramètres peuvent parfois avoir une signification physique.

— Les modèles non paramétriques sont des modèles dont il faut estimer à la fois la structure et les paramètres. Ces types de modèles sont particulièrement efficaces, de part leur

flexibilité, pour estimer des systèmes présentant des non-linéarités inconnues.

Un exemple de classe de modèles non paramétriques peut être la suivante :

$$\mathcal{F} = \left\{ f \in \mathbb{R}^{\mathcal{X}} \ : \ f : \sum_{i=1}^{M} \alpha_i f_i, \ \alpha_i \in \mathbb{R}, M \in \mathbb{N} \right\}. \tag{1.26}$$

Ici, M définit la structure du modèle et les termes α_i sont des poids attribués à chaque fonction de base f_i, qui peuvent par exemple être des fonctions à base radiale, ou encore des fonctions de noyaux.

Ce type de modélisation ne permet pas d'acquérir d'informations précises sur le système en lui-même. En effet, les poids α_i n'ont pas de lien direct avec les paramètres internes et physiques du système. Les modèles non paramétriques ont avant tout pour objectif de pouvoir réaliser des prédictions de la sortie y_k à partir de $\boldsymbol{x}_k$.

Dans cette thèse, nous nous intéressons plus particulièrement à une classe spécifique des systèmes linéaires. Plus précisément, on s'intéresse à l'identification d'un ensemble de sous-modèles linéaires qui composent un système hybride. Ce type de systèmes est présenté dans la section suivante.

1.5 Identification de systèmes hybrides

Dans cette partie, le cadre général de l'identification des systèmes hybrides est présenté. On présente notamment les différents types de méthodes pour identifier ces systèmes.

1.5.1 Modèles

Les systèmes hybrides possèdent l'aptitude de changer de comportement dynamique, ou de mode de fonctionnement. On considère alors qu'un système hybride est un système composé de plusieurs sous-systèmes, chacun décrivant sa propre dynamique. Le système hybride commute d'un mode actif à un autre, de manière arbitraire ou non. Ces modèles peuvent alors prendre différentes formes en fonction des mécanismes de commutation. Ces changements de mode sont des commutations "dures", ce qui différencie les systèmes hybrides des modèles de mélange par exemple. De plus, il existe différents formalismes pour présenter les systèmes hybrides, celui

présenté dans Liberzon (2003) en est un, où un système à commutation correspond à ce que nous définissons ici comme étant un système hybride.

On définit un système hybride par son équation d'entrée-sortie

$$y_k = f_{r_k}(\boldsymbol{x}_k) + \nu_k, \tag{1.27}$$

où $y_k \in \mathbb{R}$ est la sortie, $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}$ correspond au vecteur de regression, typiquement de forme ARX comme en (1.15), et $r_k \in \{1, \dots, C\}$ correspond à l'état discret ou mode, avec C le nombre de sous-modèles. Comme précédemment, ν_k est le terme de bruit.

À l'instant k, le système se situe dans un mode particulier j, tel que $r_k = j$, et le sous-modèle f_j est alors actif. Ainsi, à chaque instant k, un seul sous-modèle est actif, en fonction de la séquence de commutation $(r_k)_{1 \leq k \leq N}$.

Un système hybride composé de sous-modèles ARX linéaires est défini comme suit :

$$y_k = \boldsymbol{x}_k^T \boldsymbol{\theta}_{r_k} + \nu_k, \tag{1.28}$$

avec le régresseur $\boldsymbol{x}_k$ et les vecteurs paramètres $\boldsymbol{\theta}_j$, $j = 1, \dots, C$, où $\boldsymbol{\theta}_j$ garde la forme (1.5), à la seule différence qu'il y a maintenant C différents vecteurs de paramètres, correspondant aux C sous-modèles. Le prédicteur associé est donc

$$\hat{y}_k = \boldsymbol{x}_k^T \hat{\boldsymbol{\theta}}_{r_k}. \tag{1.29}$$

La forme ARX est la plus utilisée dans la littérature sur l'identification de systèmes hybrides et nous nous limiterons donc à son étude dans cette thèse.

On distingue deux classes de systèmes hybrides linéaires :

— les systèmes affines par morceaux (*PieceWise Affine systems*), dont l'état discret r_k dépend de l'état du système $\boldsymbol{x}_k$;

— les systèmes à commutation arbitraire, dont l'état discret est indépendant de $\boldsymbol{x}_k$. Ces modèles portent la dénomination de systèmes SARX pour *Switched ARX systems*.

C'est à cette dernière catégorie de systèmes qu'on s'intéressera dans les prochains chapitres.

1.5.2 Méthodes d'identification

L'objectif de l'identification des systèmes hybrides est d'obtenir un modèle $\boldsymbol{f} = \{f_j\}_{j=1}^{C}$ composé de C différents sous-modèles f_j, qui estiment le comportement continu du système pour chacun de ses modes. En introduisant la notation $[C]$ pour l'ensemble des entiers de 1 à C, le problème d'identification peut alors être formulé comme suit.

Problème 1 *À partir d'un jeu de données $\mathcal{D} = \{(\boldsymbol{x}_k, y_k)\}_{k=1}^{N}$, généré par un système hybride à commutation arbitraire, estimer le nombre C de sous-modèles, les différents vecteurs paramètres de chacun des sous-modèles $\boldsymbol{\theta}_j$, $j \in [C]$, ainsi que la séquence de commutation $\boldsymbol{r} = \{r_k\}_{k=1}^{N}$.*

En d'autres termes, pour un nombre de modes fixé, en introduisant $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \dots \boldsymbol{\theta}_C]$, on cherche à minimiser la moyenne de l'erreur de prédiction

$$J^{ME}(\boldsymbol{\Theta}) = \frac{1}{N} \sum_{k=1}^{N} \ell_{\min}(\boldsymbol{x}_k, y_k, \boldsymbol{\Theta}), \tag{1.30}$$

mesurée par la fonction de perte

$$\ell_{\min}(\boldsymbol{x}_k, y_k, \boldsymbol{\Theta}) = \min_{j \in [C]} \ell_p(y_k - \boldsymbol{x}_k^T \boldsymbol{\theta}_j), \tag{1.31}$$

qui retourne la plus petite erreur sur l'ensemble des sous-modèles, et permet ainsi d'affecter chaque point au sous-modèle qui lui est le plus proche. Le problème (1.30) est non convexe et rend l'obtention d'une solution globale très difficile (Lauer, 2016). De plus, bien que continu, le problème est également non lisse, ce qui nuit à la convergence de nombreux algorithmes d'optimisation locale que l'on pourrait appliquer.

Au cours des 20 dernières années, le domaine de l'identification des systèmes à commutations arbitraire a vu l'émergence de deux catégories de méthodes pour tenter de résoudre (1.30)–(1.31). On distingue alors :

— des méthodes qui estiment les sous-modèles à partir d'un nombre de modes fixé *a priori*, comme Vidal et al. (2003); Lauer et al. (2011); Lauer (2013); Ozay et al. (2015) ;

— et d'autres méthodes qui identifient les sous-modèles les uns après les autres en imposant une borne sur l'erreur de prédiction, telles que Bemporad et al. (2005); Bako (2011); Ozay

et al. (2012); Ohlsson and Ljung (2013).

Ci-dessous, on détaille certaines méthodes d'estimation à nombre de modes fixé, puis une présentation générale de la seconde catégorie de méthodes sera également donnée.

Méthodes algébriques pour l'estimation à nombre de modes fixé

La méthode algébrique est une méthode d'identification où l'on fait également l'hypothèse que le nombre de modes est fixé. Cette méthode a initialement été développée par Vidal et al. (2003) sous l'hypothèse que les données ne sont pas bruitées.

Le cœur de la méthode repose sur la résolution du système d'équations implémentant la contrainte de découplement hybride définie par :

$$\forall k \in \{1, \dots N\}, \quad \prod_{j=1}^{C} (y_k - \boldsymbol{x}_k^T \boldsymbol{\theta}_j) = 0. \tag{1.32}$$

À chaque instant k, un mode particulier $j = r_k$ est actif. Étant donné que les données sont supposées sans bruit, l'erreur de prédiction $y_k - \boldsymbol{x}_k^T \boldsymbol{\theta}_j$ de ce sous-modèle peut être nulle si $\boldsymbol{\theta}_j$ correspond aux vrais paramètres du système. Dans ce cas, le produit des erreurs de chaque sous-modèle est aussi nul quel que soit le mode actif r_k. Cette méthode d'identification permet de distinguer l'estimation des différents vecteurs paramètres $\boldsymbol{\theta}_j$ de celle de la séquence de commutation $\mathbf{r}$. Plus précisément, on peut résoudre le système polynomial (1.32) pour estimer les vecteurs paramètres $\boldsymbol{\theta}_j$ sans pour autant connaître le mode actif à chaque instant.

Le système d'équation (1.32) peut être réécrit sous la forme d'un polynôme homogène de degré C en $\boldsymbol{z}_k = [y_k \ \ -\boldsymbol{x}_k]^T$:

$$\forall k \in \{1, \dots N\}, \quad \prod_{j=1}^{C} (\boldsymbol{\beta}_j^T \boldsymbol{z}_k) = 0, \tag{1.33}$$

avec $\boldsymbol{\beta}_j = [1 \ \boldsymbol{\theta}_j^T]^T$. Chaque polynôme peut être réécrit sous la forme d'une combinaison linéaire de tous les monomes, au nombre de $M_C(d+1) = \binom{C+d}{d}$, $d = dim(\boldsymbol{x}_k) + 1 = n_a + n_b + 1$:

$$P_C(\boldsymbol{z}) = \prod_{j=1}^{C} (\boldsymbol{\beta}_j^T \boldsymbol{z}) = \boldsymbol{h}^T \boldsymbol{\nu}_C(\boldsymbol{z}). \tag{1.34}$$

On a ainsi un vecteur $\boldsymbol{h} \in \mathbb{R}^{M_C(d+1)}$ de coefficients dépendant des différents $\boldsymbol{\beta}_j$ et $\boldsymbol{\nu}_C(\boldsymbol{z})$ une projection de Veronese qui regroupe tous les monômes de degré C de $\boldsymbol{z}$.

Ainsi, $\boldsymbol{h}$ peut être estimé en résolvant (1.32)–(1.34), ce qui revient à résoudre le système d'équations linéaires

$$\begin{bmatrix} \boldsymbol{\nu}_C(\boldsymbol{z}_1)^T \\ \boldsymbol{\nu}_C(\boldsymbol{z}_2)^T \\ \vdots \\ \boldsymbol{\nu}_C(\boldsymbol{z}_N)^T \end{bmatrix} \boldsymbol{h} = \boldsymbol{L}_C \boldsymbol{h} = \boldsymbol{0}. \tag{1.35}$$

Dans un second temps, les paramètres $\boldsymbol{\beta}_j$ et $\boldsymbol{\theta}_j$ sont ensuite déterminés à partir de $\boldsymbol{h}$ et de la dérivation de $P_C(\boldsymbol{z})$

$$\nabla P_C(\boldsymbol{z}) = \frac{\partial P_C(\boldsymbol{z})}{\partial \boldsymbol{z}} = \sum_{j=1}^{C} \left(\prod_{l \neq j} \boldsymbol{\beta}_l^T \boldsymbol{z} \right) \boldsymbol{\beta}_j. \tag{1.36}$$

Pour un point $\boldsymbol{z}_{j_0}^*$ appartenant à un mode j_0, le produit $\prod_{l \neq j} \boldsymbol{\beta}_l^T \boldsymbol{z}$ dans le gradient (1.36) s'annule pour tous les $j \neq j_0$, on obtient alors

$$\nabla P_C(\boldsymbol{z}_{j_0}^*) = \left(\prod_{l \neq j_0} \boldsymbol{\beta}_l^T \boldsymbol{z}_{j_0}^* \right) \boldsymbol{\beta}_{j_0}, \tag{1.37}$$

représentant le seul terme de la somme restant. On peut donc retrouver $\boldsymbol{\beta}_{j_0}$ en posant

$$\boldsymbol{\beta}_{j_0} = \frac{\nabla P_C(\boldsymbol{z}_{j_0}^*)}{\prod_{l \neq j_0} \boldsymbol{\beta}_l^T \boldsymbol{z}_{j_0}^*}, \tag{1.38}$$

or, la première composante de $\boldsymbol{\beta}_j$ vaut 1, donc le produit $\prod_{l \neq j_0} \boldsymbol{\beta}_l^T \boldsymbol{z}_{j_0}^*$ est récupéré dans la première composante du gradient (1.37) que l'on note $\left(\nabla P_C(\boldsymbol{z}_{j_0}^*) \right)_1$. Ainsi, on a

$$\boldsymbol{\beta}_{j_0} = \frac{\nabla P_C(\boldsymbol{z}_{j_0}^*)}{\left(\nabla P_C(\boldsymbol{z}_{j_0}^*) \right)_1}. \tag{1.39}$$

Afin de déterminer les C vecteurs de paramètres avec (1.39), un point $\boldsymbol{z}_{j_0}^*$ doit être identifié pour chacun des modes. Ces points peuvent être choisis aux intersections entre une droite $\{\boldsymbol{z} : \boldsymbol{z} = \boldsymbol{z}_0 + \alpha \boldsymbol{z}', \alpha \in \mathbb{R}\}$ et les hyperplans $\{\boldsymbol{z} : \boldsymbol{\beta}_j^T \boldsymbol{z} = 0\}, j \in [C]$.

Le fait de contraindre les points à se situer sur une droite nous permet de les exprimer à l'aide

d'une seule variable α dont les valeurs peuvent être déterminées en résolvant $P_C(\boldsymbol{z}) = P_C(\boldsymbol{z}_0 + \alpha\boldsymbol{z}') = 0$, décrivant l'intersection avec les hyperplans. Plus précisément, pour tout $\boldsymbol{z}_0 \neq \boldsymbol{0}$ et $\boldsymbol{z}' \neq \gamma\boldsymbol{z}_0$ tel que $\boldsymbol{\beta}_j^T\boldsymbol{z}' \neq 0, j \in [C]$, pour que la droite soit en intersection avec tous les hyperplans, les points d'intérêt sont donnés par $\boldsymbol{z}_{j_0}^* = \boldsymbol{z}_0 + \alpha_{j_0}\boldsymbol{z}'$, où les α_{j_0} sont les C racines du polynôme $P_C(\boldsymbol{z}_0 + \alpha\boldsymbol{z}')$ en α.

Finalement, il reste à estimer la séquence de commutation telle que

$$\hat{r}_k = \underset{j\in[C]}{\operatorname{argmin}}(\boldsymbol{\beta}_j^T\boldsymbol{z}_k)^2, \tag{1.40}$$

ce qui coïncide avec

$$\hat{r}_k = j \quad \text{si et seulement si} \quad \boldsymbol{\beta}_j^T\boldsymbol{z}_k = 0 \tag{1.41}$$

dans le cas non bruité.

Ainsi, la méthode algébrique produit la solution exacte dans le cas où la contrainte de découplement hybride est respectée, c'est-à-dire sans bruit. La complexité algorithmique de cette approche est dominée par le calcul d'une solution du système homogène (1.35), i.e., un vecteur $\boldsymbol{h}$ non nul dans le noyau de $\boldsymbol{L}_C$. En présence de bruit, le système $\boldsymbol{L}_C\boldsymbol{h} = \boldsymbol{0}$ n'a pas de solution $\boldsymbol{h} \neq \boldsymbol{0}$ et est uniquement résolu au sens des moindres carrés, i.e.,

$$\min_{\|\boldsymbol{h}\|_2=1} \|\boldsymbol{L}_C\boldsymbol{h}\|_2, \tag{1.42}$$

via la décomposition en valeurs singulières de $\boldsymbol{L}_C$. Ceci rend la méthode particulièrement efficace en termes de calculs par rapport au nombre de données N. Néanmoins, trouver les racines α devient très complexe dès lors que le degré du polynôme, i.e., le nombre de modes, est supérieur à 4. De plus, la complexité de la méthode augmente également en fonction du nombre de monômes $M_C(d+1)$ et donc directement en fonction du nombre de modes et de la dimension du système.

Extension au cas bruité

La méthode algébrique, originellement proposée par Vidal et al. (2003), est une méthode d'estimation reposant sur l'hypothèse que les données sont non bruitées. Plus précisément, la méthode ne présente des résultats acceptables que si le niveau de bruit est modéré (Ma and

Vidal, 2005). La méthode a été étendue dans les travaux de Ozay et al. (2015) afin de surpasser cette limite. Le principe consiste à résoudre la contrainte de découplement hybride définie en (1.32) en y incluant cette fois-ci une variable de bruit η_k :

$$P_C(\boldsymbol{z}_k, \eta_k) = \prod_{j=1}^{C} (\boldsymbol{\beta}_j^T \tilde{\boldsymbol{z}}_k) = \boldsymbol{h}^T \boldsymbol{\nu}_C(\tilde{\boldsymbol{z}}_k) = 0, \tag{1.43}$$

avec $\tilde{\boldsymbol{z}}_k = [y_k + \eta_k, -\boldsymbol{x}_k^T]^T$.

Ainsi, de la même manière que pour le cas non bruité, on définit la matrice

$$\tilde{\boldsymbol{L}}_C(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{\nu}_C(\tilde{\boldsymbol{z}}_1)^T \\ \boldsymbol{\nu}_C(\tilde{\boldsymbol{z}}_2)^T \\ \vdots \\ \boldsymbol{\nu}_C(\tilde{\boldsymbol{z}}_N)^T \end{bmatrix}, \tag{1.44}$$

avec la séquence de bruit $\boldsymbol{\eta} = (\eta_k)_{1 \leq k \leq N}$.

Ici, le problème consiste d'abord à estimer la séquence de bruit $\boldsymbol{\eta}$, puis d'estimer $\boldsymbol{h}$ à l'aide de (1.35), où $\tilde{\boldsymbol{L}}_C(\boldsymbol{\eta})$ prend la place de $\boldsymbol{L}_C$. Sous l'hypothèse que le bruit est borné, la contrainte $|\eta_k| < \tau$ est imposée pour identifier la séquence de bruit $\boldsymbol{\eta}$. Obtenir un modèle qui interprète correctement les données en présence de bruit revient à chercher une séquence de bruit $\boldsymbol{\eta}$ qui retourne une matrice $\tilde{\boldsymbol{L}}_C(\boldsymbol{\eta})$ de rang déficient, permettant l'estimation du vecteur $\boldsymbol{h}$ dans le noyau de $\tilde{\boldsymbol{L}}_C(\boldsymbol{\eta})$. D'après Ozay et al. (2015), cela revient à résoudre le problème de minimisation de rang suivant :

$$\min_{\boldsymbol{\eta} \in \mathbb{R}^N} \operatorname{rank}(\tilde{\boldsymbol{L}}_C(\boldsymbol{\eta})), \quad \text{s.t. } |\eta_k| \leq \tau, \ k = 1, \ldots, N. \tag{1.45}$$

Pour résoudre (1.45), la matrice $\tilde{\boldsymbol{L}}_C(\boldsymbol{\eta})$, qui est polynomiale en $\boldsymbol{\eta}$, est d'abord linéarisée avec la méthode des moments de Lasserre (2001). Ensuite, une heuristique convexe standard basée sur la norme nucléaire (Fazel et al., 2003) est utilisée pour minimiser le rang.

Estimation à nombre de modes fixé avec k-LinReg

Initialement proposée par Lauer (2013), k-LinReg est une méthode d'estimation de systèmes à commutations arbitraires capable de traiter efficacement un grand nombre de données bruitées ou non. On définit l'Algorithme 1 développé ci-après avec $\mathbf{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]^T$ la matrice de régression, et $\boldsymbol{y} = [y_1, \dots, y_N]^T$ le vecteur des sorties désirées. Cette méthode cherche à résoudre les équations (1.30)–(1.31) en alternant entre la classification du jeu de données, afin d'affecter chaque point au sous-modèle qui lui est le plus proche, et la régression sur chacun de ces sous-modèles.

Algorithm 1: k-LinReg

Entrées: le jeu de données $(\mathbf{X}, \boldsymbol{y}) \in \mathbb{R}^{N \times d} \times \mathbb{R}^N$, le nombre de modes C, et une matrice de paramètres initiale $\boldsymbol{\Theta}^0 = [\boldsymbol{\theta}_1^0 \dots \boldsymbol{\theta}_C^0] \in \mathbb{R}^{d \times C}$.

Initialiser $i \leftarrow 0$

Répéter

Classer les points et estimer le mode de fonctionnement

$$r_k^i = \underset{j \in [C]}{\operatorname{argmin}} (y_k - \boldsymbol{x}_k^T \boldsymbol{\theta}_j^i)^2, \quad k = 1, \dots, N$$

Pour $j = 1$ *à* C **Faire**

Si $|\{k : r_k^i = j\}| < d$ **Alors**

Retourner $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}^i$ *et* $J^{ME}(\hat{\boldsymbol{\Theta}})$.

Construire la matrice $\mathbf{X}_j^i$, contenant toutes les k-ème lignes de $\mathbf{X}$ pour lesquelles $r_k^i = j$, ainsi que le vecteur $\boldsymbol{y}_j^i$ correspondant.

Mettre à jour le vecteur de paramètres du j-ème mode avec

$$\boldsymbol{\theta}_j^{i+1} = \underset{\boldsymbol{\theta}_j \in \mathbb{R}^d}{\operatorname{argmin}} \|\boldsymbol{y}_j^i - \mathbf{X}_j^i \boldsymbol{\theta}_j\|_2^2, \tag{1.46}$$

sachant que résoudre (1.46) revient à calculer (1.24), i.e.,

$$\boldsymbol{\theta}_j^{i+1} = (\boldsymbol{X}_j^{iT} \boldsymbol{X}_j^i)^{-1} \boldsymbol{X}_j^{iT} \boldsymbol{y}_j^i. \tag{1.47}$$

$i \leftarrow i + 1$

Jusqu'à *Convergence, e.g, lorsque*

$$\|\boldsymbol{\Theta}^i - \boldsymbol{\Theta}^{i-1}\|_2 \leq \tau,$$

ou lorsque la classification n'opère plus de changement;

Sorties: $\hat{\boldsymbol{\Theta}}^{i+1}$, $J^{ME}(\hat{\boldsymbol{\Theta}})$

On observe que lorsque le nombre de points assigné à un mode est trop faible, i.e., lorsque $|\{k : r_k^i = j\}| < d$, alors l'Algorithme 1 s'arrête et retourne le dernier modèle estimé. Cela signifie

généralement que l'algorithme a convergé vers une solution non satisfaisante. L'Algorithme 1 peut être interprété comme l'application d'un algorithme de descente par blocs, où l'on optimise de manière alternée entre la répartition des données au sein des sous-modèles et les paramètres des sous-modèles. Ces deux étapes étant très légères en termes de calculs, cela permet à l'algorithme d'être utilisé pour un grand nombre de données.

Cette méthode tend à converger vers un minimum local, ce qui n'assure pas la précision du modèle. Cependant, k-LinReg est une méthode d'identification relativement simple à mettre en place, et qui propose des résultats très satisfaisants en pratique. En effet, il est possible de relancer l'algorithme avec des initialisations différentes pour finalement ne retenir que le meilleur résultat. Ainsi, avec un nombre suffisant d'initialisations, on peut espérer s'approcher du minimum global, comme montré expérimentalement dans Lauer (2013). L'algorithme k-LinReg sera à la base des algorithmes régularisés présentés dans le Chapitre 4.

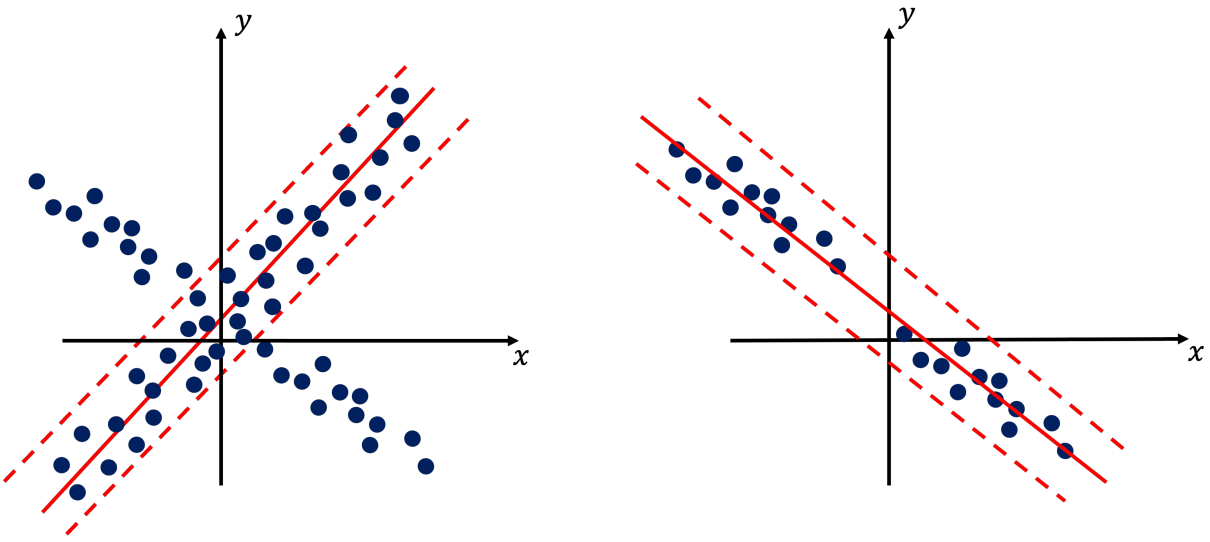

FIGURE 1.1 – Illustration de la méthode à erreur bornée. À gauche, un premier sous-modèle (droite continue rouge) est estimé de telle sorte qu'il approche un maximum de points avec une erreur inférieure à ϵ (points dans le tubes en tirets). Ensuite, sur le graphique de droite, les points appartenant à ce premier mode sont retirés pour estimer le second sous-modèle (droite continue rouge) de la même manière.

1.5.3 Méthodes à erreur bornée

La méthode algébrique, comme k-LinReg, permet d'identifier des systèmes hybrides à commutation arbitraire pour un nombre de modes fixé. Une autre catégorie de méthodes d'identifi-

cation consiste à estimer ce nombre de modes de manière à respecter une borne que l'on se fixe sur l'erreur de prédiction :

$$|y_k - f(\boldsymbol{x})| \leq \epsilon, \quad k = 1, \ldots, N, \tag{1.48}$$

où f est un modèle commutant entre un nombre de modes suffisant pour satisfaire (1.48). En quelques mots, ces méthodes consistent à estimer les sous-modèles les uns après les autres. Un premier modèle est identifié sur l'ensemble des données, on y associe à ce mode les données comprises sous une borne sur l'erreur de prédiction fixée au préalable, i.e., les points validant (1.48), puis on réitère le procédé jusqu'à ce qu'il n'y ait plus de données en dehors de cette borne, comme illustré en Figure 1.1. Comme originellement proposé par Bemporad et al. (2005), à chaque itération, le sous modèle permettant d'estimer un maximum de points du jeu de données en considérant les autres points comme valeurs aberrantes est recherché. Cette étape revient à résoudre un problème de régression robuste aux valeurs aberrantes, traité par exemple dans Bako (2011) grâce à l'optimisation parcimonieuse et la minimisation ℓ_1.

Ces méthodes permettent donc d'estimer le nombre de modes d'un système hybride, mais en ajoutant un nouvel hyperparamètre à régler qui est le seuil ϵ sur l'erreur d'estimation. Il s'agit de méthodes efficaces, mais qui requièrent une autre information tout aussi sensible que le nombre de modes à acquérir, à savoir : le niveau de bruit. Ici, l'estimation du nombre de modes correspond au nombre nécessaire pour satisfaire les contraintes de précision sur les données disponibles plutôt qu'à l'estimation du vrai nombre de modes.

1.5.4 Estimation du nombre de modes

Comme cela a été révélé dans la section précédente, quelle que soit la méthode d'identification utilisée, l'estimation du nombre de mode reste un problème critique dans la procédure d'identification des systèmes hybrides. Dans cette section, on montre qu'il est possible, par le biais de la méthode algébrique, d'estimer le nombre de modes des systèmes hybrides. D'après Vidal et al. (2003), il suffit de calculer le rang de la matrice $\boldsymbol{L}_j$ pour $j = 1$, et d'itérer jusqu'à ce que la matrice ne soit plus de rang plein :

$$\hat{C} = \min\{j \in \mathbb{N} : \text{rank}(\boldsymbol{L}_j < M_j(d+1)\}. \tag{1.49}$$

Pour la méthode étendue au cas bruité (Ozay et al., 2015), le calcul reste le même, cette fois-ci en considérant la matrice bruitée $\tilde{\boldsymbol{L}}_j(\boldsymbol{\eta})$. Cependant, il est ici nécessaire d'estimer le bruit $\boldsymbol{\eta}$ pour calculer $\tilde{\boldsymbol{L}}_j(\boldsymbol{\eta})$, comme décrit en (1.45). Si cette méthode rend possible l'identification à partir de données bruitées, elle souffre cependant de difficultés dues à une charge de calcul plus importante lorsque l'on cherche à traiter de grands jeux de données. Ceci est d'autant plus vrai si l'on cherche à estimer un grand nombre de modes. En effet, le nombre de modes régit le degré du polynôme, ce qui affecte directement la complexité globale.

Ainsi, les méthodes algébriques permettent d'estimer le nombre de modes des systèmes hybrides. Cependant, le cadre d'étude proposé par ces méthodes demeure relativement contraint en terme de complexité, qui croît en fonction du nombre de modes, et de robustesse, notamment si le bruit n'est pas borné comme en (1.45).

1.6 Conclusion

Dans ce chapitre, les principaux concepts de l'identification des systèmes dynamiques ont été présentés, au sens global dans une première partie, puis, de manière plus spécifique avec l'identification des systèmes dynamiques hybrides dans une seconde partie. Pour ces derniers, on distingue deux types de méthodes.

— les méthodes travaillant avec le nombre de sous-modèles C fixé, qui nécessitent d'estimer le bon nombre de modes. Mais, minimiser (1.30) par rapport à C ne permet pas de trouver C. En effet, l'erreur $J^{ME}(\boldsymbol{\Theta})$ (1.30) décroît de manière monotone lorsque C augmente, sans que cela ne signifie que le modèle estimé est meilleur. Plus C est grand, plus notre modèle sera considéré comme complexe, capable de se superposer au bruit, et ne correspondra pas forcément au vrai système. Ceci correspond à une forme de sur-apprentissage ; cette notion est définie de manière plus détaillée dans le chapitre suivant ;

— les méthodes travaillant à ϵ fixé estiment automatiquement C mais requièrent de choisir le seuil de tolérance ϵ. Ainsi, le choix du paramètre ϵ influence grandement la valeur de C en sortie, ce qui ne résout donc pas le problème du réglage de C, puisque cela ajoute un paramètre supplémentaire à estimer.

En conclusion, l'estimation du nombre de modes est un problème encore ouvert sur lequel il

convient de se pencher. En effet, il n'existe à ce jour que peu de méthodes pour estimer le nombre de modes d'un système hybride, et ces dernières demeurent dans l'ensemble soumises à des contraintes fortes, telles que le nombre de données considérées, le nombre maximal de modes à tester, ou encore la présence de bruit. Les travaux menés dans la thèse portent sur ce sujet, et une nouvelle méthode d'estimation du nombre de modes est présentée dans le chapitre 3. Ces résultats sont obtenus à partir des garanties statistiques pour l'identification de ce type de systèmes. Dans le chapitre suivant, on introduit les concepts généraux de la théorie de l'apprentissage, dont les bornes sur l'erreur de généralisation qui représentent ces garanties.

Chapitre 2

Théorie de l'apprentissage

Ce chapitre a pour objectif d'introduire les concepts généraux de la théorie statistique de l'apprentissage, et vise plus particulièrement à définir les bornes sur l'erreur de généralisation sur lesquelles reposent les principaux résultats obtenus pendant cette thèse. Dans la Section 2.1, on définit le cadre de travail ainsi que les classes de fonctions auxquelles on s'intéresse. Ensuite, les Sections 2.2, 2.3 et 2.4 présentent les bornes sur l'erreur de généralisation, le principe de régularisation et les méthodes de sélection de modèle en suivant la présentation de Mohri et al. (2018). Enfin, dans la Section 2.5 le lien entre la théorie de l'apprentissage et l'identification des systèmes dynamiques hybrides est proposé.

2.1 Cadre général

La théorie statistique de l'apprentissage vise à obtenir des garanties sur les modèles appris à partir de données dans un cadre peu contraint. Par exemple, la connaissance du type de bruit ou de la structure du modèle, qui sont des hypothèses fortes habituellement nécessaires en identification des systèmes, n'est pas requise. Ces garanties prennent typiquement la forme de bornes sur l'espérance de l'erreur de prédiction, autrement appelée risque ou erreur de généralisation.

Pour présenter ces notions, il est tout d'abord nécessaire d'introduire quelques notations. On définit $\mathcal{X}$ l'espace d'entrée et $\mathcal{Y}$ l'espace de sortie, avec, dans le contexte de la régression, $\mathcal{Y} = [-M, M]$ pour $M > 0$. Le lien entre l'entrée et la sortie d'un système est caractérisé par la loi de probabilité *inconnue* de la paire de variables aléatoires $Z = (X, Y) \in \mathcal{X} \times \mathcal{Y} = \mathcal{Z}$, de

densité de probabilité $p(x, y)$.

Soit la réalisation d'un échantillon $\mathbf{Z}_N = (Z_k)_{1\leq k\leq N} = ((X_k, Y_k))_{1\leq k\leq N}$ de N copies indépendantes et identiquement distribuées de Z, l'objectif de la régression est d'apprendre le modèle $f : \mathcal{X} \to \mathcal{Y}$ qui minimise, au sein d'une certaine classe de modèles $\mathcal{F}$, l'erreur de généralisation

$$L(f) = \mathbb{E}_{X,Y}\ell(f, X, Y), \tag{2.1}$$

définie comme l'espérance $\mathbb{E}_{X,Y}\ell(f, X, Y) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(f, x, y)p(x, y)\, dxdy$, de la fonction de perte $\ell(f, X, Y)$.

Cette fonction de perte mesure l'erreur de prédiction entre le modèle $f(x)$ et les sorties réelles y du système. Cette mesure prend généralement la forme $\ell(f, X, Y) = \ell_p(Y - f(X))$ avec ℓ_p donnée en (1.19) lorsque l'on considère des problèmes de régression standards sans commutation.

En pratique, le risque $L(f)$ (2.1) ne peut être calculé sans la connaissance de la distribution de Z. De ce fait, on cherche bien souvent à minimiser le risque empirique défini par

$$\hat{L}_N(f) = \frac{1}{N}\sum_{k=1}^{N} \ell(f, X_k, Y_k), \tag{2.2}$$

pour une réalisation de $\mathbf{Z}_N = \mathbf{z}_N$.

2.2 Bornes sur l'erreur de généralisation

Dans cette partie, on définit comment obtenir des garanties sur les modèles estimés par le biais de bornes sur l'erreur de généralisation $L(f)$ définie en (2.1). Par erreur de généralisation, on entend la capacité du modèle appris à prédire correctement le comportement du système sur de nouvelles entrées. Les bornes que l'on cherche à obtenir prennent la forme suivante

$$P^N\left\{\forall f \in \mathcal{F},\ L(f) \leq \hat{L}_N(f) + \epsilon(N, \mathcal{F}, \delta)\right\} \geq 1 - \delta, \tag{2.3}$$

où la borne sur $L(f)$ est valide avec une probabilité d'au moins $1-\delta$, avec $\delta \in (0, 1)$, sur le tirage de $\mathbf{Z}_N$. On distingue alors deux termes dans la borne :

— l'erreur empirique $\hat{L}_N(f)$, définie en (2.2), qui représente la précision du modèle estimée

sur les données d'apprentissage ;

— un intervalle de confiance ϵ dépendant de la complexité de la classe de fonctions $\mathcal{F}$ étudiée, qui représente la qualité de cette estimation, et que l'on détaillera par la suite.

Dans cette thèse, les bornes sont développées pour les modèles à commutations. Et, comme discuté dans le chapitre 1, il est rarement possible de garantir que ce type de modèle minimise le risque empirique. C'est pourquoi on se concentrera particulièrement sur des bornes uniformes, c'est-à-dire valables pour tout modèle f de la classe de fonctions $\mathcal{F}$, qui s'appliquent sans faire d'hypothèses sur l'algorithme d'optimisation utilisé.

Une première borne peut être obtenue en appliquant directement l'inégalité de Hoeffding :

Théorème 1 (Borne pour un modèle particulier) *Soit un modèle $f : \mathcal{X} \to \mathcal{Y}$ et la fonction de perte $\ell(f, X, Y)$ à valeurs dans $[0, B]$. Pour tout $\delta > 0$, avec une probabilité égale ou supérieur à $1 - \delta$, on a l'inégalité suivante*

$$L(f) \leq \hat{L}_N(f) + B\sqrt{\frac{\log \frac{1}{\delta}}{2N}}. \tag{2.4}$$

Preuve du Théorème 1: En appliquant la deuxième inégalité de Hoeffding (voir Théorème 9 dans l'Annexe A) , on obtient

$$P^N \left[L(f) - \hat{L}_N(f) \geq \epsilon \right] \leq \exp\left(\frac{-2N\epsilon^2}{B^2} \right) \tag{2.5}$$

On introduit δ

$$\delta = \exp\left(\frac{-2N\epsilon^2}{B^2} \right), \tag{2.6}$$

Ce qui permet d'exprimer ϵ en fonction de δ : $\epsilon = B\sqrt{\frac{\log \frac{1}{\delta}}{2N}}$.

Ainsi on a

$$P^N \left[L(f) - \hat{L}_N(f) \geq B\sqrt{\frac{\log \frac{1}{\delta}}{2N}} \right] \leq \delta \tag{2.7}$$

Finalement, appliquer le principe de l'inversion de probabilité complète la preuve. □

Le Théorème 1 introduit une borne sur l'erreur de généralisation d'une fonction f, fixée *a priori*. Cette borne indique que pour une fonction donnée f, il existe une séquence de données

pour laquelle la borne est valide avec une probabilité d'au moins $1-\delta$. Cependant, ces données peuvent être différentes pour un autre modèle f'. Cela signifie qu'il n'y a aucune garantie que la borne reste valide pour toutes les autres fonctions de $\mathcal{F}$, pour une séquence de données particulière. Cette borne n'est donc pas utilisable pour mesurer la qualité de prédiction d'un modèle issu d'un algorithme. Pour pallier cette limitation, il faut alors rendre la borne valide pour un plus grand ensemble de modèles, cela signifie rendre la borne uniformément valide pour tout $f \in \mathcal{F}$.

On s'intéresse alors à des bornes de la forme

$$\sup_{f\in\mathcal{F}}(L(f) - \hat{L}_N(f)) \leq \epsilon, \tag{2.8}$$

où $\mathcal{F}$ est la classe de fonctions regroupant l'ensemble des fonctions f implémentées par l'algorithme. Par exemple, la borne suivante s'applique à des classes $\mathcal{F}$ finies.

Théorème 2 (Borne sur une classe de modèles finie $\mathcal{F}$) *Soit une classe de fonctions $\mathcal{F}$ de cardinalité $|\mathcal{F}|$. Avec une probabilité d'au moins $1-\delta$, on a*

$$\forall f \in \mathcal{F},\ L(f) \leq \hat{L}_N(f) + B\sqrt{\frac{\log|\mathcal{F}| + \log\frac{1}{\delta}}{2N}}. \tag{2.9}$$

Preuve du Théorème 2: Pour chaque fonction $f_i \in \mathcal{F}$, on définit A_i l'évènement qui correspond à l'observation d'un jeu de données donnant un grand écart entre le risque et le risque empirique. On a alors, par application du théorème d'Hoeffding

$$P^N(A_i) = P^N\left\{L(f_i) - \hat{L}_N(f_i) \geq \epsilon\right\} \leq \exp\left(\frac{-2N\epsilon^2}{B^2}\right) \tag{2.10}$$

La probabilité que le risque soit élevé pour n'importe lequel des $|\mathcal{F}|$ modèles de $\mathcal{F}$ correspond à la probabilité

$$P^N(A_1 \cup \cdots \cup A_n) \leq \sum_{i=1}^{|\mathcal{F}|} P^N(A_i) \tag{2.11}$$

ce qui donne

$$P^N\left\{\exists f \in \mathcal{F} \ : \ L(f) - \hat{L}_N(f) \geq \epsilon\right\} \leq \sum_{i=1}^{|\mathcal{F}|} \exp\left(\frac{-2N\epsilon^2}{B^2}\right) \tag{2.12}$$

$$\leq |\mathcal{F}| \exp\left(\frac{-2N\epsilon^2}{B^2}\right). \tag{2.13}$$

Appliquer le même raisonnement que pour le Théorème 1 avec $\delta = |\mathcal{F}| \exp\left(\frac{-2N\epsilon^2}{B^2}\right)$ complète la preuve. □

La borne (2.9) est valide pour toute fonction issue de la classe $\mathcal{F}$ finie. Elle se révèle bien plus utile que la première étant donné qu'elle est valide sur l'ensemble de la classe de fonctions $\mathcal{F}$ mais présente cependant encore une limite non négligeable. Dans le cadre de la régression linéaire, par exemple, les classes de fonctions étudiées sont composées d'un nombre infini de fonctions. Dans le Théorème 2, l'étude de ce type de classes n'est pas possible, car la borne serait infinie. Pour contourner cette limite, on cherchera à prendre en compte ces classes de fonctions non pas au travers du nombre de modèles possibles mais par des mesures de complexité plus fines. Dans la section suivante, on définit la complexité de Rademacher, qui correspond à l'une de ces mesures de complexités.

2.2.1 Complexité de Rademacher

Pour obtenir des bornes sur l'erreur de généralisation avec des classes de fonctions de cardinalité infinie, il est nécessaire de mesurer leur complexité. Il existe plusieurs méthodes utilisées en théorie de l'apprentissage qui s'appuient par exemple sur la fonction de croissance, la VC-dimension (pour dimension de Vapnik-Chervonenkis (Vapnik, 1998)), ou la complexité de Rademacher. Dans cette section, c'est cette dernière que l'on étudie. Un avantage de cette mesure de complexité est que sa version empirique dépend directement des données d'entraînement, et prend alors mieux en compte les propriétés des distributions qui ont généré les données.

Pour garantir les performances d'une classe de fonctions $\mathcal{F}$, on mesure la complexité de la classe de fonctions de perte, qui mesure l'erreur de notre modèle

$$\mathcal{L} = \left\{\ell \in \mathbb{R}^{\mathcal{Z}} \ : \ \ell(z) = \ell(f, x, y), \ f \in \mathcal{F}\right\}. \tag{2.14}$$

Définition 1 *(Complexité de Rademacher empirique) Soit une séquence de variables aléatoires* $\mathbf{Z}_N = (Z_k)_{1\leq k\leq N}$*, avec* $Z_k \in \mathcal{Z}$*, la complexité de Rademacher empirique d'une classe de fonctions* $\mathcal{L}$ *est définie par*

$$\hat{\mathcal{R}}_{\mathbf{Z}_N}(\mathcal{L}) = \mathbb{E}_{\boldsymbol{\sigma}_N}\left[\sup_{\ell\in\mathcal{L}} \frac{1}{N}\sum_{k=1}^{N} \sigma_k \ell(Z_k) \middle| \mathbf{Z}_N\right], \tag{2.15}$$

où $\sigma_N = (\sigma_k)_{1\leq k\leq N}$ *est une séquence de variables de Rademacher indépendantes, c'est-à-dire des variables aléatoires uniformément distribuées dans* $\{-1, 1\}$*.*

En d'autres termes, la complexité de Rademacher mesure la capacité de la classe de fonctions à générer des modèles dont les sorties s'alignent sur le bruit, qui est ici illustré par les variables de Rademacher.

En considérant la complexité de Rademacher, on obtient la borne générale suivante.

Théorème 3 (Borne à base de complexité de Rademacher. Mohri et al. (2018)) *Soit* $\mathcal{L}$ *une classe de fonctions de* $\mathcal{Z}$ *à valeurs dans* $[0, B]$ *et* $\mathbf{Z}_N = (Z_k)_{1\leq k\leq N}$ *une séquence de copies indépendantes de la variable aléatoire* $Z \in \mathcal{Z}$*. Alors, pour tout* $\delta \in (0, 1)$ *fixé* a priori*, avec une probabilité d'au moins* $1 - \delta$ *sur le tirage de* $\mathbf{Z}_N$*, pour tout* $\ell \in \mathcal{L}$*,*

$$\mathbb{E}_Z \ell(Z) \leq \frac{1}{N}\sum_{k=1}^{N} \ell(Z_k) + 2\hat{\mathcal{R}}_{\mathbf{Z}_N}(\mathcal{L}) + 3B\sqrt{\frac{\log\frac{2}{\delta}}{2N}}. \tag{2.16}$$

Ici, si $\ell(Z)$ correspond à la fonction de perte à la base du risque (2.1), la borne (2.16) correspond à celle présentée initialement en (2.3), où l'intervalle de confiance ϵ est développé en considérant la complexité de Rademacher : $\epsilon(N, \mathcal{F}, \delta) = 2\hat{\mathcal{R}}_{\mathbf{Z}_N}(\mathcal{L}) + 3B\sqrt{\frac{\log\frac{2}{\delta}}{2N}}$, avec une dépendance à $\mathcal{F}$ au travers de $\mathcal{L}$.

Pour utiliser les bornes présentées dans les Théorème 1 à 3 en régression, il faut borner la fonction de perte $\ell(f, x, y)$ en faisant l'hypothèse que la sortie est bornée par M, i.e., $Y \in [-M, M]$, puis tronquer les fonctions f.

Définition 2 (Fonction tronquée) *Pour tout* $M > 0$ *et* $t \in \mathbb{R}$*, on définit la version tronquée* $\bar{t}$ *de* t *par*

$$\bar{t} = \min(M, \max(-M, t)).$$

La version tronquée $\overline{f}$ *de* $f : \mathcal{X} \to \mathbb{R}$ *est obtenue en tronquant la sortie :* $\forall x \in \mathcal{X},\ \overline{f}(x) = \overline{f(x)}$. *Et* $\overline{\mathcal{F}}$ *est la classe de fonctions tronquées* $\{\overline{f} : f \in \mathcal{F}\}$.

Ainsi, sous l'hypothèse que $Y \in [-M, M]$, il est juste de dire que pour tout $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $\ell(\overline{f}, x, y) \leq \ell(f, x, y)$, et que dans la continuité, l'erreur de généralisation $L(\overline{f})$ sera toujours plus faible que $L(f)$. Dans la suite, on considérera alors $\overline{f}$ à la place de f pour réaliser des prédictions, et les bornes seront dérivées pour le risque $L(\overline{f})$ du modèle tronqué.

L'exemple 1 montre comment appliquer ce type de bornes en régression linéaire.

Exemple 1 *(Régression linéaire)*
Dans le cadre de la régression linéaire avec $\boldsymbol{X} \in \mathcal{X} = \mathbb{R}^d$, *on considère la classe de modèles suivante*

$$\mathcal{F} = \{f : f(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x},\ \boldsymbol{\theta} \in \mathbb{R}^d,\ \|\boldsymbol{\theta}\|_2 \leq \Lambda\}, \tag{2.17}$$

où $\|\boldsymbol{\theta}\|_2$ *mesure la norme euclidienne du vecteur de paramètres, que l'on considère bornée par* Λ.

On définit la classe de fonctions de perte issue du modèle tronqué à partir de la perte ℓ_2

$$\mathcal{L} = \{\ell \in [0, 4M^2]^{\mathcal{Z}} : \ell(\boldsymbol{z}) = (y - \overline{f}(\boldsymbol{x}))^2,\ f \in \mathcal{F}\}.$$

La complexité de Rademacher de $\mathcal{L}$ *peut être exprimée en fonction de la classe* $\mathcal{F}$ *en utilisant le lemme de contraction (voir le Théorème 10 dans l'Annexe A), ce qui donne*

$$\hat{\mathcal{R}}_{\mathbf{Z}_N}(\mathcal{L}) \leq 4M \hat{\mathcal{R}}_{\mathbf{X}_N}(\mathcal{F}),$$

que l'on sait borner par des méthodes standards de calcul de la complexité de Rademacher (Théorème 11 de l'Annexe A)

$$\hat{\mathcal{R}}_{\mathbf{X}_N}(\mathcal{F}) \leq \frac{\Lambda}{N} \sqrt{\sum_{k=1}^{N} \|\boldsymbol{X}_k\|_2^2}. \tag{2.18}$$

Ainsi, par application du Théorème 3, pour tout $\delta \in (0, 1)$, *avec une probabilité d'au moins* $1 - \delta$, *pour tout* $f \in \mathcal{F}$,

$$L(\overline{f}) \leq \hat{L}_N(\overline{f}) + \frac{8M\Lambda \sqrt{\sum_{k=1}^{N} \|\boldsymbol{X}_k\|_2^2}}{N} + 12M^2 \sqrt{\frac{\log \frac{2}{\delta}}{2N}}, \tag{2.19}$$

où, dans ce cas, $L(\overline{f})$ correspond à l'espérance de l'erreur quadratique, tandis que $\hat{L}_N(\overline{f})$ correspond à l'erreur quadratique moyenne sur l'échantillon de taille N.

L'Exemple 1 propose une borne sur l'erreur de généralisation en étudiant une classe de fonctions régularisée. En effet, la classe $\mathcal{F}$ est définie telle que $\|\boldsymbol{\theta}\|_2 \leq \Lambda$. Ce type de régularisation est assez classique, néanmoins, il existe d'autres formes de régularisation possibles que nous traiterons notamment dans le Chapitre 4. Avant cela, on définit le principe de régularisation dans la section suivante.

2.3 Régularisation

La régularisation est une technique standard de contrôle de la complexité du modèle pendant son apprentissage. Si l'on ne régularise pas, on permet à l'algorithme de choisir des modèles trop complexes pour apprendre les données, et ces modèles risquent fortement d'être mauvais avec de nouvelles données. Ce comportement est appelé le surapprentissage (voir Figure 2.1). La

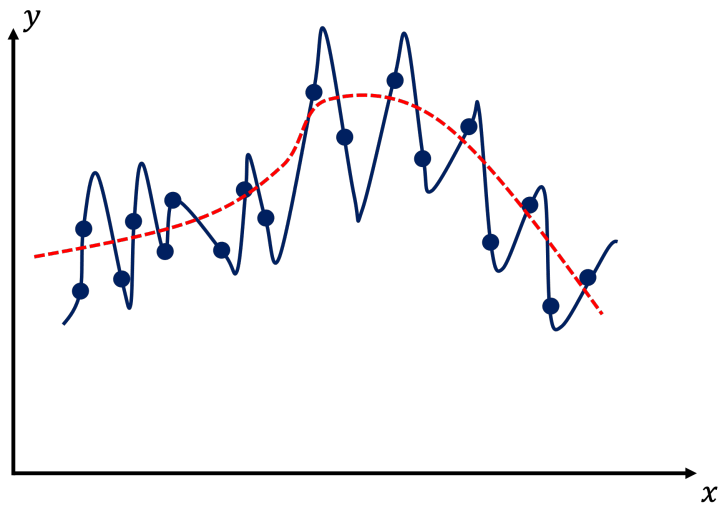

FIGURE 2.1 – Illustration du surapprentissage. Sans contrôle de la complexité, la courbe en trait plein bleu ne fait pas d'erreur sur les points et présente beaucoup de variations, tandis qu'en pointillés rouges, le modèle est moins complexe et prédit mieux le comportement du système.

régularisation sert donc à préciser l'apprentissage d'un modèle en y ajoutant de l'information afin d'éviter le surapprentissage. Il s'agit d'apprendre un modèle en minimisant un compromis entre l'erreur d'apprentissage $\mathcal{E}(f, \mathbf{X}, \boldsymbol{y})$ et un terme de régularisation $\Gamma(f)$, qui pénalise les modèles

les plus complexes

$$\min_{f \in \mathcal{F}} \mathcal{E}(f, \mathbf{X}, \boldsymbol{y}) + \lambda \Gamma(f), \tag{2.20}$$

où $\lambda > 0$ est un hyperparamètre qui contrôle le compromis entre les deux termes. À partir de cette représentation, différentes méthodes de régularisation peuvent émerger suivant le choix du calcul de l'erreur ainsi que celui de la complexité du modèle.

L'erreur est typiquement définie comme la somme de l'erreur sur les données d'apprentissage

$$\mathcal{E}(f, \mathbf{X}, \boldsymbol{y}) = \sum_{k=1}^{N} \ell(f, \boldsymbol{x}_k, y_k) \tag{2.21}$$

avec $\ell(f, \boldsymbol{x}_k, y_k)$ une fonction de perte comme définie en (2.1).

En régularisant, on cherche donc à contrôler la complexité du modèle retenu par l'algorithme d'apprentissage en pénalisant les modèles les plus complexes, via le terme $\Gamma(f)$, souvent exprimé comme une norme de f ou de ses paramètres. Le concept de régularisation sera exploité de manière plus approfondie dans le Chapitre 4.

2.4 Sélection de modèle

Un des paramètres clés en apprentissage est le choix de la classe de fonctions dans laquelle l'algorithme va choisir le modèle. Ce problème est celui de la sélection de modèle. Il s'agit de choisir une classe de modèles suffisamment complexe, ou riche, pour que l'algorithme soit en mesure de s'adapter à de nombreux problèmes, sans pour autant choisir une classe trop complexe, ce qui aurait pour conséquence de mener à des problèmes de surapprentissage.

Généralement, on présente ces deux notions en exprimant la différence entre l'erreur de généralisation d'un modèle estimé $L(\hat{f})$ et celle du modèle optimal, $L^* = \min_{f : \mathcal{X} \to \mathcal{Y}} L(f)$.

$$L(\hat{f}) - L^* = \left(L(\hat{f}) - \min_{f \in \mathcal{F}} L(f) \right) + \left(\min_{f \in \mathcal{F}} L(f) - L^* \right), \tag{2.22}$$

où le premier terme correspond à l'erreur d'estimation tandis que le second à l'erreur d'approximation.

— L'erreur d'estimation correspond à la différence entre l'erreur de la fonction choisie par

l'algorithme et l'erreur de la meilleure des fonctions de la classe considérée $\mathcal{F}$;

— L'erreur d'approximation quant à elle sera plus on moins grande en fonction de la classe de fonctions $\mathcal{F}$ choisie et de sa capacité à approcher le modèle optimal.

Plus la classe de fonctions sera riche, plus l'erreur d'approximation sera faible, mais plus l'erreur d'estimation sera grande car il sera plus délicat d'identifier un bon modèle parmi un choix plus vaste. Ce problème de sélection de modèle peut être illustré par la Figure 2.2, où la classe de fonctions $\mathcal{F}$ est choisie comme étant celle qui minimise (2.22) par rapport à un hyperparamètre γ.

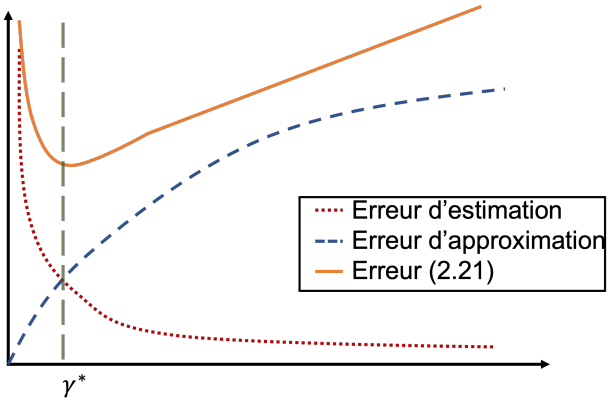

FIGURE 2.2 – Sélection de modèle, où la somme de la courbe de l'erreur d'approximation (en tirets bleus) et celle de l'erreur d'estimation (en pointillés rouges) donne la courbe de la borne (en trait plein orange) et où un minimum est atteint en γ^*, l'hyperparamètre que l'on cherche à régler.

Dans la section suivante, on présente une approche de sélection de modèle, le principe de minimisation du risque structurel, qui vise à calibrer la classe de fonctions sur laquelle on travaille.

2.4.1 Minimisation du risque structurel

La question qui découle de l'introduction de cette section est la suivante : comment choisir notre classe de fonctions $\mathcal{F}$ afin d'optimiser l'erreur de généralisation, qui ne peut être calculée ? Le principe de minimisation du risque structurel répond à cette problématique. Celui-ci consiste à définir dans un premier temps la classe de fonctions la plus riche possible. En supposant que

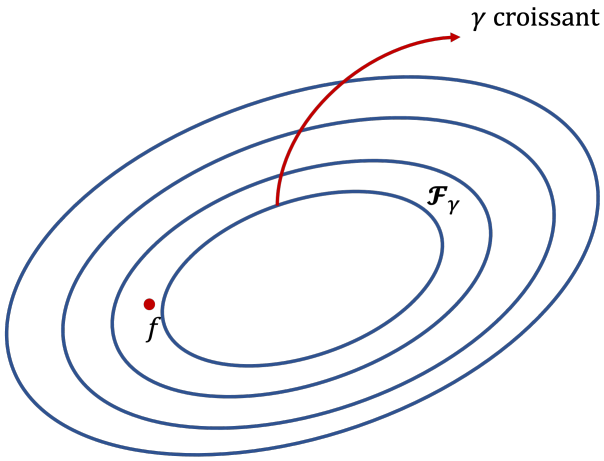

FIGURE 2.3 – Décomposition d'une classe $\mathcal{F}$ en K sous-classes de capacité croissante

l'on peut décomposer cette classe $\mathcal{F}$ comme l'union des classes imbriquées $\mathcal{F}_\gamma$ telles que

$$\mathcal{F} = \bigcup_{\gamma \in [K]} \mathcal{F}_\gamma,$$

et $\mathcal{F}_\gamma \subset \mathcal{F}_{\gamma+1}$, voir figure 2.3, on va traquer quelle sous-classe $\mathcal{F}_\gamma$ répond au mieux au compromis entre erreur d'estimation et erreur d'approximation. En résumé, on cherche l'index γ, ainsi que la fonction $f \in \mathcal{F}_\gamma$ qui minimisent l'erreur de généralisation (cf. Figure 2.2).

Comme il n'est pas possible de minimiser l'erreur de généralisation directement, une approche consiste à minimiser une borne sur cette erreur, ce qui revient à minimiser un critère de la forme

$$J(\gamma) = \underset{\gamma \geq 1, f \in \mathcal{F}_\gamma}{\operatorname{argmin}} \hat{L}_N(f) + \epsilon(N, \mathcal{F}_\gamma, \delta),$$

qui correspond à la partie droite de notre borne sur l'erreur de généralisation et où $\epsilon(N, \mathcal{F}_\gamma, \delta)$ peut être exprimé en considérant la complexité de Rademacher.

Les Sections 2.1 à 2.4 ont permis de définir les bases de la théorie statistique de l'apprentissage, notamment des bornes sur l'erreur de généralisation. Dans la section suivante, on introduit un premier lien entre la théorie de l'apprentissage et l'identification des systèmes.

2.5 Application à l'identification de systèmes dynamiques

Les différentes sections de ce chapitre ont pu témoigner de l'intérêt et de l'efficacité de la théorie de l'apprentissage pour obtenir des garanties sur nos modèles dans un contexte de régression. Les méthodes de régularisation et de sélection de modèle, qui visent à optimiser la structure des modèles appris, ont également été introduites. Ici, l'objectif est maintenant de faire le lien entre ce chapitre et l'identification des systèmes dynamiques présentée dans le Chapitre 1. Une hypothèse forte de la théorie de l'apprentissage est la nature indépendante et identiquement distribuée des données, qui n'est pas compatible avec l'étude des systèmes dynamiques qui présentent des dépendances entre les données d'entrée-sortie. Cependant, de récents travaux traitant de bornes sur l'erreur de généralisation ont étudié leur application au cas des données interdépendantes. Ces travaux font appel à des mesures de dépendance entre les données issues des processus mélangeants, et plus spécifiquement des processus β-mélangeants (voir Bradley (2005) pour une vue globale de ces processus).

2.5.1 Bornes pour des données non-i.i.d et processus β-mélangeants

À partir de cette section, sauf si le contraire est mentionné, on suppose que l'échantillon $\mathbf{Z}_N$ est issu d'un processus stationnaire β-mélangeant. On définit donc la stationnarité et les processus β-mélangeant.

Définition 3 (Stationnarité) *Une séquence de variables aléatoires $\mathbf{Z} = \{Z_t\}_{t=-\infty}^{\infty}$ est dite stationnaire si, pour tout t et tout entier non négatif m et k, les vecteurs aléatoires $(Z_t, \ldots, Z_m)$ et $(Z_{t+k}, \ldots, Z_{t+k+m})$ ont la même distribution.*

Définition 4 (Processus β-mélangeant) *Soit $\mathbf{Z} = \{Z_t\}_{t=-\infty}^{\infty}$ une séquence de variables aléatoires stationnaires. Pour tout $i, j \in \mathbb{Z} \cup \{-\infty, \infty\}$, on définit σ_i^j comme la tribu engendrée par les variables aléatoires $Z_k, i \leq k \leq j$. Ainsi, pour tout entier positif k, le coefficient de mélange β du processus aléatoire $\mathbf{Z}$ est défini par*

$$\beta(k) = \mathbb{E}_{B \in \sigma_{-\infty}^{0}} \left\{ \sup_{A \in \sigma_k^{\infty}} |\mathbb{P}(A|B) - \mathbb{P}(A)| \right\}. \tag{2.23}$$

Si $\beta(k) \to 0$ lorsque $k \to \infty$, alors $\mathbf{Z}$ est dit β-mélangeant.

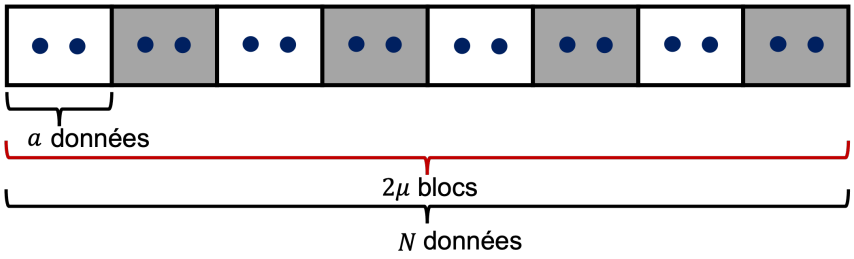

FIGURE 2.4 – Illustration de la séquence de blocs avec $N = 20$, $a = 2$ points de données par bloc et $\mu = 5$ blocs impairs (blanc) et μ blocs pairs (gris). L'échantillon $\boldsymbol{Z}_\mu = (Z_1, Z_{2a+1}, Z_{4a+1}, \ldots, Z_{8a+1})$ contient le premier point de chaque bloc impair. La dépendance entre les blocs impairs (et les points de $\boldsymbol{Z}_\mu$) décroît en fonction de la longueur a des blocs de rang pair les séparants.

Le cœur de la méthode permettant d'obtenir des bornes pour des données non-i.i.d. réside dans l'hypothèse que les données sont β-mélangeantes. Car, comme défini ci-dessus, si les données sont issues d'un processus β-mélangeant, alors, la dépendance entre deux points d'un jeu de données diminue en fonction de la distance temporelle qui sépare ces deux points. Plus les points sélectionnés sont éloignés les uns des autres, moins les données sont dépendantes entre elles.

C'est en se basant sur cette hypothèse que Yu (1994) propose une méthode pour créer un jeu de données qui tend à être i.i.d à partir de données dépendantes. Cette méthode est ensuite appliquée dans les travaux de Mohri and Rostamizadeh (2009) pour dériver des bornes sur l'erreur de généralisation à base de complexité de Rademacher pour des processus non-i.i.d.

Séquence de blocs indépendants

La méthode proposée par Yu (1994), et illustrée en Figure 2.4, consiste dans un premier temps à séparer une séquence de taille N de données dépendantes issues d'un processus β-mélangeant en une séquence de blocs de taille a. Ensuite, il suffit de ne considérer qu'un bloc sur deux, et de jeter les autres, pour obtenir une séquence de μ blocs. Ainsi, en fonction de la taille des blocs a, la dépendance entre deux des μ blocs sera moindre qu'entre deux variables consécutives de la séquence originelle, ce qui permet de dériver des bornes.

Théorème 4 (Théorème 2 dans Mohri and Rostamizadeh (2009)) *Soit $\mathcal{L}$ une classe de fonctions de $\mathcal{Z}$, à valeurs dans $[0, B]$ et $\mathbf{Z}_N = (Z_k)_{1 \leq k \leq N}$ une séquence issue d'un processus β-mélangeant. Pour tout $\mu, a > 0$ avec $2\mu a = N$ et $\delta > 4(\mu - 1)\beta(a)$, avec une probabilité d'au*

moins $1-\delta$*, uniformément pour tout* $\ell \in \mathcal{L}$*,*

$$\mathbb{E}\ell(Z_1) \leq \frac{1}{N}\sum_{k=1}^{N}\ell(Z_k) + 2\mathcal{R}_{\mathbf{Z}_\mu}(\mathcal{L}) + 3B\sqrt{\frac{\log\frac{4}{\delta'}}{2\mu}}, \tag{2.24}$$

où $\delta' = \delta - 4(\mu-1)\beta(a)$ *et* $\mathbf{Z}_\mu = (Z_{2a(k-1)+1})_{1\leq k\leq N}$ *est un échantillon de longueur* μ *comme illustré en Figure. 2.4.*

En comparaison avec le Théorème 3, le Théorème 4 voit son intervalle de confiance décroître en fonction de μ plutôt que N, car on ne considère qu'une certaine partie de l'échantillon de départ en appliquant la méthode des blocs indépendants. De plus, l'intervalle de confiance sera également impacté par l'utilisation de δ' à la place de δ.

La borne présentée dans le Théorème 4 permet l'analyse de fonctions obtenues à partir de données issues d'un processus stationnaire β-mélangeant qui par exemple, peuvent correspondre à des séries temporelles (McDonald et al., 2015) ou des trajectoires de systèmes ARX (Vidyasagar and Karandikar, 2008). Il est alors possible d'étendre cette étude à une application plus ciblée comme l'identification des systèmes hybrides, et c'est ce qui sera présenté dans le Chapitre 3.

2.5.2 Autres approches

Ces dernières années, d'autres approches pour dériver des bornes adaptées à des données non-i.i.d mais n'employant pas les processus mélangeant ont émergées. Dans Usunier et al. (2005); Ralaivola and Amini (2015), les auteurs dérivent des bornes à base complexité de Rademacher pour la classification en s'inspirant des travaux menés dans Janson (2004). Ils emploient une méthode de coloration de graphes qui revient à créer des sous-ensembles au sein desquels on regroupe les données indépendantes entre elles. Cela permet alors de ne considérer qu'un seul sous-ensemble à la fois. Dans Lampert et al. (2018), ces travaux intègrent les processus mélangeant pour considérer toutes les données et prendre en compte les dépendances au sein des groupes. Des bornes valides dans le cas de données dépendantes sont étudiées dans Shalaeva et al. (2020), dans un cadre bayésien et différent du cadre considéré dans cette thèse, bien que les résultats pour l'identification de systèmes linéaires y soient présentés. Les travaux de Simchowitz et al. (2018), quand à eux, proposent de s'affranchir des conditions de mélange pour analyser l'es-

timateur des moindres carrés dans un contexte d'identification linéaire. Cependant, les techniques utilisées ne paraissent pas adaptées à l'identification de systèmes hybrides. En effet, cette méthode semble limitée à l'analyse du modèle minimisant l'erreur empirique, $\hat{f} \in \operatorname{argmin}_{f \in \mathcal{F}} \hat{L}_N(f)$, qui n'est pas atteignable dans la plupart des cas pour les systèmes hybrides, comme rappelé dans la Section 1.5.

2.6 Conclusion

Dans ce chapitre, le cadre général de la théorie de l'apprentissage a été présenté, en s'intéressant tout particulièrement aux bornes sur l'erreur de généralisation à base de complexité de Rademacher. Les travaux de Mohri and Rostamizadeh (2009) démontrent qu'il est possible d'appliquer la théorie de l'apprentissage aux systèmes dynamiques en dérivant des bornes de ce type. Ainsi, le chapitre suivant présente une des contributions de la thèse qui s'appuie sur les notions présentées dans les deux premiers chapitres pour développer une borne à base de complexité de Rademacher valide pour l'identification des systèmes dynamiques hybrides.

Chapitre 3

Théorie de l'apprentissage pour l'identification des systèmes hybrides

Dans ce chapitre, le lien entre théorie de l'apprentissage et l'identification des systèmes est développée pour proposer de nouvelles approches dédiées à l'identification de systèmes dynamiques hybrides. Ainsi, de nouvelles bornes sur l'erreur de généralisation valides pour l'identification des systèmes hybrides sont développées. Pour ce faire, les résultats présentés dans la Section 2.5 sont associés au cadre d'étude de l'identification des systèmes hybrides de la Section 1.5. Ensuite, ces bornes sont appliquées dans le cadre de la sélection de modèles et une nouvelle méthode d'estimation du nombre de modes est également présentée.

Pour ce faire, on considère des systèmes hybrides ARX de la forme

$$\begin{aligned} y_k &= f_{r_k}(\boldsymbol{x}_k) + e_k, \\ f_j(\boldsymbol{x}_k) &= \boldsymbol{\theta}_j^T \boldsymbol{x}_k, \qquad j = 1, \dots, C, \end{aligned} \tag{3.1}$$

où $y_k \in \mathbb{R}$ correspond à la sortie, $\boldsymbol{x}_k \in \mathcal{X} \subset \mathbb{R}^d$ au vecteur de régression, $r_k \in \{1, \dots, C\}$ à la séquence de commutation, C au nombre de sous-modèles, f_j avec $j \in \{1, \dots, C\}$ aux sous-modèles linéaires de vecteurs de paramètres $\boldsymbol{\theta}_j \in \mathbb{R}^d$ et $e_k \in \mathbb{R}$ est un terme de bruit. Le regresseur $\boldsymbol{x}_k \in \mathbb{R}^d$, $d = n_a + n_b$, d'ordres n_a et n_b a la forme

$$\boldsymbol{x}_k = [-y_{k-1}, \ \dots, -y_{k-n_a}, \ u_{k-1}, \dots, u_{k-n_b}]^T, \tag{3.2}$$

où les entrées u_{k-t} représentent les entrées passées.

La Section 3.1 présente plus précisément la classe de modèles que l'on étudie, notamment en définissant la manière dont on mesure sa complexité. La Section 3.2 introduit une borne sur l'erreur de généralisation valide pour cette classe de modèles, tandis que dans la Section 3.3, on étend cette borne afin de l'utiliser pour proposer une nouvelle approche permettant d'estimer le nombre de modes.

3.1 Théorie de l'apprentissage et identification des systèmes hybrides

Dans le cadre de l'identification des systèmes hybrides, on considère des classes de fonctions $\boldsymbol{f} = (f_j)_{1\leq j\leq C}$ à valeurs vectorielles. Pour apprendre ce type de classe, il faut adapter la définition de l'erreur de généralisation :

$$L(\boldsymbol{f}) = \mathbb{E}_{X,Y}\ell_{\min}(\boldsymbol{X}, Y, \boldsymbol{f}), \tag{3.3}$$

avec $\ell_{\min}(\boldsymbol{x}_k, y_k, \boldsymbol{f}) = \min_{j\in[C]} |y_k - f_j(\boldsymbol{x}_k)|^p$ définie comme en (1.31) [2], et l'erreur empirique

$$\hat{L}_N(\boldsymbol{f}) = \frac{1}{N}\sum_{k=1}^{N} \ell_{\min}(\boldsymbol{X}_k, Y_k, \boldsymbol{f}). \tag{3.4}$$

Ainsi, la mesure de l'erreur de généralisation correspond au problème d'identification (1.30) des systèmes hybrides présenté en Section 1.5.

Ici, la mesure de la complexité des modèles à commutation se fait à deux niveaux :

— au niveau de chaque composante f_j en mesurant la norme de chaque vecteur de paramètres $\|\boldsymbol{\theta}_j\|_2$;

— au niveau global, en prenant en compte l'ensemble des complexités des sous-modèles f_j avec $\Omega(\boldsymbol{f})$

$$\Omega(\boldsymbol{f}) = [\|\boldsymbol{\theta}_1\|_2,\ \ldots,\ \|\boldsymbol{\theta}_C\|_2]^T. \tag{3.5}$$

On définit alors la classe de fonctions dont la complexité globale est contrôlée par un hyperpa-

2. On rappelle la notation $[C] = \{1, \ldots, C\}$.

ramètre Λ :

$$\mathcal{F} = \left\{ \boldsymbol{f} = (f_j)_{1 \leq j \leq C} : f_j(\boldsymbol{x}) = \boldsymbol{\theta}_j^T \boldsymbol{x},\ j = 1 \dots C,\ \|\Omega(\boldsymbol{f})\|_q \leq \Lambda \right\}, \tag{3.6}$$

où le niveau global de la mesure de complexité est paramétré par $q \in \{1, 2, \infty\}$, suivant la forme de régularisation considérée. Le choix de q définit une approche spécifique de régularisation, qui dépend des propriétés attendues du système. Ainsi, à partir de la classe de modèles hybrides régularisés $\mathcal{F}$ définie en (3.6), on développe dans la section suivante des bornes pour l'identification des systèmes hybrides.

3.2 Bornes sur l'erreur de généralisation pour l'identification des systèmes hybrides

Pour obtenir des bornes sur l'erreur de généralisation pour l'identification des systèmes hybrides, il faut étudier la complexité de la classe de fonctions de perte

$$\mathcal{L}_{\mathcal{F}} = \{\ell \in [0, 4M^2]^{\mathcal{Z}} : \ell(\boldsymbol{z}) = \min_{j \in [C]} |y_k - \overline{f}_j(\boldsymbol{x}_k)|^p, \boldsymbol{f} \in \mathcal{F}\}, \tag{3.7}$$

où $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ avec $\mathcal{Y} = [-M, M]$ et $\overline{f}_j$ est le j-ème sous-modèle tronqué (voir la Définition 2). Dans Lauer (2020a), le cas où $\mathcal{F} = \prod_{j=1}^{C} \mathcal{F}_j$ est un produit de classes de fonctions composantes indépendantes est étudié et la complexité de (3.7) est décomposée en fonction de la complexité de Rademacher empirique des classes de fonctions $\mathcal{F}_j,\ j = 1, \dots, C$.

Lemme 5 (d'après le Théorème 3 de Lauer (2020a)) *Soit une classe $\mathcal{F} = \prod_{j=1}^{C} \mathcal{F}_j$ et $\mathcal{L}_{\mathcal{F}}$ la classe de fonctions de perte (3.7). Alors,*

$$\hat{\mathcal{R}}_{\mathbf{Z}_N}(\mathcal{L}_{\mathcal{F}}) \leq p(2M)^{p-1} \sum_{j=1}^{C} \hat{\mathcal{R}}_{\mathbf{X}_N}(\mathcal{F}_j). \tag{3.8}$$

Le Lemme 5 permet alors d'obtenir une borne pour $\mathcal{F}$ en (3.6) dans le cas $q = \infty$, que l'on peut réécrire comme

$$\mathcal{F} = \prod_{j=1}^{C} \left\{ f_j : f_j(\boldsymbol{x}) = \boldsymbol{\theta}_j^T \boldsymbol{x}, \|\boldsymbol{\theta}_j\|_2 \leq \Lambda \right\}. \tag{3.9}$$

Dans le cas statique, cela donne le résultat suivant.

Théorème 6 *Soit une classe de fonctions composantes* $\mathcal{F} = \prod_{j=1}^{C} \mathcal{F}_j$*, avec* $\mathcal{F}_j$ *des classes de fonctions linéaires comme en* (3.9)*, et la classe de fonctions de perte* $\mathcal{L}_\mathcal{F}$ *définie en* (3.7)*, on a, avec une probabilité d'au moins* $1-\delta$*, et pour tout* $\boldsymbol{f} \in \mathcal{F}$*,*

$$L(\overline{\boldsymbol{f}}) \leq \hat{L}_N(\overline{\boldsymbol{f}}) + \frac{2p(2M)^{p-1}C\Lambda\sqrt{\sum_{k=1}^{N} \|\boldsymbol{X}_k\|^2}}{N} + 12M^2\sqrt{\frac{\log\frac{2}{\delta}}{2N}}.$$

Preuve du Théorème 6: Cette borne est obtenue par l'application du Théorème 3. Dans un premier temps, la complexité de Rademacher de $\mathcal{L}_\mathcal{F}$ est exprimée en fonction de celle des $\mathcal{F}_j$ en utilisant le Lemme 5. Ensuite, borner la complexité des classes de fonctions composantes linéaires avec le Théorème 11 de l'Annexe A complète la preuve. □

On a ici une borne dont la complexité croît linéairement en fonction du nombre de modes C, et qui décroît avec un taux de $1/\sqrt{N}$.

L'objectif est d'adapter cette borne au cas dynamique, tout en considérant différentes formes de régularisations. Sur la base des outils introduits en Section 2.5, et en considérant à présent la forme régularisée de notre classe de modèle $\mathcal{F}$ définie en (3.1)–(3.6) pour des valeurs de $q \neq \infty$, on présente le théorème suivant :

Théorème 7 *On considère* $\mathcal{F}$ *une classe de fonctions à valeurs vectorielles avec* C *sous-modèles comme dans* (3.6) *et la fonction de perte* $\ell_{\min}$ (1.31)*. Pour tout échantillon* $\mathbf{Z}_N = ((\boldsymbol{X}_k, Y_k))_{1\leq k\leq N} \in (\mathbb{R}^d \times \mathcal{Y})^N$ *issu d'un processus* β*-mélangeant, et pour tout* $\mu, a > 0$ *avec* $2\mu a = N$ *et* $\delta > 4(\mu-1)\beta(a)$*, avec une probabilité d'au moins* $1-\delta$*, l'inégalité suivante sur le risque à commutations* (3.3) *est valide pour tout* $\boldsymbol{f} \in \mathcal{F}$ *:*

$$L(\overline{\boldsymbol{f}}) \leq \hat{L}_N(\overline{\boldsymbol{f}}) + 2p(2M)^{p-1}\alpha(C,q)\frac{\Lambda\sqrt{\sum_{k=1}^{\mu} \|\boldsymbol{X}_{2a(k-1)+1}\|_2^2}}{\mu} + 3(2M)^p\sqrt{\frac{\log\frac{4}{\delta'}}{2\mu}},$$

où $\delta' = \delta - 4(\mu - 1)\beta(a)$ *et*

$$\alpha(C, q) = \begin{cases} C, & si\ q = \infty, \\ \frac{q}{q-1} C^{1-\frac{1}{q}}, & si\ 1 < q < \infty, \\ 1 + \log C, & si\ q = 1, \\ \frac{1}{1-q}, & si\ 0 < q < 1. \end{cases} \tag{3.10}$$

Le Théorème 7 est obtenu par une analyse de la complexité de Rademacher des systèmes hybrides à commutation arbitraire inspirée des résultats présentés dans Lauer (2020b), et par l'extension de la théorie de l'apprentissage à l'identification des systèmes en se servant du Théorème 4.

Pour ce faire, on utilise des fonctions de perte invariantes aux permutations. Cela implique que la permutation de composantes $\pi(j)$ au sein de $\ell_{\min}$ n'impacte pas la fonction de perte : pour toute permutation $\pi : [C] \to [C]$,

$$\ell_{\min}(\boldsymbol{x}, y, \boldsymbol{f}) = \min_{j \in [C]} |y_k - \overline{f}_j(\boldsymbol{x}_k)|^p = \ell_{\min}(\boldsymbol{x}, y, (f_{\pi(j)})_{1 \le j \le C}). \tag{3.11}$$

Alors, on a $L(\boldsymbol{f}) = L(\tilde{\boldsymbol{f}})$ pour n'importe quel modèle $\tilde{\boldsymbol{f}}$ identique à $\boldsymbol{f}$ à une permutation de ses composantes près. Cela permet de plutôt borner $L(\tilde{\boldsymbol{f}})$ pour une classe de fonctions permutées $\tilde{\mathcal{F}}$ et ainsi faire apparaître une structure plus exploitable. En particulier, $\tilde{\mathcal{F}}$ peut être inclue dans un produit de classes indépendantes Π_q sur lequel on peut ensuite appliquer le Lemme 5.

Preuve du Théorème 7 : On considère la fonction de perte $\ell_{\min}(\boldsymbol{x}, y, \boldsymbol{f})$ invariante aux permutations, ce qui permet de définir une classe de fonctions ordonnées $\tilde{\mathcal{F}}$, dont les composantes des modèles $\tilde{\boldsymbol{f}}$ sont ordonnées en fonction de leur complexité décroissante :

$$\forall \boldsymbol{f} = (f_j)_{1 \le j \le C}, \quad \tilde{\boldsymbol{f}} = (f_{\pi(j)})_{1 \le j \le C}, \tag{3.12}$$

où $\pi(j)$ est le j-ème élément d'une permutation de [C] qui assure que

$$\|\boldsymbol{\theta}_{\pi(1)}\|_2 \ge \|\boldsymbol{\theta}_{\pi(2)}\|_2 \cdots \ge \|\boldsymbol{\theta}_{\pi(C)}\|_2. \tag{3.13}$$

On considère donc une classe de fonctions ordonnées

$$\tilde{\mathcal{F}} = \{\tilde{\boldsymbol{f}} : \boldsymbol{f} \in \mathcal{F}\}. \tag{3.14}$$

En appliquant le Théorème 12 en Annexe A on a

$$\tilde{\mathcal{F}} \subseteq \Pi_q = \prod_{j=1}^{C} \left\{ f_j : f_j(\boldsymbol{x}) = \boldsymbol{\theta}_j^T \boldsymbol{x},\ \|\boldsymbol{\theta}_j\|_2 \leq j^{-\frac{1}{q}} \Lambda \right\}, \tag{3.15}$$

et donc $\mathcal{L}_{\tilde{\mathcal{F}}} \subseteq \mathcal{L}_{\Pi_q}$, ce qui engendre

$$\hat{\mathcal{R}}_{\mathbf{Z}_n}(\mathcal{L}_{\tilde{\mathcal{F}}}) \leq \hat{\mathcal{R}}_{\mathbf{Z}_n}(\mathcal{L}_{\Pi_q}). \tag{3.16}$$

Comme Π_q est le produit de classes de composantes indépendantes, le résultat de décomposition du Lemme 5 donne alors

$$\hat{\mathcal{R}}_{\mathbf{Z}_N}(\mathcal{L}_{\tilde{\mathcal{F}}}) \leq p(2M)^{p-1} \sum_{j=1}^{C} \hat{\mathcal{R}}_{\mathbf{X}_N} \left(\left\{ f_j : f_j(\boldsymbol{x}) = \boldsymbol{\theta}_j^T \boldsymbol{x},\ \|\boldsymbol{\theta}_j\|_2 \leq j^{-\frac{1}{q}} \Lambda \right\} \right). \tag{3.17}$$

Appliquer le Théorème 11 de l'Annexe A donne

$$\hat{\mathcal{R}}_{\mathbf{X}_N} \left(\left\{ f_j : f_j(\boldsymbol{x}) = \boldsymbol{\theta}_j^T \boldsymbol{x},\ \|\boldsymbol{\theta}_j\|_2 \leq j^{-\frac{1}{q}} \Lambda \right\} \right) \leq \frac{\Lambda}{N} \sqrt{\sum_{k=1}^{N} \|\mathbf{X}_k\|_2^2}, \tag{3.18}$$

et finalement, comme décrit dans l'Annexe B, $\sum_{j=1}^{C} j^{\frac{-1}{q}} \leq \alpha(C, q)$ pour tout $C \geq 2$ et $q \in (0, \infty]$, ce qui complète la preuve. □

Dans la borne présentée dans le Théorème 7, les effets des différentes approches de régularisation sont portés par $\alpha(C, q)$.

— Le cas où $q = \infty$ correspond au cas où l'on considère tous les sous-modèles indépendants entre eux et la complexité globale $\|\Omega(\boldsymbol{f})\|_\infty$ correspond à la plus grande des complexités des sous-modèles indépendamment de leur nombre ;

— Le cas où $q = 2$ représente le scénario le plus commun, où la dépendance entre les sous-modèles est considérée. Dans ce cas, si la norme d'un des sous-modèles $\|f_j\|_2$ augmente,

cela aura pour effet de faire diminuer la complexité d'un autre sous-modèle pour respecter la contrainte $\|\Omega(\boldsymbol{f})\|_2 \leq \Lambda$;

— Pour $q = 1$, on favorise les modèles parcimonieux avec quelques vecteurs de paramètres $\boldsymbol{\theta}_j$ non nuls, ce qui implique une faible dépendance à C pour la borne.

Comme l'illustre l'Exemple 2, considérer la régularisation permet de contrôler l'impact du nombre de modes C sur la borne. Aussi, la borne décroît ici en fonction de μ plutôt que N. Le choix de a pour la création des blocs indépendants a un fort impact sur la borne, qui décroît donc $2a$ fois moins vite dans le cadre non-i.i.d.

Exemple 2 *Soit une classe de fonctions $\mathcal{F}$ définie comme en (3.6), et la classe de fonctions de perte (3.7). Dans le cas où $p = 2,\ q = 2,\ M = 1/2$, on a $\alpha(C, q) = 2\sqrt{2}$. La borne suivante est alors obtenue :*

$$L(\overline{\boldsymbol{f}}) \leq \hat{L}(\overline{\boldsymbol{f}}) + 8\sqrt{C}\frac{\Lambda\sqrt{\sum_{k=1}^{\mu} \|\boldsymbol{X}_{2a(k-1)+1}\|_2^2}}{\mu} + 3\sqrt{\frac{\log \frac{4}{\delta'}}{2\mu}}, \tag{3.19}$$

avec une dépendance à C sous linéaire grâce à la régularisation par $\|\Omega(\boldsymbol{f})\|_2$.

Ici, C est considéré comme une constante prédéfinie. Dans la section suivante, on rend la borne du Théorème 7 uniforme par rapport à cet hyperparamètre afin de l'utiliser comme critère de sélection de modèle pour l'identification de systèmes hybrides.

3.3 Borne uniforme par rapport à C pour la sélection de modèles

Comme discuté dans le Chapitre 1, un problème critique de l'identification des systèmes hybrides est l'estimation du nombre de modes. Il s'agit d'un problème de sélection de modèle que nous proposons d'aborder avec une approche issue de la théorie de l'apprentissage.

Le principe de minimisation du risque structurel consiste à minimiser l'erreur de généralisation par rapport à certains hyperparamètres. Dans notre cas, l'hyperparamètre est le nombre de modes C. Formellement, nous considérons le schéma général de l'Algorithme 2, dans lequel nous supposons que nous avons accès à un algorithme générique d'identification de systèmes à commutations travaillant avec un nombre de modes C fixé pour estimer $\boldsymbol{f} \in \mathcal{F}$ et $\boldsymbol{r} \in [C]^N$. Dans ce schéma, l'algorithme générique est appliqué pour tous les nombres de modes C dans une plage

prédéfinie. Ensuite, le « meilleur » modèle est sélectionné sur la base d'un critère $J(C)$ conçu ci-dessous à partir de la borne sur l'erreur de généralisation développée dans le Théorème 8.

Algorithm 2: Méthode d'estimation du nombre de modes à base de minimisation du risque structurel

Entrées: Les données $((\boldsymbol{x}_k, y_k))_{1\leq k\leq N} \in (\mathbb{R}^d \times \mathcal{Y})^N$ et un nombre maximum de sous-modèles $\overline{C}$

Pour $C = 1$ *à* $\overline{C}$ **Faire**

Estimer $\boldsymbol{f}$ avec C modes en utilisant l'Algorithme 1 (ou tout autre algorithme travaillant à C fixé)

Calculer

$$J(C) = \hat{L}_N(\overline{\boldsymbol{f}}) \;+\; 2p(2M)^{p-1}\alpha(C,q)\tilde{\Lambda}(\boldsymbol{f})\frac{\sqrt{\sum_{k=1}^{\mu}\|\boldsymbol{x}_{2a(k-1)+1}\|_2^2}}{\mu}.$$

Déterminer $\hat{C} = \operatorname{argmin}_{C\in[\overline{C}]} J(C)$.

Sorties: $\hat{C}$ et le modèle $\boldsymbol{f}$ correspondant

Étant donné que le fonctionnement de l'Algorithme 2 nécessite le calcul d'une borne pour tout $C \in [\overline{C}]$ sur un même jeu de données, il est nécessaire que la borne soit valide uniformément pour chaque valeur de C. De plus, la borne présentée dans le Théorème 7 est valide avec un Λ défini a priori pour la classe de modèles (3.6), tandis que la plupart des algorithmes d'identification de systèmes hybrides (comme l'Algorithme 1 vu en Section 1.5.2) n'imposent pas ce type de contrainte sur les vecteurs de paramètres $\boldsymbol{\theta}_j$. Pour combler ce vide entre la théorie et la pratique, il faudrait alors proposer des bornes où le rayon Λ serait calculé a posteriori comme

$$\Lambda(\boldsymbol{f}) = \|\Omega(\boldsymbol{f})\|_q \tag{3.20}$$

à partir du modèle $\boldsymbol{f}$ estimé. Plus précisément, on considère une valeur discrétisée de ce rayon :

$$\tilde{\Lambda}(\boldsymbol{f}) = \min\left\{\Lambda \in \{\Lambda_1, \ldots, \Lambda_I\} \;:\; \Lambda \geq \|\Omega(\boldsymbol{f})\|_q\right\}. \tag{3.21}$$

En d'autres termes, on considère une grille de I valeurs possibles pour Λ, et $\tilde{\Lambda}(\boldsymbol{f})$ représente le premier point de la grille supérieur à la vraie valeur de $\Lambda(\boldsymbol{f})$.

De ce fait, en calculant $\tilde{\Lambda}(\boldsymbol{f})$ a posteriori, on peut considérer un Λ très grand et peu contraignant dans (3.6).

Théorème 8 *Soit un nombre de sous-modèles maximal $\overline{C}$ fixé, une classe de modèles $\mathcal{F}$ définie comme en (3.6) pour une valeur maximale et prédéfinie de Λ, ainsi qu'une grille $\{\Lambda_1, \ldots, \Lambda_I\}$ de I valeurs définie en (3.21) telle que $\Lambda_I = \Lambda$. Pour un échantillon $\mathbf{Z}_N$ de taille N issu d'un processus β-mélangeant, et pour tout $\mu, a > 0$ avec $2\mu a = N$, et $\delta > 4\overline{C}I(\mu - 1)\beta(a)$, avec une probabilité d'au moins $1 - \delta$, pour tout $C \in [\overline{C}]$ et pour tout $\boldsymbol{f} \in \mathcal{F}$,*

$$L(\overline{\boldsymbol{f}}) \leq \hat{L}_N(\overline{\boldsymbol{f}}) + \frac{2p(2M)^{p-1}\alpha(C,q)\tilde{\Lambda}(\boldsymbol{f})\sqrt{\sum_{k=1}^{\mu} \|\boldsymbol{X}_{2a(k-1)+1}\|_2^2}}{\mu} + 3(2M)^p\sqrt{\frac{\log(\overline{C}I) + \log\frac{4}{\delta'}}{2\mu}}, \tag{3.22}$$

où $\delta' = \delta - 4\overline{C}I(\mu - 1)\beta(a)$ et $\tilde{\Lambda}(\boldsymbol{f})$ est calculé comme en (3.21).

Preuve du Théorème 8: Soit $\mathcal{F}_i$ définie comme en (3.6), avec Λ_i à la place de Λ. Pour tout C fixé, Λ_i et $\delta_0' > 0$, on définit $\delta_0 = \delta_0' + 4(\mu - 1)\beta(a)$ et

$$\epsilon(C, \Lambda_i, \delta_0') = \frac{2p(2M)^{p-1}\alpha(C,q)\Lambda_i\sqrt{\sum_{k=1}^{\mu} \|\boldsymbol{X}_{2a(k-1)+1}\|_2^2}}{\mu} + 3(2M)^p\sqrt{\frac{\log\frac{4}{\delta_0'}}{2\mu}}.$$

Alors, le Théorème 7 indique que

$$\mathbb{P}\left\{\exists \boldsymbol{f} \in \mathcal{F}_i,\ L(\overline{\boldsymbol{f}}) > \hat{L}_N(\overline{\boldsymbol{f}}) + \epsilon(C, \Lambda_i, \delta_0')\right\} \leq \delta_0.$$

De ce fait, en appliquant la borne de l'union et $\overline{C}I$ fois le Théorème 7 avec l'indice de confiance δ_0, on a

$$\begin{aligned}
&\mathbb{P}\left\{\exists C \in [\overline{C}], i \in [I],\ \boldsymbol{f} \in \mathcal{F}_i, L(\overline{\boldsymbol{f}}) \geq \hat{L}_N(\overline{\boldsymbol{f}}) + \epsilon(C, \Lambda_i, \delta_0')\right\} \\
&\quad \leq \sum_{C=1}^{\overline{C}}\sum_{i=1}^{I} \mathbb{P}\left\{\exists \boldsymbol{f} \in \mathcal{F}_i,\ L(\overline{\boldsymbol{f}}) \geq \hat{L}_N(\overline{\boldsymbol{f}}) + \epsilon(C, \Lambda_i, \delta_0')\right\} \\
&\quad \leq \overline{C}I\delta_0.
\end{aligned}$$

Ensuite, pour tout $\delta = \delta' + 4\overline{C}I(\mu - 1)\beta(a)$ avec $\delta' > 0$, on définit $\delta_0' = \delta'/\overline{C}I$, ce qui donne

finalement $\delta_0 = \delta/\overline{C}I$ et donc

$$\begin{aligned}
&\mathbb{P}\left\{\forall C \in [\overline{C}],\ i \in [I],\ \boldsymbol{f} \in \mathcal{F}_i,\ L(\overline{\boldsymbol{f}}) \leq \hat{L}_N(\overline{\boldsymbol{f}}) + \epsilon(C, \Lambda_i, \delta_0')\right\} \\
&= 1 - \mathbb{P}\left\{\exists C \in [\overline{C}], i \in [I],\ \boldsymbol{f} \in \mathcal{F}_i,\ L(\overline{\boldsymbol{f}}) > \hat{L}_N(\overline{\boldsymbol{f}}) + \epsilon(C, \Lambda_i, \delta_0')\right\} \\
&\geq 1 - \overline{C}I\delta_0 = 1 - \delta
\end{aligned} \tag{3.23}$$

avec

$$\epsilon(C, \Lambda_i, \delta_0') = \frac{2p(2M)^{p-1}\alpha(C,q)\Lambda_i\sqrt{\sum_{k=1}^{\mu}\|\boldsymbol{X}_{2a(k-1)+1}\|_2^2}}{\mu} + 3(2M)^p\sqrt{\frac{\log(\overline{C}I) + \log\frac{4}{\delta'}}{2\mu}}.$$

Pour tout $C \in [\overline{C}]$, on peut réécrire $\mathcal{F}$ en (3.6) avec la contrainte $\|\Omega(\boldsymbol{f})\|_q \leq \Lambda$ comme $\mathcal{F} = \bigcup_{i\in[I]}\mathcal{F}_i$; et tout $\boldsymbol{f} \in \mathcal{F}$ appartient alors aussi à $\mathcal{F}_i$ avec, par l'Équation (3.21), i tel que $\Lambda_i = \tilde{\Lambda}(\boldsymbol{f})$. Ainsi, (3.23) est équivalent à (3.22), ce qui complète la preuve. □

Le Théorème 8 permet ensuite de formuler le critère $J(C)$ nécessaire pour la méthode de sélection de modèle de l'Algorithme 2 en y appliquant cette borne. Il est cependant à noter que l'on peut se passer de prendre en compte les termes constants ne dépendant ni de C, ni du modèle $\boldsymbol{f}$ estimé dans la borne (3.22). On obtient alors

$$\begin{aligned}
J(C) &= \hat{L}_N(\overline{\boldsymbol{f}}) + \epsilon(\boldsymbol{f}, C), \\
\epsilon(\boldsymbol{f}, C) &= \frac{2p(2M)^{p-1}\alpha(C,q)\tilde{\Lambda}(\boldsymbol{f})\sqrt{\sum_{k=1}^{\mu}\|\boldsymbol{X}_{2a(k-1)+1}\|_2^2}}{\mu}
\end{aligned} \tag{3.24}$$

avec a et μ tels que $2\mu a = N$.

Ainsi, le critère $J(C)$ est composé du risque empirique $\hat{L}_N(\overline{\boldsymbol{f}})$, ainsi que de l'intervalle de confiance $\epsilon(\boldsymbol{f}, C)$. Ce critère représente la partie droite de la borne (3.22), privée des termes constants n'influençant pas la sélection de modèle. De ce fait, l'algorithme ne dépend pas du coefficient de mélange $\beta(a)$ ni du choix de I dans la grille des Λ_i.

Dans l'Exemple 3 ci-dessous, on applique notre méthode d'estimation du nombre de modes (Algorithme 2) en utilisant l'algorithme k-LinReg (Algorithme 1) à chaque itération.

Exemple 3 *On considère un système hybride composé de $C = 3$ sous-modèles linéaires d'ordre*

$n_a = n_b = 2$ *avec pour vecteurs de paramètres*

$$\begin{aligned} \boldsymbol{\theta}_1 &= [-0.4 \quad 0.25 \quad -0.15 \quad 0.08]^T, \\ \boldsymbol{\theta}_2 &= [1.55 \quad -0.58 \quad -2.1 \quad 0.96]^T, \\ \boldsymbol{\theta}_3 &= [1 \quad -0.24 \quad -0.65 \quad 0.30]^T. \end{aligned} \tag{3.25}$$

On génère à partir de ce système un jeu de données de taille $N = 80\ 000$ *points en respectant les conditions suivantes. Le signal d'excitation* u_k *est un signal gaussien de moyenne nulle et de variance unitaire. Le bruit* e_k *est un bruit blanc gaussien dont l'amplitude respecte un ratio signal sur bruit de* $30dB$ *par rapport au signal de sortie. Le mode actif* r_k *est distribué uniformément dans* $\{1, 2, 3\}$.

En appliquant l'Algorithme 2, avec $p = 2$ *et* $q = \infty$, *on obtient une courbe représentant le critère* $J(C)$ *dans la Figure 3.1, où l'on peut voir qu'un minimum est atteint en* $C = 3$. *De manière plus détaillée, on peut observer que l'intervalle de confiance*

$$\epsilon(\boldsymbol{f}, C) = \frac{4C\tilde{\Lambda}(\boldsymbol{f})\sqrt{\sum_{k=1}^{\mu} \|\boldsymbol{X}_{2a(k-1)+1}\|_2^2}}{\mu} \tag{3.26}$$

croit de manière linéaire par rapport au nombre de modes considéré. À l'opposé, l'erreur empirique décroit de manière monotone. Enfin, la somme de ces deux courbes nous donne $J(C)$, *avec un minimum indiquant le bon nombre de modes.*

3.4 Conclusion

Dans ce chapitre, les premières contributions de la thèse ont été présentées. On a montré ici qu'il est tout a fait possible et pertinent d'appliquer la théorie statistique de l'apprentissage pour l'identification de systèmes dynamiques, et notamment pour l'identification des systèmes hybrides. Ainsi, après avoir développé une borne sur l'erreur de généralisation valide pour ce cadre d'étude, on l'emploie en appliquant le principe de minimisation du risque structurel afin d'estimer le nombre de modes des systèmes hybrides. Cette méthode est valide quels que soit la structure du modèle, le type de bruit ou encore l'algorithme d'identification utilisé à l'intérieur

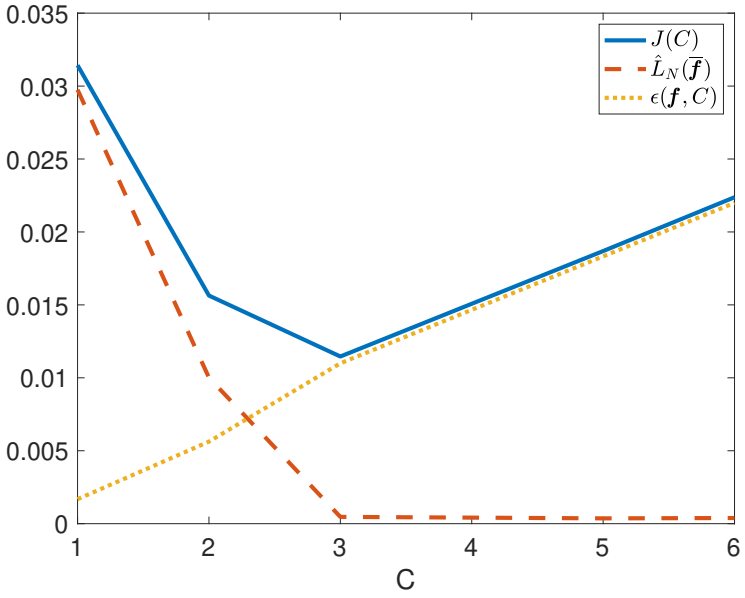

FIGURE 3.1 – Illustration de la méthode d'estimation de C dans l'Exemple 3. La courbe continue en bleu représente le critère $J(C)$. La courbe en tirets oranges représentent le risque empirique $\hat{L}_N(\overline{\boldsymbol{f}})$. La courbe en pointillés jaunes montre l'intervalle de confiance $\epsilon(\boldsymbol{f}, C)$ (3.26).

de la boucle de l'Algorithme 2. Ici, la connaissance du coefficient de mélange $\beta(a)$ n'est pas nécessaire pour l'estimation du nombre de modes car il apparaît au sein d'un terme constant dans la borne (3.22). Il sera néanmoins nécessaire de calculer ce paramètre pour utiliser ces bornes comme des garanties sur le modèle estimé. Pour obtenir la borne à la base de cette méthode, nous avons considéré une forme régularisée des classes de fonctions que l'on étudie. Les aspects algorithmiques liés à l'apprentissage régularisé des modèles à commutations sont étudiés dans le chapitre suivant.

Chapitre 4

Apprentissage de modèles hybrides régularisés

Dans ce chapitre, on développe de nouveaux algorithmes d'optimisation afin de prendre en compte la régularisation dans l'identification des systèmes à commutations arbitraires. Pour ce faire, on considère dans notre apprentissage, comme introduit dans la Section 2.3, un compromis entre l'erreur d'apprentissage $\mathcal{E}(\boldsymbol{\theta}, \mathbf{X}, \boldsymbol{y})$ et un terme de régularisation $\Gamma(\boldsymbol{\theta})$ en respectant le cadre d'étude associé aux systèmes hybrides. Cela conduit à poser le problème d'apprentissage comme un problème d'optimisation de forme générale :

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{Cd}} \mathcal{E}(\boldsymbol{\theta}, \mathbf{X}, \boldsymbol{y}) + \lambda \Gamma(\boldsymbol{\theta}), \tag{4.1}$$

avec $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_C^T]^T \in \mathbb{R}^{Cd}$ le vecteur de paramètres du modèle composé de C modes en dimension d et $\lambda > 0$ un hyperparamètre à régler pour contrôler le compromis entre les deux termes.

À partir de là, il existe de multiples façons de régulariser en choisissant différentes méthodes pour calculer l'erreur, ainsi que la complexité du modèle.

Dans le cadre de l'identification des systèmes hybrides, le terme $\mathcal{E}(\boldsymbol{\theta}, \mathbf{X}, \boldsymbol{y})$ prend généralement la forme de (1.31) :

$$\mathcal{E}(\boldsymbol{\theta}, \mathbf{X}, \boldsymbol{y}) = \sum_{k=1}^{N} \min_{j \in [C]} |y_k - \boldsymbol{\theta}_j^T \boldsymbol{x}_k|^p, \tag{4.2}$$

avec $p \in \{1, 2\}$ permettant de choisir entre l'erreur absolue et l'erreur quadratique.

Pour ce qui est de la régularisation, on considère dans ce chapitre

$$\Gamma(\boldsymbol{\theta}) = \|\Omega(\boldsymbol{\theta})\|_q, \tag{4.3}$$

$$\text{avec} \quad \Omega(\boldsymbol{\theta}) = [\|\boldsymbol{\theta}_1\|_2, \ \ldots, \ \|\boldsymbol{\theta}_C\|_2]^T, \tag{4.4}$$

correspondant au terme mesurant la complexité (3.5) de la classe de fonctions (3.6) dans le chapitre précédent.

Ainsi, on choisit de traiter le problème d'identification de systèmes hybrides régularisés (4.1), que l'on caractérise par la paire (p, q), dont les paramètres définissent respectivement la fonction de perte et le schéma de régularisation considérés.

Pour toute paire (p, q), (4.1) est un problème d'optimisation très difficile, en raison de l'opération de minimisation dans le calcul de l'erreur (4.2) qui exprime la spécificité des systèmes commutés d'une part, mais rend également la fonction objectif non convexe et même non lisse et non différentiable d'autre part. Dans la section suivante, des extensions de l'algorithme k-LinReg (Lauer, 2013), présenté dans le Chapitre 1, qui incluent les diverses formes de régularisation obtenues pour différentes valeurs de q, sont présentées.

4.1 Identification de systèmes hybrides régularisés

L'algorithme original k-LinReg présenté dans l'Algorithme 1 est une méthode efficace du point de vue du calcul pour obtenir des solutions approximatives au problème d'identification des systèmes commutés non régularisés pour $p = 2$. Pour rappel, la méthode proposée par k-LinReg consiste à alterner entre des étapes de classification pour estimer la séquence de modes $\boldsymbol{r} \in [C]^N$ et de régression pour estimer les paramètres $\boldsymbol{\theta}_j \in \mathbb{R}^d$. Ainsi l'étape de régression consiste à mettre à jour les vecteurs de paramètres $\boldsymbol{\theta}_j$ en minimisant

$$\sum_{k=1}^{N} (y_k - \boldsymbol{x}_k^T \boldsymbol{\theta}_j)^2 = \sum_{j=1}^{C} \sum_{k \in S_j} (y_k - \boldsymbol{x}_k^T \boldsymbol{\theta}_j)^2, \tag{4.5}$$

et où

$$S_j = \{k \in [N] : r_k = j\}, \tag{4.6}$$

est déterminé par la classification. C'est au niveau de l'étape de régression que la régularisation apporte des changements à l'algorithme d'identification. Cependant, la régularisation affecte l'algorithme de manière non triviale, en particulier sur le fait que les sous-modèles ne peuvent pas toujours être estimés de façon indépendante. Spécifiquement, l'étape de régression précédemment décrite par la minimisation de (4.5) doit maintenant résoudre (4.1)–(4.2) pour la classification fixée par les S_j :

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{Cd}} \sum_{j=1}^{C} \sum_{k\in S_j} |y_k - \boldsymbol{x}_k^T \boldsymbol{\theta}_j|^p + \lambda \|\Omega(\boldsymbol{\theta}\|_q. \tag{4.7}$$

Algorithm 3: k-LinReg régularisé

Entrées: le jeu de données $(\mathbf{X}, \boldsymbol{y}) \in \mathbb{R}^{N\times d} \times \mathbb{R}^N$, le choix du couple (p, q), le nombre de modes C, et les paramètres initiaux $\boldsymbol{\theta} \in \mathbb{R}^{Cd}$.

Répéter

Classer les points et estimer la séquence de commutation $\boldsymbol{r}$

Mettre à jour les paramètres $\boldsymbol{\theta}$ du modèle en résolvant (4.7)

Jusqu'à Convergence;

Sorties: $\hat{\boldsymbol{\theta}}$ et $\hat{\boldsymbol{r}}$

L'algorithme générique est alors donné par l'Algorithme 3 et les différentes approches proposées pour effectuer l'optimisation de (4.7) sont détaillées ci-dessous pour les différents choix de $q \in \{1, 2, \infty\}$.

4.1.1 Régularisation par la norme infinie ($q = \infty$)

Comme introduit dans le Chapitre 3, chaque valeur de q considérée réfère à un schéma particulier de régularisation. Pour $q = \infty$, on considère que tous les sous-modèles de $\boldsymbol{\theta}$ sont indépendants entre eux, et la complexité globale mesurée par $\|\Omega(\boldsymbol{\theta})\|_\infty$ correspond à la plus grande des complexités des sous-modèles, indépendamment de leur nombre.

Ici, avec $q = \infty$, l'étape de régression (4.7) revient à résoudre

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{Cd}} \sum_{j=1}^{C} \sum_{k\in S_j} |y_k - \boldsymbol{\theta}_j^T \boldsymbol{x}_k|^p + \lambda \max_{j\in[C]} \|\boldsymbol{\theta}_j\|_2, \tag{4.8}$$

qui est un problème d'optimisation convexe. Plus précisément, il peut s'écrire comme un problème de programmation conique du second ordre (SOCP pour *second order cone programming*). Par exemple, si $\boldsymbol{X}_j$ et $\boldsymbol{y}_j$ sont les sous-ensembles de lignes de $\boldsymbol{X}$ et $\boldsymbol{y}$ ayant un indice dans S_j, et que l'on définit $\boldsymbol{\xi}_j$ à partir du vecteur d'erreur $\boldsymbol{\xi} \in \mathbb{R}^N$ de la même manière, cela donne, pour $p = 1$ et la perte absolue,

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{Cd},\boldsymbol{\xi}\in\mathbb{R}^N,\tau\in\mathbb{R}} \mathbf{1}^T\boldsymbol{\xi} + \lambda\tau \tag{4.9}$$

$$\text{s.t. } -\boldsymbol{\xi}_j \leq \boldsymbol{y}_j - \mathbf{X}_j\boldsymbol{\theta}_j \leq \boldsymbol{\xi}_j, \quad j = 1,\dots,C \tag{4.10}$$

$$\|\boldsymbol{\theta}_j\|_2 \leq \tau, \quad j = 1,\dots,C. \tag{4.11}$$

Ici, pour $k \in S_j$, ξ_k est la k-ème entrée de $\boldsymbol{\xi}$ qui appartient également à $\boldsymbol{\xi}_j$ et correspond, à l'optimum, à la perte absolue pour le k-ème point de données affecté au j-ème mode :

$$\xi_k^* = |y_k - \boldsymbol{\theta}_j^T\boldsymbol{x}_k|. \tag{4.12}$$

D'un autre côté, à l'optimum, $\tau^* = \max_{j\in[C]} \|\boldsymbol{\theta}_j\|_2$ et (4.9) correspond bien à (4.8) pour $p = 1$.

Pour le cas de la perte quadratique où $p = 2$, une formulation similaire peut être obtenue :

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{Cd},\boldsymbol{\xi}\in\mathbb{R}^C,\tau\in\mathbb{R}} \mathbf{1}^T\boldsymbol{\xi} + \lambda\tau \tag{4.13}$$

$$\text{s.t. } \left\| \begin{bmatrix} (1-\xi_j)/2 \\ \boldsymbol{y}_j - \mathbf{X}_j\boldsymbol{\theta}_j \end{bmatrix} \right\|_2 \leq (1+\xi_j)/2, \quad j = 1,\dots,C \tag{4.14}$$

$$\|\boldsymbol{\theta}_j\|_2 \leq \tau, \quad j = 1,\dots,C, \tag{4.15}$$

où $\boldsymbol{\xi} \in \mathbb{R}^C$ contient maintenant une seule composante pour chaque mode. À l'optimum, ξ_j est minimisé, la contrainte de (4.13) est active et l'on obtient

$$\left\| \begin{bmatrix} (1-\xi_j^*)/2 \\ \boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\theta}_j \end{bmatrix} \right\|_2 = (1+\xi_j^*)/2. \tag{4.16}$$

Ainsi, on a

$$\left(\frac{1-\xi_j^*}{2}\right)^2 + \|\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\theta}_j\|_2^2 = \left(\frac{1+\xi_j^*}{2}\right)^2, \tag{4.17}$$

qui donne finalement

$$-\xi_j^* + \|\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\theta}_j\|_2^2 = 0, \tag{4.18}$$

$$\text{ou} \quad \xi_j^* = \|\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\theta}_j\|_2^2. \tag{4.19}$$

Ainsi, avec τ qui coïncide encore avec $\max_{j\in[C]} \|\boldsymbol{\theta}_j\|_2$ à l'optimum, (4.13) est bien équivalent à (4.8) pour $p = 2$.

4.1.2 Régularisation ℓ_1 $(q = 1)$

La régularisation avec $q = 1$ peut être utilisée pour favoriser les modèles parcimonieux en termes de nombre de sous-modèles : la minimisation de la norme ℓ_1 de $\Omega(\boldsymbol{\theta})$ est susceptible de donner une solution avec de nombreux zéros dans $\Omega(\boldsymbol{\theta})$, ce qui correspond à des vecteurs de paramètres dont la norme $\|\boldsymbol{\theta}_j\|_2$ est nulle et qui peuvent probablement être écartés du modèle hybride.

Formellement, cela conduit au problème d'optimisation

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{Cd}} \sum_{j=1}^{C} \sum_{k\in S_j} |y_k - \boldsymbol{\theta}_j^T \boldsymbol{x}_k|^p + \lambda \sum_{j=1}^{C} \|\boldsymbol{\theta}_j\|_2, \tag{4.20}$$

qui peut être décomposé en C sous-problèmes indépendants et convexes de forme

$$\min_{\boldsymbol{\theta}_j\in\mathbb{R}^d} \sum_{k\in S_j} |y_k - \boldsymbol{\theta}_j^T \boldsymbol{x}_k|^p + \lambda \|\boldsymbol{\theta}_j\|_2. \tag{4.21}$$

Bien que convexe, ce problème peut être difficile à résoudre en raison de la présence de la norme de $\boldsymbol{\theta}_j$ qui n'est pas considérée au carré. En pratique, il peut être défini comme un SOCP, similaire à (4.9) ou (4.13) avec les contraintes $\|\boldsymbol{\theta}_j\|_2 \leq \tau$ remplacées par $\|\boldsymbol{\theta}_j\|_2 \leq \tau_j$, $j = 1, \ldots, C$, et le terme $\lambda \sum_{j=1}^{C} \tau_j$ substitué à $\lambda\tau$ dans l'objectif. Par exemple, pour $p = 1$ cela

donne le problème d'apprentissage

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{Cd},\boldsymbol{\xi}\in\mathbb{R}^N,\tau\in\mathbb{R}} \mathbf{1}^T\boldsymbol{\xi} + \lambda\sum_{j=1}^{C}\tau_j \tag{4.22}$$

$$\text{s.t. } -\boldsymbol{\xi}_j \leq \boldsymbol{y}_j - \mathbf{X}_j\boldsymbol{\theta}_j \leq \boldsymbol{\xi}_j, \quad j=1,\ldots,C \tag{4.23}$$

$$\|\boldsymbol{\theta}_j\|_2 \leq \tau_j, \quad j=1,\ldots,C. \tag{4.24}$$

Nous proposons ci-dessous une alternative numériquement plus légère à ces SOCP.

Algorithme itératif de repondération

La solution de problèmes comme (4.21) peut être approchée par un schéma itératif, dans lequel on résout

$$\hat{\boldsymbol{\theta}}_j = \underset{\boldsymbol{\theta}_j\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{k\in S_j} |y_k - \boldsymbol{\theta}_j^T\boldsymbol{x}_k|^p + \lambda\eta_j\|\boldsymbol{\theta}_j\|_2^2 \tag{4.25}$$

avec η_j initialisé à 1 et mis à jour à chaque itération par

$$\eta_j = \frac{1}{\varepsilon + \|\hat{\boldsymbol{\theta}}_j\|_2}$$

sur la base de la solution courante et pour une petite valeur de ε empêchant une division par zéro. Le raisonnement est que, au fil des itérations, le terme $\eta_j\|\boldsymbol{\theta}_j\|_2^2$ tend vers la valeur souhaitée de $\|\boldsymbol{\theta}_j\|_2$. L'avantage par rapport à (4.21) est purement calculatoire. Par exemple, pour $p=2$, (4.25) est un problème de régression ridge standard dont la solution peut être facilement calculée par

$$\hat{\boldsymbol{\theta}}_j = (\mathbf{X}_j^T\mathbf{X}_j + \lambda\eta_j\mathbf{I})^{-1}\mathbf{X}_j^T\boldsymbol{y}_j, \tag{4.26}$$

où $\mathbf{X}_j$ et $\boldsymbol{y}_j$ correspondent aux sous-ensembles de lignes dans $\mathbf{X}$ et $\boldsymbol{y}$ avec un indice dans S_j, et $\mathbf{I}$ est la matrice identité de taille $d\times d$. De plus, pour les grands ensembles de données (grands N), la plupart des calculs dans (4.26) résident dans les produits $\mathbf{X}_j^T\mathbf{X}_j$ et $\mathbf{X}_j^T\boldsymbol{y}_j$, qui peuvent être précalculés pour donner des itérations de (4.25) très simples qui se résument à inverser une petite matrice $d\times d$.

4.1.3 Régularisation ℓ_2 $(q = 2)$

Pour $q = 2$, le problème (4.7) devient

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{Cd}} \sum_{j=1}^{C} \sum_{k\in S_j} |y_k - \boldsymbol{\theta}_j^T \boldsymbol{x}_k|^p + \lambda \left(\sum_{j=1}^{C} \|\boldsymbol{\theta}_j\|_2^2 \right)^{1/2}. \tag{4.27}$$

Encore une fois, il s'agit d'un problème convexe, mais plus exigeant sur le plan numérique que dans le cas $q = 1$, puisqu'il ne peut être découplé en C sous-problèmes indépendants en raison de la forme du terme de régularisation. Des formulations SOCP similaires à (4.9) ou (4.13) sont obtenues en remplaçant les contraintes $\|\boldsymbol{\theta}_j\|_2 \leq \tau$ par $\|\boldsymbol{\theta}_j\|_2 \leq \tau_j$, $j = 1, \ldots, C$, ainsi que $\|[\tau_1, \ldots, \tau_C]^T\|_2 \leq \tau$. Pour $p = 1$, cela donne :

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{Cd}, \boldsymbol{\xi}\in\mathbb{R}^N, \tau\in\mathbb{R}} \mathbf{1}^T \boldsymbol{\xi} + \lambda\tau \tag{4.28}$$

$$\text{s.t. } -\boldsymbol{\xi}_j \leq \boldsymbol{y}_j - \mathbf{X}_j\boldsymbol{\theta}_j \leq \boldsymbol{\xi}_j, \quad j = 1, \ldots, C \tag{4.29}$$

$$\|\boldsymbol{\theta}_j\|_2 \leq \tau_j, \quad j = 1, \ldots, C \tag{4.30}$$

$$\|[\tau_1, \ldots, \tau_C]^T\|_2 \leq \tau \tag{4.31}$$

En utilisant une approche de repondération itérative, une alternative est d'itérer sur

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\mathbb{R}^{Cd}}{\operatorname{argmin}} \sum_{j=1}^{C} \sum_{k\in S_j} |y_k - \boldsymbol{\theta}_j^T \boldsymbol{x}_k|^p + \lambda\eta \sum_{j=1}^{C} \|\boldsymbol{\theta}_j\|_2^2 \tag{4.32}$$

avec η initialisé à 1 et mis à jour à chaque itération par

$$\eta = \frac{1}{\varepsilon + \sqrt{\sum_{j=1}^{C} \|\hat{\boldsymbol{\theta}}_j\|_2^2}} = \frac{1}{\varepsilon + \|\hat{\boldsymbol{\theta}}\|_2}. \tag{4.33}$$

Contrairement au cas ℓ_1, la procédure itérative globale doit être exécutée simultanément pour tous les sous-modèles afin de calculer η à chaque itération. Cependant, à chaque itération, le problème d'optimisation (4.32) peut être décomposé en C sous-problèmes,

$$\hat{\boldsymbol{\theta}}_j = \underset{\boldsymbol{\theta}_j\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{k\in S_j} |y_k - \boldsymbol{\theta}_j^T \boldsymbol{x}_k|^p + \lambda\eta \|\boldsymbol{\theta}_j\|_2^2, \quad j = 1, \ldots, C, \tag{4.34}$$

qui sont similaires à (4.25), sauf que le même coefficient global η est utilisé au lieu des η_j individuels.

4.2 Optimisation et temps de calcul

Dans cette section, les deux approches d'optimisation reposant soit sur une formulation directe sous forme de SOCP, soit sur un schéma de repondération itérative (IR) sont comparées. Les expériences sont menées avec N points de données uniformément tirés dans $\boldsymbol{x}_k \in [-5, 5]^d$, pour lesquels la sortie est calculée par $y_k = \boldsymbol{x}_k^T \boldsymbol{\theta}_{r_k} + e_k$, avec une séquence de commutation aléatoire $\boldsymbol{r} \in [C]^N$, des paramètres aléatoires $\boldsymbol{\theta}_j \in [-1, 1]^d$ et un bruit gaussien e_k de moyenne nulle et de variance unitaire. Différentes tailles de problèmes N sont considérées pour évaluer la complexité des différentes approches par rapport à N.

La Figure 4.1 montre le rapport entre le temps de calcul nécessaire pour résoudre le SOCP à l'aide du logiciel d'optimisation MOSEK (MOSEK ApS, 2019) et celui du schéma IR pour $p = 2$ et $q \in \{1, 2\}$, tracé par rapport au nombre de données N pour $C = 3$ et $d = 3$. Tous ces résultats sont obtenus avec Matlab et un ordinateur équipé d'un processeur i7 de 4 cœurs de 1.7GHz et 16 GB de mémoire. Dans tous les cas, une seule initialisation de k-LinReg est considérée avec $\lambda = 1$ et le schéma IR est arrêté après 5 itérations.

Ces résultats montrent que, pour $p = 2$, les schémas IR peuvent offrir des accélérations significatives, jusqu'à des facteurs de quelques centaines, par rapport à une optimisation directe. Des accélérations similaires ont été observées pour différentes valeurs de C et d. Cependant, pour le cas $p = 1$, le problème (4.25) reste un programme quadratique, avec des contraintes linéaires en utilisant la perte absolue, et pour lequel aucune formule explicite telle que (4.26) n'est disponible. Dans ce cas, le schéma IR itérant sur (4.25), bien que supprimant les contraintes coniques, s'est avéré plus lent que l'optimisation directe de (4.21). En ce qui concerne l'écart d'optimalité relatif induit par les approximations IR, celui-ci est mesuré par le rapport

$$\frac{obj_{IR} - obj_{SOCP}}{obj_{SOCP}}, \tag{4.35}$$

où $obj_{méthode}$ correspond à la valeur de la fonction objectif de (4.7). Cette dernière est calculée

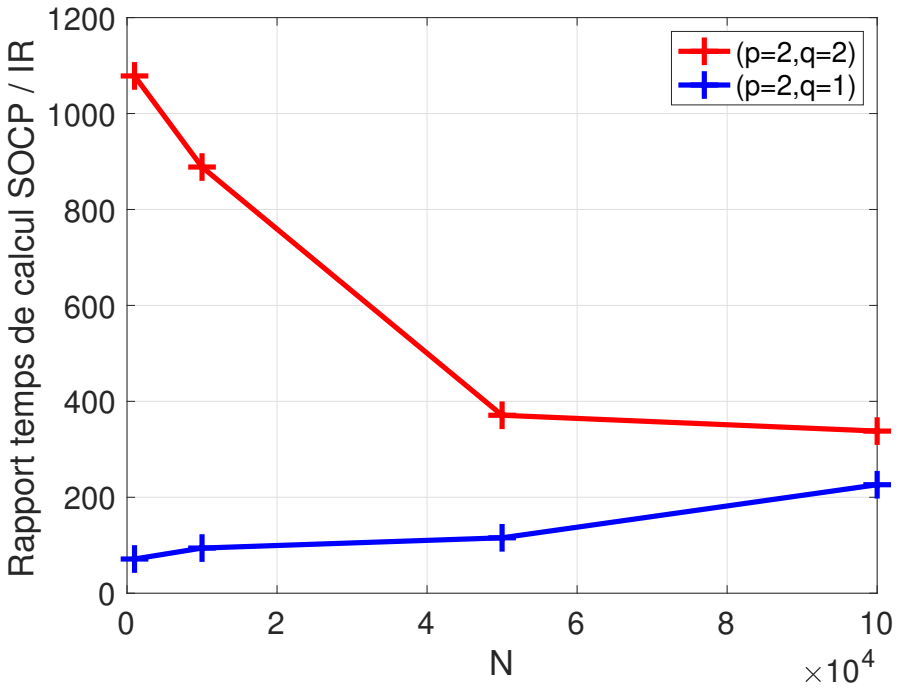

FIGURE 4.1 – Rapport du temps de calcul de l'optimisation directe des SOCP et des schémas IR en fonction du nombre de données N.

avec la valeur de $\boldsymbol{\theta}$ estimée par la méthode. Le résultat, inférieur à 1% dans tous nos tests, est satisfaisant. Suite à ces observations, dans toutes les expériences du Chapitre suivant, nous utiliserons toujours les schémas IR pour $p = 2$.

4.3 Conclusion

Dans ce chapitre, de nouveaux algorithmes d'optimisation permettant de prendre en compte la régularisation dans l'identification de systèmes à commutations arbitraires ont été présentés. Différents schémas de régularisation sont étudiés. On y distingue deux types d'approche. La première est une approche directe employant le logiciel d'optimisation générique MOSEK, tandis que la seconde repose sur une méthode de repondération itérative. Cette dernière présente l'avantage, dans le cas où l'on considère la perte quadratique ($p = 2$), d'accélérer considérablement le temps de calcul de l'algorithme d'identification.

Dans le chapitre suivant, on propose d'évaluer les performances de la méthode de sélection de

modèle introduite au Chapitre 3. Dans cette étude, on évaluera également les différents schémas de régularisation proposés dans ce chapitre.

Chapitre 5

Expériences numériques

Dans le Chapitre 1, la présentation des méthodes d'identification de systèmes hybrides met en lumière l'importance du réglage du paramètre C définissant le nombre de mode. Pour maîtriser ce paramètre, il existe différents types de méthodes comme les méthodes algébriques, introduites en Section 1.5.2, ou la méthode SRM (Massucci et al., 2022) proposée et présentée au Chapitre3. Dans ce chapitre, on propose de poursuivre l'analyse dans l'étude de l'estimation du nombre de modes pour l'identification des systèmes hybrides en proposant une analyse comparative de ces méthodes. Plus particulièrement, l'objectif est de caractériser de manière précise dans quel champ d'application il convient d'utiliser une méthode plutôt qu'une autre. Ainsi, la Section 5.1 rappelle le fonctionnement des différentes méthodes d'estimation du nombre de modes considérés. Dans la Section 5.2, on présente le plan d'expérience, et de premières observations sont présentées dans la Section 5.3. Les principaux résultats sont analysés et discutés dans la Section 5.4. Enfin, la Section 5.5 conclue ce chapitre.

5.1 Méthodes d'estimation du nombre de modes

Les 3 méthodes étudiées dans ce chapitre sont tout d'abord brièvement rappelées. Les méthodes algébriques sont décrites plus en détails dans le Chapitre 1 en Section 1.5.2, tandis que la méthode SRM est présentée dans le Chapitre 3.

Algorithm 4: ALG

Entrées: L'ensemble de données $\mathcal{D} = \{(\boldsymbol{x}_k, y_k)\}_{k=1}^{N} \subset \mathbb{R}^d \times \mathbb{R}$, un nombre maximal de modes $\overline{C}$ et un seuil $\overline{\sigma}$ sur les valeurs singulières pour le calcul du rang.

Pour $C = 1$ *à* $\overline{C}$ **Faire**

Calculer $M_C(d+1)$ et la matrice $\boldsymbol{L}_C$ comme en (1.35)

Si $\text{rank}(\boldsymbol{L}_C) < M_C(d+1)$ **Alors**

Sorties: Stopper l'algorithme et retourner le nombre de modes C

Sorties: Une erreur indiquant que $C > \overline{C}$

5.1.1 Méthode algébrique (ALG)

La méthode algébrique (ALG) classique estime le nombre de modes avec (1.49), rappelée ici :

$$\hat{C} = \min\{C \in \mathbb{N} : \text{rank}(\boldsymbol{L}_C) < M_C(d+1)\}, \tag{5.1}$$

où $\boldsymbol{L}_C$ est donnée en (1.35) et $M_C(d+1) = \binom{C+d}{d}$. La formulation (5.1) correspond à la méthode algébrique (ALG) d'origine proposée par Vidal et al. (2003). Ainsi, pour estimer C, on applique l'Algorithme 4.

L'algorithme met en lumière un paramètre important à régler qui est un seuil $\overline{\sigma}$ appliqué aux valeurs singulières pour estimer le rang par le nombre de valeurs singulières supérieures à $\overline{\sigma}$. La méthode algébrique est numériquement efficace car il suffit de calculer $\boldsymbol{L}_C$ et sa décomposition en valeurs singulières pour estimer le nombre de modes.

5.1.2 Méthode algébrique bruitée (ALG-N)

Algorithm 5: ALG-N

Entrées: L'ensemble de données $\mathcal{D} = \{(\boldsymbol{x}_k, y_k)\}_{k=1}^{N} \subset \mathbb{R}^d \times \mathbb{R}$, un nombre maximal de modes $\overline{C}$ et le seuil τ sur le bruit

Pour $C = 1$ *à* $\overline{C}$ **Faire**

Estimer la séquence de bruit $\boldsymbol{\eta}$ et la matrice $\tilde{\boldsymbol{L}}_C$ en résolvant (5.2)

Calculer $M_C(d+1)$

Si $\text{rank}(\tilde{\boldsymbol{L}}_C) < M_C(d+1)$ **Alors**

Sorties: Stopper l'algorithme et retourner le nombre de modes C

Sorties: Une erreur indiquant que $C > \overline{C}$

Dans l'extension au cas bruité de la méthode algébrique (ALG-N) proposée par Ozay et al. (2015), la base du problème à résoudre est la même que pour ALG, c'est-à-dire (5.1), en rempla-

çant la matrice non bruitée $\boldsymbol{L}_C$ par sa version bruitée $\tilde{\boldsymbol{L}}_C$. Cependant, cette version bruitée est plus complexe à obtenir car elle repose sur l'estimation de la séquence de bruits $\boldsymbol{\eta}$, pour laquelle il est également nécessaire de fixer un seuil τ :

$$\min_{\boldsymbol{\eta}\in\mathbb{R}^N} \operatorname{rank}(\tilde{\boldsymbol{L}}_C(\boldsymbol{\eta})), \quad \text{s.t. } |\eta_k| \leq \tau,\ k = 1, \ldots, N. \tag{5.2}$$

En pratique, le problème (5.2) est résolu de manière approximative en passant par une linéarisation de $\tilde{\boldsymbol{L}}_C(\boldsymbol{\eta})$ et une relaxation convexe de la minimisation du rang.

5.1.3 Méthode SRM

Dans le cas de la méthode SRM (Massucci et al., 2022) proposée au Chapitre 3, l'idée est de minimiser

$$\begin{aligned} J(C) &= \hat{L}_N(\overline{\boldsymbol{f}}) + \epsilon(\boldsymbol{f}, C), \\ \epsilon(\boldsymbol{f}, C) &= \frac{2p(2M)^{p-1}\alpha(C,q)\tilde{\Lambda}(\boldsymbol{f})\sqrt{\sum_{k=1}^{\mu} \|\boldsymbol{X}_{2a(k-1)+1}\|_2^2}}{\mu}, \end{aligned} \tag{5.3}$$

pour C allant de 1 à $\overline{C}$, comme décrit dans l'Algorithme 2.

5.2 Plan d'expériences

Dans cette section, on propose un plan d'expériences afin d'évaluer les avantages et inconvénients de chacune des méthodes présentées ci-dessus. Ici, l'objectif est de caractériser quelle méthode utiliser plutôt qu'une autre suivant le contexte d'étude. Certains paramètres expérimentaux principaux ayant une influence sur l'efficacité des méthodes ont été identifiés, tels que la taille de l'ensemble de données N, le nombre de modes C ou le type et le niveau de bruit.

Les expériences sont réalisées en étudiant un système à commutations arbitraires composé

de 3 sous-modèles linéaires d'ordre $n_a = n_b = 2$, dont les vecteurs de paramètres sont

$$\begin{aligned} \boldsymbol{\theta}_1 &= [-0.4 \quad 0.25 \quad -0.15 \quad 0.08]^T, \\ \boldsymbol{\theta}_2 &= [1.55 \quad -0.58 \quad -2.1 \quad 0.96]^T, \\ \boldsymbol{\theta}_3 &= [1 \quad -0.24 \quad -0.65 \quad 0.30]^T. \end{aligned} \tag{5.4}$$

Ce système a déjà été utilisé pour évaluer l'efficacité de méthodes d'identification dans Bako (2011). Il est utilisé ici pour générer un ensemble de données de N points avec (3.1). L'entrée u_k est un signal d'excitation de type gaussien de moyenne nulle et de variance $\sigma_u = 0.02$. Pour chaque expérience spécifique, une simulation de Monte Carlo avec 100 essais est considérée.

Les méthodes sont confrontées à des ensembles de données générés avec différents niveaux de bruit, correspondant à un rapport signal/bruit (SNR) de 30 dB, 20 dB ou 10 dB, et différents types de bruit : bruit blanc uniforme, bruit blanc gaussien et bruit gaussien coloré.

Nous utilisons les implémentations Matlab des méthodes mises à disposition par leurs auteurs [3] [4], fonctionnant sur un ordinateur portable avec un processeur i7 (4 cœurs) à 1,7 GHz et 16 Go de mémoire.

5.3 Observations préliminaires

Quelques expériences préliminaires ont conduit aux considérations suivantes concernant l'applicabilité des méthodes.

5.3.1 Méthode ALG

La méthode algébrique de Vidal et al. (2003) peut être appliquée à de petits comme à de grands ensembles de données, avec des temps de calcul de l'ordre de quelques dizaines de secondes pour des ensembles de données de quelques centaines de milliers de points. Cependant, le nombre de modes C influence le degré du polynôme (1.34), ce qui peut légèrement impacter le temps de calcul, mais aussi empêcher l'estimation des vecteurs paramètres, pour lesquels on doit, en pratique, trouver les racines d'un autre polynôme de degré similaire. Cependant, si l'on considère

3. Le code de la méthode ALG est disponible à http://vision.jhu.edu/code/
4. Le code de la méthode ALG-N est disponible à https://web.eecs.umich.edu/~necmiye/ols11_sub.html

TABLE 5.1 – Temps de calcul de la méthode ALG-N appliquée avec un nombre maximal de modes $\overline{C}$ à des ensembles de données de N points.

	$N = 100$	$N = 1000$	$N = 10\,000$
$\overline{C} = 2$	8 s	> 10 min	> 10 min
$\overline{C} = 3$	60 s	> 10 min	> 10 min
$\overline{C} = 4$	> 10 min	> 10 min	> 10 min

uniquement l'estimation du nombre de modes, l'Algorithme 4 nécessite simplement le calcul du rang de $\boldsymbol{L}_C$, ce qui reste faisable pour la plupart des valeurs de C. En ce qui concerne la limite inférieure du nombre de données, la seule condition est que $N > M_C(d+1)$, ce qui implique par exemple que pour $d = 4$ et $N = 100$, $C \leq 4$.

La méthode ALG n'a pas *a priori* d'hyperparamètre à spécifier. Cependant, en pratique, lorsqu'on traite des données bruitées, $\boldsymbol{L}_C$ est toujours de rang plein et il faut fixer un seuil $\overline{\sigma}$ sur les valeurs singulières considérées comme nulles pour calculer le rang numérique effectif. Dans nos expériences, $\overline{\sigma}$ a été réglé de telle sorte que la méthode donne un taux de réussite de 100% sur des données sans bruit.

5.3.2 Méthode ALG-N

Quelques expériences préliminaires rapportées dans le Tableau 5.1 montrent que la méthode algébrique étendue au cas bruité (ALG-N) a des difficultés à traiter de grands ensembles de données en raison de la charge de calcul élevée. En effet, pour résoudre (5.2), Ozay et al. (2015) propose de passer par une séquence de programmes semi-définis dont la taille et le nombre de variables augmentent avec le nombre de données N. En outre, le temps de calcul est également influencé par le nombre de modes C, qui régit directement le degré du polynôme (1.43) et affecte donc la complexité globale.

Le principal hyperparamètre de la méthode ALG-N est le seuil τ sur l'amplitude du bruit dans (5.2). Pour un bruit uniforme, nous utilisons une valeur basée l'amplitude de la vraie séquence de bruit, comme décrit dans le Tableau 5.2. Pour un bruit gaussien, nous utilisons plutôt une valeur basée sur l'écart-type σ_e du bruit. Comme pour la méthode ALG, le rang de $\tilde{\boldsymbol{L}}_C(\boldsymbol{\eta})$ utilisé dans l'Algorithme 4 pour la méthode ALG-N est calculé par rapport à un seuil $\overline{\sigma}$

dont la valeur influence les résultats. Ce seuil est réglé dans nos expériences de la même manière que pour la méthode ALG, avec les valeurs indiquées dans le Tableau 5.2.

5.3.3 Méthode SRM

La méthode SRM proposée au Chapitre 3 nécessite de grands ensembles de données pour fournir des résultats pertinents. En effet, elle n'est efficace que si le terme de complexité de la borne du Théorème 8, c'est-à-dire le second terme $\epsilon(\boldsymbol{f}, C)$ de (5.3), est du même ordre de grandeur que le risque empirique $\hat{L}_N(\overline{\boldsymbol{f}})$. En raison de la nature uniforme de la borne, un grand ensemble de données est nécessaire pour obtenir cette configuration. En ce qui concerne le temps de calcul induit par une grande quantité de données, des expériences préliminaires utilisant l'algorithme k-LinReg comme méthode générique d'identification des systèmes commutés dans l'Algorithme 2 montrent que la méthode SRM est plutôt efficace avec des temps de l'ordre de quelques dizaines de secondes pour des ensembles de données de quelques centaines de milliers de points.

5.3.4 Impact sur la configuration des expériences

Compte tenu de ces spécificités, des ensembles de données de différentes tailles sont considérés dans les expériences. On teste d'ailleurs la méthode ALG sur 80 000 points, noté ALG, et sur 100 points, noté ALG_{100}. De même, le nombre maximum de modes $\overline{C}$ utilisé par les algorithmes est fixé afin d'éviter des temps de calcul déraisonnables. Les paramètres précis et les valeurs des hyperparamètres pour toutes les méthodes sont donnés dans le Tableau 5.2.

TABLE 5.2 – Paramètres expérimentaux utilisés pour les différentes méthodes, notamment en ce qui concerne le nombre de données N et le nombre maximal de modes $\overline{C}$. La méthode ALG_{100} correspond à la méthode ALG avec un réglage spécifique pour $N = 100$.

Méthode	N	$\overline{C}$	Autres paramètres
SRM	80 000	6	$p = 1$, $a = 2$, $q = \infty$
ALG-N	100	3	$\tau = 1.1 \max_k \lvert e_k \rvert$ ou $\tau = 1.1\sigma_e$
			$\overline{\sigma} = 10^{-5}$ ou $\overline{\sigma} = 10^{-3}$
ALG	80 000	6	$\overline{\sigma} = 3.4 \times 10^{-3}$
ALG_{100}	100	4	$\overline{\sigma} = 7 \times 10^{-5}$

5.4 Évaluation des méthodes

TABLE 5.3 – Taux de réussite (en %) des méthodes dans différentes conditions de bruit.

Bruit blanc uniforme			
SNR	**30dB**	**20dB**	**10dB**
ALG	100	100	100
ALG_{100}	90	81	85
ALG-N $\overline{\sigma} = 10^{-5}$	84	44	3
ALG-N $\overline{\sigma} = 10^{-3}$	100	100	100
SRM	100	100	100

Bruit blanc gaussien			
SNR	**30dB**	**20dB**	**10dB**
ALG	100	96	0
ALG_{100}	84	84	72
ALG-N $\overline{\sigma} = 10^{-5}$	8	0	0
ALG-N $\overline{\sigma} = 10^{-3}$	100	100	100
SRM	100	100	92

Bruit blanc coloré			
SNR	**30dB**	**20dB**	**10dB**
ALG	100	87	0
ALG_{100}	88	80	60
ALG-N $\overline{\sigma} = 10^{-5}$	0	0	0
ALG-N $\overline{\sigma} = 10^{-3}$	100	100	100
SRM	100	100	91

Les résultats sont présentés dans le Tableau 5.3 pour les différents types et niveaux de bruit en termes de taux de réussite, c'est-à-dire le pourcentage d'essais pour lesquels le véritable nombre de modes $C = 3$ a été retrouvé.

5.4.1 Méthode ALG

Comme prévu, la méthode ALG estime bien le nombre de modes dans les situations très peu bruitées, proches du cas sans bruit pour lequel elle a été conçue. Cependant, l'estimation de C dépend fortement du seuil $\overline{\sigma}$ et le Tableau 5.3 montre que le réglage de sa valeur sur des données sans bruit générées aléatoirement peut également donner de bons résultats dans d'autres cas.

La Figure 5.1 illustre la sensibilité de la méthode par rapport à ce paramètre, dont la valeur optimale ainsi que la forme globale de son influence dépendent du nombre de données.

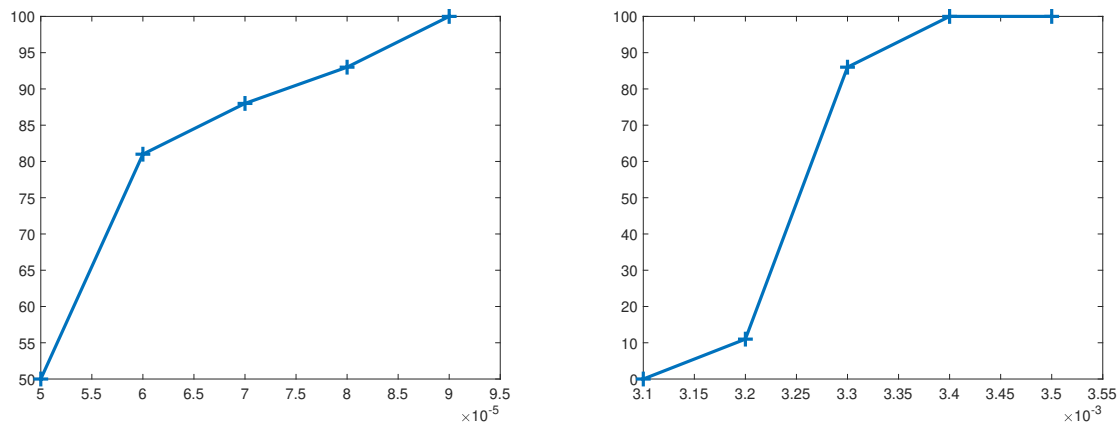

FIGURE 5.1 – Taux de réussite de la méthode ALG sur des données non bruitées par rapport à $\overline{\sigma}$ pour $N = 100$ (gauche) et $N = 80\,000$ (droite).

5.4.2 Méthode ALG-N

Comme pour la méthode ALG, la précision de ALG-N dépend fortement de la valeur du seuil $\overline{\sigma}$ et pourrait être largement améliorée par un réglage approprié de ce paramètre. En effet, en utilisant $\overline{\sigma} = 10^{-3}$, la méthode a retrouvé le vrai nombre de modes dans toutes les expériences. Cependant, la valeur correcte pour décider du rang dépend fortement des données elles-mêmes et pourrait être différente pour d'autres ensembles de données. Il est également à noter que l'utilisation de la fonction par défaut *rank* de Matlab, basée sur un seuil impliquant la taille de la matrice, a échoué dans tous les cas.

5.4.3 Méthode SRM

Le Tableau 5.3 montre que la méthode SRM récupère correctement le vrai nombre de modes avec un taux de réussite supérieur à 90% dans tous les cas, même pour des niveaux de bruit élevés (SNR = 10 dB). Bien qu'elle puisse facilement être appliquée à de grands ensembles de données, cette méthode n'a pas donné de résultats satisfaisants sur de petits ensembles de données.

5.4.4 Discussion des résultats

Tout d'abord, les résultats obtenus ci-dessus sont encourageants, car ils montrent qu'il y a au moins une des méthodes étudiées efficace pour la majorité des scénarios considérés. Cependant, ces expériences ont également mis en évidence un cas qui semble être en dehors du champ d'application de chacune des méthodes. Pour aider à la caractérisation de ce cas, ainsi que pour fournir un « guide du praticien » pour la détermination de la méthode la plus appropriée, nous proposons un arbre de décision décrit dans la Figure 5.2. D'une part, cet arbre fournit une approximation assez rudimentaire des capacités des méthodes, mais d'autre part, il devrait permettre d'identifier rapidement une méthode appropriée dans une situation donnée. La Figure 5.2 permet ainsi l'analyse suivante.

La méthode ALG montre des résultats satisfaisants pour les scénarios où l'on considère des données peu bruitées. Cette méthode permet donc également d'estimer des systèmes composés d'un grand nombre de modes, mais reste, de par sa définition, une méthode peu robuste au bruit. La méthode ALG-N, bien qu'efficace face au bruit, se montre rapidement bridée par la charge de calcul importante pour estimer le nombre de modes. Pour cette dernière, son utilisation lorsque le nombre de modes supposé est élevé n'est pas envisageable. Enfin, sous condition d'un nombre de données suffisamment élevé, la méthode SRM est efficace quel que soit le type ou le niveau de bruit dans les données. Cependant, son utilisation ne peut être considérée lorsque le nombre de points est trop faible. Ce manquement se fait resentir du côté droit de l'arbre de décision de la Figure 5.2, où aucune des 3 méthodes ne permet d'estimer le nombre de modes d'un système composé d'un nombre élevé de sous-modèles et dont les données sont fortement bruitées et en nombre limité. Pour combler ce manquement, considérer d'autres formes de régularisation pour d'autres valeurs de q pourrait permettre une analyse plus fine des données sans pour autant nécessiter autant de points que la version originelle.

5.5 Conclusions

Dans ce chapitre, on évalue l'efficacité de la méthode SRM par rapport aux méthodes ALG et ALG-N dans le but d'identifier laquelle est la plus adaptée pour estimer le nombre de modes d'un système hybride dans un contexte donné. Pour ce faire, ces trois méthodes sont soumises à

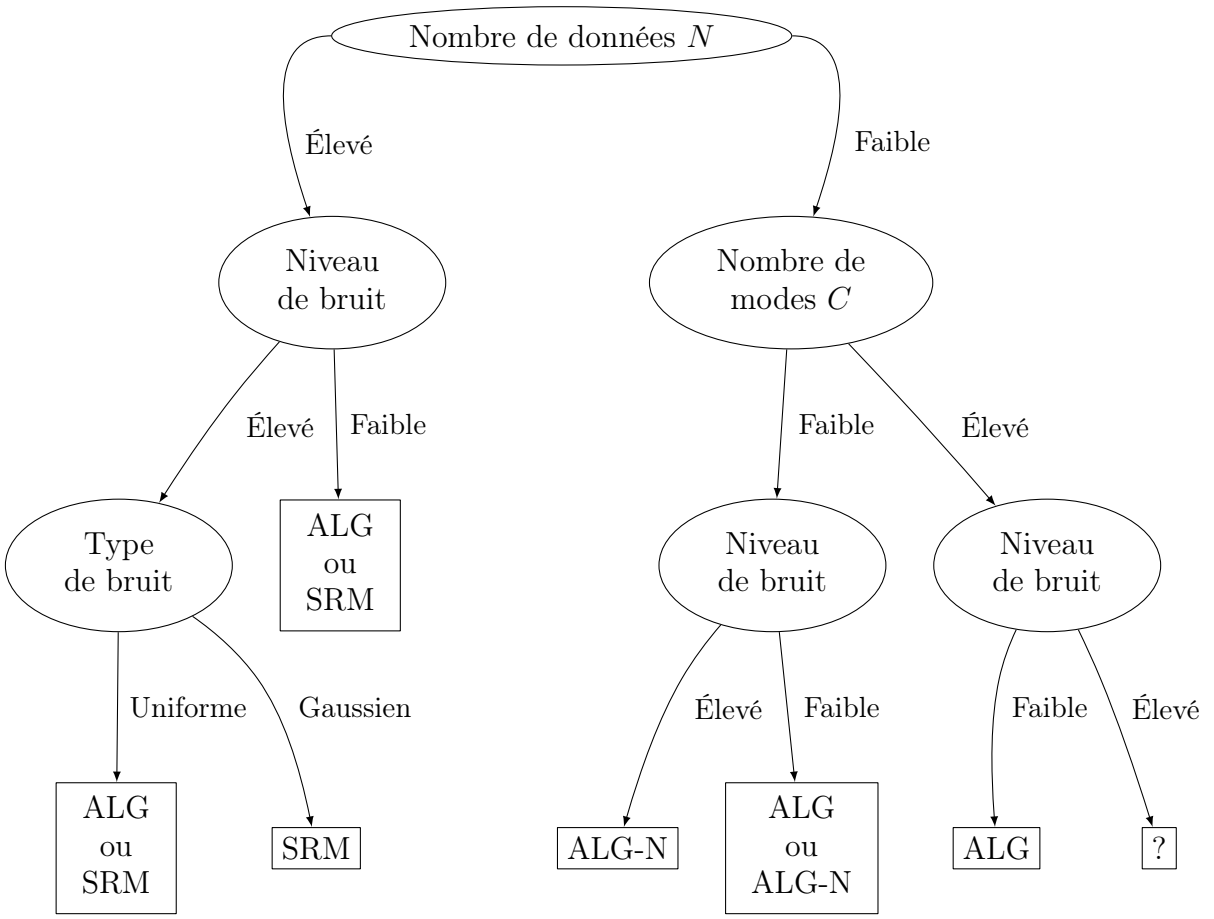

FIGURE 5.2 – Guide pour choisir une méthode appropriée pour estimer C.

différents scénarios où l'on fait varier le niveau et le type de bruit, le nombre de points, ainsi que le nombre de modes. Cette étude permet de mettre en lumière l'efficacité de la méthode SRM proposée au Chapitre 3 face à des données fortement bruitées. Cependant, la méthode n'est pour le moment efficace que si l'on dispose de grands jeux de données. Cette contrainte se fait d'ailleurs ressentir dans des contextes très bruités où les systèmes présentent un grand nombre de modes. En effet, dans un contexte où l'on dispose de peu de données mais très bruitées, il n'y a aucune des méthodes évaluées qui permette d'estimer un nombre de modes élevé.

Chapitre 6

Conclusions et Perspectives

Dans cette thèse, l'objectif est de mettre en lien les techniques d'identification des systèmes dynamiques, et celles issues de la théorie de l'apprentissage afin de proposer de nouvelles solutions pour l'identification des systèmes dynamiques hybrides.

Le Chapitre 1 introduit les fondamentaux de l'identification des systèmes et présente différentes méthodes et techniques d'identification des systèmes dynamiques hybrides. Ce chapitre met en évidence le problème de l'estimation du nombre de modes. En effet, il n'existe que très peu de méthodes permettant de calculer ce paramètre, et la plupart de ces méthodes sont soumises à des contraintes fortes telles que le type de bruit ou le nombre de modes maximal à estimer. C'est à ce problème que nous apportons une solution au Chapitre 3.

Dans le Chapitre 2, on présente les principaux théorèmes issus de la théorie statistique de l'apprentissage qui permettent d'obtenir des garanties non-asymptotiques sur l'apprentissage de modèles dans le cas statique et pour un nombre de données fini. Ces méthodes sont d'abord valides dans le cas où les données étudiées sont supposées i.i.d, mais de récents résultats, présentés dans la Section 2.5.1, surpassent cette contrainte. En effet, en faisant l'hypothèse que les données sont issues d'un processus β-mélangeant, il est possible d'obtenir des bornes sur l'erreur de généralisation qui soient valides pour l'apprentissage des systèmes dynamiques.

C'est ainsi que dans le Chapitre 3 le lien entre les deux domaines est proposé et mis en oeuvre. On propose une borne sur l'erreur de généralisation, dans le Théorème 8, valide pour l'identification des systèmes dynamiques hybrides. À partir de cette borne, une nouvelle méthode de sélection de modèles permettant d'estimer le nombre de modes C d'un système hybride est

développée, comme décrit dans l'Algorithme 2. La méthode de minimisation du risque structurel (méthode SRM) que nous proposons se révèle efficace face à tout type et niveau de bruit, et cela même lorsque le nombre de modes est élevé, là où d'autres méthodes, évaluées au Chapitre 5, ne parviennent pas à estimer le nombre de modes lorsqu'elles sont confrontées à certaines de ces contraintes. Cependant, les différentes expérimentations numériques ont permis d'observer que la méthode SRM nécessite un grand nombre de données pour trouver le bon nombre de modes.

C'est pour cette raison que nous avons également proposé différentes variantes de la régularisation considérée dans la méthode SRM, dans le but d'affiner nos résultats en limitant l'influence de C dans nos bornes, ce qui permet d'estimer plus facilement un grand nombre de modes. La borne sur l'erreur de généralisation est adaptée pour traiter différents scénarios de régularisation, et de nouveaux algorithmes d'optimisation permettant de prendre en compte la régularisation dans l'identification des systèmes hybrides sont proposés dans le Chapitre 4. Enfin, la Figure 5.2 du Chapitre 5 offre au lecteur une vue objective des avantages et inconvénients des méthodes que l'on y présente.

Un travail de thèse permet de traiter un sujet en profondeur tout en restant ouvert à de nouvelles idées d'études. Ainsi, plusieurs perspectives directes ou plus lointaines à ce travail peuvent être énoncées. La principale hypothèse sur laquelle reposent les bornes que nous proposons dans la thèse est que les données sont issues d'un processus β-mélangeant. Ainsi, une des pistes à privilégier pour renforcer notre théorie serait de s'assurer et de démontrer que les données issues d'un système hybride sont de cette nature. Un début de réponse se trouve dans les travaux de Vidyasagar (2013) et Doukhan (1994) où il est mentionné que cette hypothèse est valide et naturelle pour les processus stochastiques non-i.i.d. et les systèmes ARX linéaires.

Par ailleurs, nous ne proposons pas dans nos travaux de méthode pour estimer le coefficient de mélange $\beta(a)$ qui intervient dès lors que l'on applique l'hypothèse de mélange sur la nature des données. Nous avons pu nous soustraire à ce calcul pour la sélection de modèle car $\beta(a)$ apparaît dans un terme constant par rapport au nombre de modes C dans nos travaux. Cependant, l'estimation de ce paramètre permettrait de calculer complètement la borne, et ainsi d'obtenir des garanties sur l'apprentissage, ainsi que de comprendre à quel point $\beta(a)$ pourrait pénaliser la borne. Dans McDonald et al. (2015), une méthode d'estimation par histogrammes semble fournir de bons résultats pour estimer $\beta(a)$ et constituerait une première approche pour maîtriser ce

paramètre.

On peut également envisager d'autres méthodes n'employant pas la condition de mélange, comme dans Shalaeva et al. (2020) où des bornes bayesiennes non-i.i.d. sont proposées, ou encore dans Simchowitz et al. (2018), où l'utilisation des processus mélangeants est remis en question. Pour ce dernier, on pourrait dans un premier temps chercher à appliquer cette méthode pour des systèmes hybrides à faible dimension, pour lesquels on parvient à minimiser l'erreur empirique (Lauer, 2018).

Par ailleurs, un autre axe de perspective intéressant serait de poursuivre et d'étendre les travaux effectués au Chapitre 3 de ce manuscrit sur la régularisation. En effet, à la lumière des expérimentations présentées au Chapitre 5, on voit que la méthode proposée dans cette thèse (SRM) fonctionne bien lorsque le jeu de données à disposition est grand mais présente quelques difficultés dans un cas de nombre de données plus faible. Une étude plus fine permettrait de proposer des schémas de régularisation adaptés à ce problème. En outre, la majeure partie de cette thèse s'intéresse à l'identification de systèmes hybrides, mais focalise plus particulièrement sur l'estimation du nombre de modes. L'estimation des paramètres est très classique et se fonde sur des modèles ARX estimés par moindres carrés (ou par minimisation de l'erreur absolue). Une extension permettant de mieux gérer les erreurs du modèle et le bruit sur les données serait de s'intéresser à d'autres techniques d'estimation, comme par exemple, la technique de variable instrumentale (Soderstrom and Stoica, 1983; Young, 2011) qui a montré son efficacité depuis de nombreuses années.

Bibliographie

Bako, L. (2011). Identification of switched linear systems via sparse optimization. *Automatica*, 47(4) :668–677.

Bartlett, P. and Mendelson, S. (2002). Rademacher and Gaussian complexities : Risk bounds and structural results. *Journal of Machine Learning Research*, 3 :463–482.

Bemporad, A., Garulli, A., Paoletti, S., and Vicino, A. (2005). A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10) :1567–1580.

Bradley, R. (2005). Basic properties of strong mixing conditions. A survey and some open questions. *Probability Surveys*, 2 :107–144.

Doukhan, P. (1994). *Mixing : Properties and Examples*. Springer.

Fazel, M., Hindi, H., and Boyd, S. (2003). Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proc. of the American Control Conference, Denver, CO, USA*.

Janson, S. (2004). Large deviations for sums of partly dependent random variables. *Random Structures Algorithms*, 24 :234–248.

Lampert, C. H., Ralaivola, L., and Zimin, A. (2018). Dependency-dependent bounds for sums of dependent random variables.

Lasserre, J. (2001). Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3) :796–817.

Lauer, F. (2013). Estimating the probability of success of a simple algorithm for switched linear regression. *Nonlinear Analysis : Hybrid Systems*, 8 :31–47.

Lauer, F. (2016). On the complexity of switching linear regression. *Automatica*, 74 :80–83.

Lauer, F. (2018). Global optimization for low-dimensional switching linear regression and bounded-error estimation. *Automatica*, 89 :73–82.

Lauer, F. (2020a). Error bounds for piecewise smooth and switching regression. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4) :1183–1195.

Lauer, F. (2020b). Risk bounds for learning multiple components with permutation-invariant losses. In *Proc. of the 23th Int. Conf. on Artificial Intelligence and Statistics (AISTATS), Virtual Meeting*.

Lauer, F. and Bloch, G. (2019). *Hybrid System Identification : Theory and Algorithms for Learning Switching Models*. Springer.

Lauer, F., Bloch, G., and Vidal, R. (2011). A continuous optimization framework for hybrid system identification. *Automatica*, 47(3) :608–613.

Ledoux, M. and Talagrand, M. (1991). *Probability in Banach Spaces : Isoperimetry and Processes*. Springer.

Liberzon, D. (2003). *Switching in Systems and Control*, volume 190. Springer.

Ljung, L. (1999). *System Identification : Theory for the User*. Prentice Hall PTR, Upper Saddle River, NJ, USA, second edition.

Ma, Y. and Vidal, R. (2005). Identification of deterministic switched arx systems via identification of algebraic varieties. In *International Workshop on Hybrid Systems : Computation and Control*, pages 449–465. Springer.

Massucci, L., Lauer, F., and Gilson, M. (2022). A statistical learning perspective on switched linear system identification. *Automatica*, 145 :110532.

McDonald, D. J., Shalizi, C. R., and Schervish, M. (2015). Estimating beta-mixing coefficients via histograms. *Electronic Journal of Statistics*, 9(2) :2855–2883.

Mohri, M. and Rostamizadeh, A. (2009). Rademacher complexity bounds for non-i.i.d. processes. In *Advances in Neural Information Processing Systems 21*, pages 1097–1104.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of Machine Learning*. The MIT Press, second edition.

MOSEK ApS (2019). *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*

Ohlsson, H. and Ljung, L. (2013). Identification of switched linear regression models using sum-of-norms regularization. *Automatica*, 49(4) :1045–1050.

Ozay, N., Lagoa, C., and Sznaier, M. (2015). Set membership identification of switched linear systems with known number of subsystems. *Automatica*, 51 :180–191.

Ozay, N., Sznaier, M., Lagoa, C., and Camps, O. (2012). A sparsification approach to set membership identification of switched affine systems. *IEEE Transactions on Automatic Control*, 57(3) :634–648.

Ralaivola, L. and Amini, M.-R. (2015). Entropy-based concentration inequalities for dependent variables. In *International Conference on Machine Learning*, pages 2436–2444.

Schoukens, J. and Pintelon, R. (2014). *Identification of Linear Systems : a Practical Guideline to Accurate Modeling*. Elsevier.

Shalaeva, V., Esfahani, A. F., Germain, P., and Petreczky, M. (2020). Improved PAC-Bayesian bounds for linear regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5660–5667.

Simchowitz, M., Mania, H., Tu, S., Jordan, M., and Recht, B. (2018). Learning without mixing : Towards a sharp analysis of linear system identification. In *Proceedings of the Conference on Computational Learning Theory (COLT), Stockholm, Sweden.*

Soderstrom, T. and Stoica, P. G. (1983). *Instrumental Variable Methods for System Identification*. Springer.

Usunier, N., Amini, M. R., and Gallinari, P. (2005). Generalization error bounds for classifiers trained with interdependent data. *Advances in Neural Information Processing Systems*, 18.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vidal, R., Soatto, S., Ma, Y., and Sastry, S. (2003). An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proc. of the 42nd IEEE Conference on Decision and Control (CDC), Maui, HI, USA*, pages 167–172.

Vidyasagar, M. (2013). *Learning and Generalisation : with Applications to Neural Networks*. Springer.

Vidyasagar, M. and Karandikar, R. L. (2008). A learning theory approach to system identification and stochastic adaptive control. *Journal of Process Control*, 18(3) :421 – 430.

Young, P. C. (2011). *Recursive estimation and time-series analysis : An introduction for the student and practitioner*. Springer.

Yu, B. (1994). Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, 22(1) :94–116.

Annexe A

Résultats issus de la littérature

Théorème 9 (Inégalité de Hoeffding) *Pour N variables aléatoires X_k, $k = 1, \dots, N$, qui sont des copies indépendantes d'une variable aléatoire bornée $X \in [a, b]$, on a, pour tout $\epsilon > 0$,*

$$P^N \left[\frac{1}{N} \sum_{k=1}^{N} X_k - \mathbb{E}(X) \geq \epsilon \right] \leq \exp\left(\frac{-2N\epsilon^2}{(b-a)^2} \right) \tag{A.1}$$

$$P^N \left[\frac{1}{N} \sum_{k=1}^{N} X_k - \mathbb{E}(X) \leq -\epsilon \right] \leq \exp\left(\frac{-2N\epsilon^2}{(b-a)^2} \right) \tag{A.2}$$

$$P^N \left[\left| \frac{1}{N} \sum_{k=1}^{N} X_k - \mathbb{E}(X) \right| \leq -\epsilon \right] \leq 2\exp\left(\frac{-2N\epsilon^2}{(b-a)^2} \right), \tag{A.3}$$

où P^N décrit la distribution jointe des X_k, $k = 1, \dots, N$.

Théorème 10 (Principe de contraction (Ledoux and Talagrand, 1991)) *Si une fonction $\phi : \mathbb{R} \to \mathbb{R}$ est L_ϕ-Lipshitz, i.e., $\forall (u, v) \in \mathbb{R}^2,\ |\phi(u) - \phi(v)| \leq L_\phi |u - v|$, alors,*

$$\hat{\mathcal{R}}_{\mathbf{Z}_N} \left(\{\phi \circ f : f \in \mathcal{F}\} \right) \leq L_\phi \hat{\mathcal{R}}_{\mathbf{Z}_N}(\mathcal{F}).$$

La complexité de Rademacher de modèles linéaires peut être bornée grâce au théorème suivant.

Théorème 11 (Bartlett and Mendelson (2002)) *Soit $\mathcal{F}$ une classe de fonctions linéaires $\mathcal{F} = \{f : f(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x},\ \|\boldsymbol{\theta}\|_2 \leq \Lambda\}$. Alors,*

$$\hat{\mathcal{R}}_{\mathbf{X}_N}(\mathcal{F}) \leq \frac{\Lambda}{N} \sqrt{\sum_{k=1}^{N} \|\mathbf{X}_k\|_2^2}.$$

Théorème 12 (Lemme 2 de Lauer (2020b)) *Soit $\mathcal{F}$ définie comme en (3.6), et $q \in (0, \infty]$. Alors, $\tilde{\mathcal{F}}$ définie en (3.12)–(3.14) satisfait*

$$\tilde{\mathcal{F}} \subseteq \Pi_q = \prod_{j=1}^{C} \left\{ f_j \in \mathbb{R}^{\mathcal{X}} : \ f_j(\boldsymbol{x}) = \boldsymbol{\theta}_j^T \boldsymbol{x}, \|\boldsymbol{\theta}_j\|_2 \leq \Lambda j^{-\frac{1}{q}} \right\}. \tag{A.4}$$

Preuve du Théorème 12 : Si $q = \infty$, alors $\mathcal{F}$ et $\tilde{\mathcal{F}}$ peuvent se réécrire comme un produit de classe de fonctions composantes indépendantes :

$$\mathcal{F} = \left\{ \boldsymbol{f} \in \mathbb{R}^{\mathcal{X}} : \max_{j \in [C]} \|\theta_j\|_2 \leq \Lambda \right\} = \left\{ \boldsymbol{f} \in \mathbb{R}^{\mathcal{X}} : \max_{j \in [C]} \|\boldsymbol{\theta}_{\pi(j)}\|_2 \leq \Lambda \right\} = \tilde{\mathcal{F}} \tag{A.5}$$

$$= \prod_{j=1}^{C} \left\{ f_j \in \mathbb{R}^{\mathcal{X}} : \|\boldsymbol{\theta}_{\pi(j)}\|_2 \leq \Lambda \right\}. \tag{A.6}$$

Si $q < \infty$, l'invariance aux permutations de la norme ℓ_q implique que pour tout $\boldsymbol{f} \in \mathcal{F}$,

$$\|\Omega(\tilde{\boldsymbol{f}})\|_q^q = \|\Omega(\boldsymbol{f})\|_q^q \leq \Lambda^q, \tag{A.7}$$

tandis que, pour $j \in [C]$,

$$\|\Omega(\tilde{\boldsymbol{f}})\|_q^q = \sum_{i=1}^{C} \|\tilde{\boldsymbol{\theta}}_i\|_2^q \geq \sum_{i=1}^{j} \|\tilde{\boldsymbol{\theta}}_i\|_2^q \geq j \|\tilde{\boldsymbol{\theta}}_j\|_2^q, \tag{A.8}$$

où la dernière inégalité est obtenue grâce à la structure de $\tilde{\mathcal{F}}$, imposée par (3.13). Finalement, pour tout $\tilde{\boldsymbol{f}} \in \tilde{\mathcal{F}}$ et tout $j \in [C]$,

$$\|\tilde{\boldsymbol{\theta}}_j\|_2^q \leq \frac{\Lambda^q}{j}, \tag{A.9}$$

ce qui complète la preuve. □

Annexe B

Caractérisation de la dépendance à C pour le Théorème 7

On démontre ici, comme dans Lauer (2020b), que pour tout entier $C \geq 2$ et $q \in (0, \infty]$, avec $\alpha(C, q)$ défini comme en (3.10),

$$\sum_{k=1}^{C} k^{-\frac{1}{q}} \leq \alpha(C, q). \tag{B.1}$$

Pour $q = \infty$, on voit simplement que $\sum_{k=1}^{C} k^{-\frac{1}{q}} = \sum_{k=1}^{C} 1 = C$.

Pour $q < \infty$, on écrit

$$\sum_{k=1}^{C} k^{-\frac{1}{q}} = 1 + \sum_{k=2}^{C} k^{-\frac{1}{q}} \leq 1 + \int_{1}^{C} x^{-\frac{1}{q}} dx. \tag{B.2}$$

Ainsi, pour $q = 1$ on a

$$\sum_{k=1}^{C} k^{-\frac{1}{q}} \leq 1 + \int_{1}^{C} \frac{1}{x} dx = 1 + \log C - \log 1 = 1 + \log C, \tag{B.3}$$

tandis que pour $q \neq 1$, on a

$$\sum_{k=1}^{C} k^{-\frac{1}{q}} \leq 1 + \frac{q}{q-1}(C^{(q-1)/q} - 1) = \frac{qC^{(1-q)/q} - 1}{q-1}. \tag{B.4}$$

Ainsi, pour $q > 1$ on a

$$\sum_{k=1}^{C} k^{-\frac{1}{q}} < \frac{qC^{(1-q)/q}}{q-1}, \tag{B.5}$$

tandis que pour $q < 1$

$$\sum_{k=1}^{C} k^{-\frac{1}{q}} \leq \frac{1 - qC^{(1-q)/q}}{1-q} \leq \frac{1}{1-q}. \tag{B.6}$$

Résumé

Cette thèse porte sur l'application de la théorie statistique de l'apprentissage pour l'identification de systèmes dynamiques hybrides. La théorie de l'apprentissage permet d'obtenir des garanties statistiques sur la précision de modèles et ce pour un nombre de données fini. Ici, en étendant son cadre d'application à celui des systèmes dynamiques, on propose de nouvelles solutions pour l'identification des systèmes hybrides. En effet une nouvelle borne sur l'erreur de généralisation valide pour l'identification des systèmes à commutation arbitraire est obtenue, et son utilisation donne lieu à une nouvelle méthode de sélection de modèle pour l'estimation du nombre de sous-modèles des systèmes hybrides. La borne est adaptée pour différents scénarios de régularisation, et de nouveaux algorithmes d'optimisation prenant en compte ces différents scénarios sont proposés. Cette nouvelle méthode d'estimation du nombre de modes est confrontée à d'autres méthodes existantes, et les avantages et inconvénients de chacune de ces méthodes sont étudiés.

Abstract

This thesis deals with the application of statistical learning theory to the identification of hybrid dynamical systems. Learning theory allows one to obtain statistical guarantees on the accuracy of models for a finite number of data. Here, by extending the framework of this theory to that of dynamic systems, we propose new solutions for the identification of switched systems. Indeed, a new generalization error bound valid for the identification of switched systems is obtained, and its use gives rise to a new model selection method for estimating the number of submodels of hybrid systems. The bound is adapted for different regularization scenarios, and new optimization algorithms taking into account these different scenarios are proposed. This new method for estimating the number of modes is compared with existing ones, and the advantages and disadvantages of each of these methods are studied.