

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
- T L E M C E N -



المدرسة العليا في العلوم التطبيقية
École Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي
المدرسة العليا في العلوم التطبيقية
- تلمسان -

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique
Spécialité : Automatique

Présenté par : Nidal Omar KOTBI et Boumediene BENZELLAT

Thème

**Identification des systèmes non linéaires
par l'opérateur de Koopman**

Soutenu publiquement, le 02/07/ 2023 , devant le jury composé de :

M.Meghebbar Abdelmadjid	MCA	ESSA. Tlemcen	Président
M.Cherki Brahim	Professeur	ESSA. Tlemcen	Encadreur
M.Arichi Fayçal	MCA	ESSA. Tlemcen	Examineur 1
M.Mokhtari Mohammed Rida	MCA	ESSA. Tlemcen	Examineur 2

Année universitaire : 2022 / 2023

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

REMERCIEMENTS

Nous tenons à exprimer nos sincères remerciements à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire.

*Tout d'abord, nous souhaitons exprimer notre profonde gratitude à notre encadreur, Monsieur **Cherki Brahim**, pour sa précieuse guidance, son expertise et son soutien tout au long de ce travail. Ses conseils éclairés, sa disponibilité et ses encouragements ont été d'une importance capitale pour la réussite de ce projet. Nous remercions également Messieurs **A.Meghebbar**, **F.Arichi** et **M.R.Mokhtari**, pour avoir accepté d'examiner notre travail.*

Nous tenons également à remercier du fond du cœur nos parents et notre famille pour leur soutien constant, leur amour inconditionnel et leur présence tout au long de ce parcours. Leur soutien moral, leurs encouragements et leurs sacrifices ont été une source d'inspiration et de motivation inestimable pour nous. Nous leur sommes profondément reconnaissants pour tout ce qu'ils ont fait pour nous.

Nous souhaitons également exprimer notre reconnaissance envers nos collègues et amis qui ont partagé cette aventure avec nous. Leurs échanges, leur collaboration et leur amitié ont enrichi cette expérience et ont rendu ce travail plus stimulant. Leurs encouragements et leur support ont été précieux.

Enfin, nous voudrions adresser nos remerciements à toutes les personnes qui ont apporté leur contribution, même minime, à la réalisation de ce mémoire. Vos conseils, vos commentaires et votre soutien ont été d'une grande valeur et ont contribué à améliorer ce travail.

Nous sommes reconnaissants envers chacun d'entre vous pour votre précieuse contribution et votre présence à nos côtés tout au long de ce parcours. Vos encouragements et votre soutien ont été essentiels et nous sommes profondément reconnaissants de vous avoir à nos côtés.

Merci sincèrement à tous.

Nidal Omar KOTBI et Boumediene BENZELLAT.

المخلص

يركز مشروع العام الأخير هذا على خوارزمية التعرف المتناثر على الديناميكيات غير الخطية (SINDy) ، والتي تسمح باستنتاج المعادلات الأساسية لنظام ديناميكي غير خطي من البيانات التجريبية. نستكشف الأسس النظرية لـ SINDy ، مع التأكيد على مفاهيم التشتت والتسوية، وكذلك الأدوات الرياضية المستخدمة لحل مشكلة التعريف. نطبق هذا النهج على أمثلة للأنظمة الديناميكية الخطية وغير الخطية باستخدام عمليات المحاكاة التي يتم إجراؤها باستخدام MATLAB ، بهدف تصميم هذه الأنظمة وتحديدتها. نقوم أيضًا بفحص قدرة SINDy على تحديد الأنظمة غير الخطية في وجود الضوضاء. النتائج التي تم الحصول عليها واعدة، وتناقش الاستنتاجات المهمة.

الكلمات الرئيسية : سيندي (تحديد الديناميكا غير الخطية المتناثرة)، نظام ديناميكي غير خطي، بيانات تجريبية، قلة الكثافة، التنظيم، الأدوات الرياضية، مشكلة التعرف، أنظمة ديناميكية خطية وغير خطية، محاكاة، نمذجة، ضيغ.

ABSTRACT

This final year project focuses on the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm, which allows deducing the underlying equations of a nonlinear dynamical system from experimental data. We explore the theoretical foundations of SINDy, emphasizing the concepts of sparsity and regularization, as well as the mathematical tools used to solve the identification problem.

We apply this approach to examples of linear and nonlinear dynamical systems using simulations conducted with MATLAB, aiming to model and identify these systems. We also examine the ability of SINDy to identify nonlinear systems in the presence of noise. The obtained results are promising, and important conclusions are discussed.

Keywords : SINDy (Sparse Identification of Nonlinear Dynamics), nonlinear dynamical system, experimental data, sparsity, regularization, mathematical tools, identification problem, linear and nonlinear dynamical systems, simulations, modeling, noise.

RÉSUMÉ

Ce projet de fin d'études se concentre sur l'algorithme de SINDy (Sparse Identification of Nonlinear Dynamics), qui permet de déduire les équations sous-jacentes d'un système dynamique non linéaire à partir de données expérimentales. Nous explorons les fondements théoriques de SINDy, en mettant l'accent sur les concepts de sparsité et de régularisation, ainsi que les outils mathématiques utilisés pour résoudre le problème d'identification.

Nous appliquons cette approche à des exemples de systèmes dynamiques linéaires et non linéaires en utilisant des simulations réalisées avec MATLAB, dans le but de modéliser et d'identifier ces systèmes. Nous examinons également la capacité de SINDy à identifier les systèmes non linéaires en présence de bruit. Les résultats obtenus sont encourageants, et des conclusions importantes sont discutées.

Mots-clés : SINDy (Sparse Identification of Nonlinear Dynamics), système dynamique non linéaire, données expérimentales, sparsité, régularisation, outils mathématiques, problème d'identification, systèmes dynamiques linéaires et non linéaires, simulations, modélisation, bruit.

TABLE DES MATIÈRES

Table des Figures	9
Liste d'abréviations	10
Introduction générale	11
1 Généralités sur SINDy	12
1.1 Introduction :	12
1.2 Identification parcimonieuse des Dynamiques Non Linéaires "SINDy" :	13
1.3 Contexte historique de SINDy :	13
1.4 Applications et Extensions de SINDy : [11]	14
1.5 Les limites de SINDy :	15
1.6 Conclusion :	16
2 L'opérateur de Koopman et la décomposition en mode dynamique (DMD)	17
2.1 Introduction :	17
2.2 L'opérateur de Koopman : [11, 49]	18
2.2.1 Introduction à l'opérateur de Koopman :	18
2.2.2 Les mathématiques derrière l'opérateur Koopman :	18
2.3 La décomposition en mode dynamique (DMD) : [11, 49, 40]	20
2.3.1 Introduction à la DMD :	20
2.3.2 L'algorithme DMD :	21
2.3.3 La décomposition spectrale et le développement DMD :	24
2.3.4 La décomposition en mode dynamique avec contrôle (DMDc) : [11, 40]	26
2.3.5 Décomposition en Mode Dynamique étendu (<i>Extended DMD, EDMD</i>) :	29
2.4 Conclusion :	31
3 Identification parcimonieuse de la dynamique non linéaire 'SINDy'	32
3.1 Introduction :	32
3.2 Contexte : [12]	33
3.2.1 Régression symbolique et apprentissage automatique : [12]	33
3.2.2 Représentation parcimonieuse et acquisition comprimée : [12]	33
3.3 Identification parcimonieuse de la dynamique non linéaire 'SINDy' : [11]	34

3.3.1	Exemple de compréhension : Équations de Lorenz	36
3.3.2	Algorithme pour la représentation parcimonieuse de la dynamique avec du bruit :	37
3.4	Conclusion :	38
4	Application de la méthode de Identification parcimonieuse de la dynamique non linéaire 'SINDy'	39
4.1	Introduction :	39
4.2	Exemple 1 : Système linéaire	40
4.2.1	Résultat de simulation de système original (Prenant une condition initiale $x_0 = [2; 0; 1]$) :	41
4.2.2	Résultat de l'identification :	42
4.2.3	Interprétation :	43
4.3	Exemple 2 : Équation de Duffing	43
4.3.1	Résultat de simulation de système original : (Prenant une condition initiale $x_0 = [2; -2]$)	44
4.3.2	Résultat de l'identification :	45
4.3.3	Interprétation :	46
4.4	Exemple 3 : Pendule simple	47
4.4.1	Description du système :	47
4.4.2	La représentation d'état :	47
4.4.3	Résultats de simulation de système original (Prenant une condition initiale $x_0 = [2; 2]$) :	48
4.4.4	Résultat de l'identification :	49
4.4.5	L'évolution du modèle dynamique résultant :	50
4.4.6	Erreur de reconstruction des états par SINDy :	50
4.4.7	Interprétation :	51
4.5	Exemple 4 : Équation de Van der pol (oscillateur non forcé)	51
4.5.1	Résultats de simulation de système original (Prenant $\mu = 1$ et une condition initiale $x_0 = [2; 3]$) :	52
4.5.2	Résultat de l'identification :	53
4.5.3	L'évolution du modèle dynamique résultant :	53
4.5.4	Erreur de reconstruction des états par SINDy :	54
4.5.5	Interprétation des résultats :	54
4.5.6	Élimination du bruit :	54
4.6	Conclusion :	55
	Conclusion générale	56

TABLE DES FIGURES

1.1	Présentation schématique de l'identification de modèles non linéaires à partir de données de grande dimension à l'aide de l'identification parcimonieuse de la dynamique non linéaire (SINDy) [12]. Cette procédure est modulaire, de sorte que différentes techniques peuvent être utilisées pour les étapes d'extraction de caractéristiques et de régression. Dans cet exemple d'écoulement devant un cylindre, SINDy découvre le modèle de Noack et al. [37]. Modifié de Brunton et al. [12].	14
2.1	Schéma illustrant l'opérateur de Koopman pour les systèmes dynamiques non linéaires, d'après [8]. Les lignes en pointillés de $y_k \rightarrow x_k$ indiquent que nous aimerions pouvoir récupérer l'état d'origine	20
2.2	Vue d'ensemble de la DMD illustrée sur l'écoulement de fluide autour d'un cylindre circulaire à un nombre de Reynolds de 100. Reproduite à partir de [30].	24
3.1	Moindres carrés seuillés séquentiellement, de [12].	35
3.2	Schéma de l'algorithme d'identification parcimonieuse de la dynamique non linéaire, démontré sur les équations de Lorenz. Reproduite de [12].	36
4.1	Le portrait de phase du système étudié	41
4.2	L'évolution temporelle du système étudié X_{data}	41
4.3	Comparaison des données reconstruites par la methode SINDy X_{sindy} (en vert) avec les données du système étudié X_{data} (en bleu, marron et orange).	42
4.4	Erreur de reconstruction des états par SINDy : Différence entre les données reconstruites et les données originales.	43
4.5	Le portrait de phase du système étudié	44
4.6	L'évolution temporelle du système étudié X_{data}	44
4.7	Comparaison des données reconstruites par la sindy X_{sindy} (en vert) avec les données du système étudié X_{data} (en bleu et marron).	45
4.8	Erreur de reconstruction des états par SINDy : Différence entre les données reconstruites et les données originales.	46
4.9	Le portrait de phase du système étudié.	48
4.10	L'évolution temporelle du système étudié Xdata.	49
4.11	Comparaison des données reconstruites par la sindy X_{sindy} (en vert) avec les données du système étudié X_{data} (en bleu et marron).	50

4.12 Erreur de reconstruction des états par SINDy : Différence entre les données reconstruites et les données originales.	50
4.13 Le portrait de phase du système original de Van Der Pol.	52
4.14 L'évolution du système original de Van Der Pol	52
4.15 Comparaison des données reconstruites par la sindy X_{sindy} (en vert) avec les données du système étudié X_{data} (en bleu et marron).	53
4.16 Erreur de reconstruction des états par SINDy avec bruit : Différence entre les données reconstruites et les données originales.	54
4.17 Erreur de reconstruction des états par SINDy sans bruit : Différence entre les données reconstruites et les données originales non bruitées.	55

LISTE D'ABRÉVIATIONS

Acronyme	Signification
SINDy	Sparse Identification of Nonlinear Dynamics
LASSO	Least Absolute Shrinkage and Selection Operator
POD	Proper Orthogonal Decomposition
STLS	Sequentially Trimmed Least Squares
NARMAX	Nonlinear AutoRegressive Moving Average with eXogenous inputs
DMD	Dynamic Mode Decomposition
DMDc	Dynamic Mode Decomposition with control
EDMD	Extended Dynamic Mode Decomposition
SVD	Singular Value Decomposition
SRC	Sparse Representation Classification

INTRODUCTION GÉNÉRALE

Les systèmes dynamiques non linéaires constituent un domaine d'étude complexe mais crucial dans de nombreux domaines scientifiques et techniques. Comprendre et modéliser ces systèmes est essentiel pour prédire leur comportement, concevoir des contrôles efficaces et prendre des décisions éclairées. Cependant, l'analyse et la modélisation des systèmes dynamiques non linéaires peuvent souvent être difficiles en raison de leur complexité intrinsèque et du manque de connaissances sur les équations sous-jacentes régissant leur comportement.

Dans ce contexte, notre travail de mémoire se concentre sur l'algorithme de SINDy (Sparse Identification of Nonlinear Dynamics), une méthode efficace permettant de déterminer les équations sous-jacentes d'un système dynamique non linéaire complexe à partir de données expérimentales. SINDy offre une approche novatrice et prometteuse pour analyser les systèmes dynamiques basée sur des données, en exploitant la régression symbolique et la représentation parcimonieuse.

Notre mémoire vise à présenter en détail l'algorithme de SINDy, en mettant l'accent sur les principes mathématiques qui sous-tendent cette méthode. Nous explorons également les concepts clés de la sparsité et de la régularisation, ainsi que leur application pour résoudre le problème d'identification dans SINDy. Grâce à la combinaison de la régression symbolique et de la représentation parcimonieuse, SINDy permet de découvrir de manière efficace les équations non linéaires décrivant les relations entre les variables mesurées et les dynamiques observées d'un système.

Pour situer SINDy dans le contexte de l'analyse des systèmes dynamiques, nous aborderons également d'autres méthodes utilisées pour étudier ces systèmes à partir de données, telles que l'opérateur de Koopman et l'algorithme DMD (Dynamic Mode Decomposition). Ces méthodes sont complémentaires à SINDy et offrent des approches différentes pour analyser les données, en combinant l'identification des équations, l'analyse des modes et la prédiction du comportement futur.

Nous explorerons plus en détail l'opérateur de Koopman et l'algorithme DMD, ainsi que leurs relations avec l'approximation de l'opérateur de Koopman à partir des données de l'espace d'état. Comprendre ces concepts et leurs implications dans l'analyse des systèmes dynamiques nous permettra d'appréhender plus précisément le rôle de SINDy et de sa méthode d'identification parcimonieuse dans cette discipline.

En résumé, notre mémoire se consacre à une analyse approfondie de l'algorithme de SINDy et à sa mise en application dans l'analyse des systèmes dynamiques non linéaires. Nous examinons également le contexte dans lequel cet algorithme s'inscrit, c'est-à-dire l'analyse des systèmes dynamiques à partir de données. À travers l'exploration des concepts essentiels tels que la sparsité, la régularisation et les outils mathématiques utilisés dans SINDy, nous mettons en avant son efficacité.

CHAPITRE 1

GÉNÉRALITÉS SUR SINDY

1.1 Introduction :

La méthode SINDy (Sparse Identification of Nonlinear Dynamics) est une approche révolutionnaire pour identifier les équations sous-jacentes aux systèmes dynamiques non linéaires à partir de données observées, sans connaître a priori leur forme exacte. Elle utilise une régression parcimonieuse pour identifier les termes les plus importants dans une équation candidate, permettant ainsi d'obtenir une représentation concise et interprétable de la dynamique du système. Depuis son introduction en 2016 par Brunton, Proctor et Kutz [12], SINDy a eu un impact significatif sur le domaine de la modélisation des systèmes dynamiques. Cette méthode a été appliquée avec succès dans divers domaines scientifiques et présente des extensions prometteuses, telles que l'identification de modèles d'équations aux dérivées partielles. Ce chapitre offre une vue d'ensemble de SINDy, une revue historique, ainsi que ses applications et extensions. Il aborde également les limites auxquelles cette méthode peut faire face.

1.2 Identification parcimonieuse des Dynamiques Non Linéaires "SINDy" :

SINDy (Sparse Identification of Nonlinear Dynamics) est une méthode utilisée pour découvrir les systèmes dynamiques non linéaires à partir de données. Elle se base sur le principe de parcimonie, qui suggère que les équations régissant de nombreux systèmes dynamiques peuvent être représentées de manière précise en utilisant seulement quelques termes importants.

L'algorithme SINDy utilise des techniques de régression parcimonieuse pour identifier les termes significatifs dans une équation candidate qui expliquent au mieux la dynamique observée. Plus précisément, il utilise des méthodes de régression L1 (également connue sous le nom de régulation LASSO, "Least Absolute Shrinkage and Selection Operator") pour estimer les coefficients associés à chaque terme de l'équation. En formulant le problème comme une tâche de régression parcimonieuse, SINDy peut automatiquement sélectionner les termes les plus pertinents, permettant ainsi d'obtenir une représentation concise du comportement du système.

Une caractéristique clé de SINDy est sa capacité à découvrir des équations non linéaires à partir de données, sans avoir besoin de connaître a priori la forme fonctionnelle exacte des équations. Cela la distingue des méthodes traditionnelles d'identification de systèmes qui se concentrent principalement sur des modèles linéaires ou qui supposent une connaissance préalable de la structure du système.

SINDy a été largement utilisé dans divers domaines de recherche, tels que la physique, la biologie, la chimie et l'ingénierie. elle a permis de découvrir les lois régissant des systèmes complexes, d'extraire des modèles à partir de données expérimentales et de prédire le comportement futur des systèmes. En outre, SINDy peut être étendu pour traiter des défis tels que les données bruitées, les mesures manquantes ou les observations partielles.

Cette approche a ouvert de nouvelles perspectives pour l'étude des systèmes dynamiques complexes, en fournissant une méthode efficace pour extraire des connaissances à partir de données expérimentales. Les recherches dans le domaine de SINDy se poursuivent, avec des efforts visant à améliorer sa précision, à développer de nouvelles variantes de l'algorithme et à explorer de nouveaux domaines d'application.

1.3 Contexte historique de SINDy :

L'histoire de l'Identification parcimonieuse des Dynamiques Non Linéaires (SINDy) remonte au développement des techniques d'identification de systèmes et à l'émergence de la modélisation basée sur les données.

Le domaine de l'identification de systèmes a émergé au milieu du XXe siècle en tant que branche de la théorie du contrôle et de l'ingénierie. Son objectif principal était de développer des méthodes permettant d'estimer des modèles mathématiques, souvent sous la forme d'équations différentielles, qui représentaient de manière précise le comportement de systèmes dynamiques à partir de données d'entrée-sortie observées. Les techniques traditionnelles d'identification de systèmes se concentraient sur des modèles linéaires et supposaient une connaissance préalable de la structure du système.

Cependant, avec la complexité croissante des systèmes dynamiques et l'abondance de données, il est devenu nécessaire de disposer de méthodes capables de gérer des dynamiques non linéaires et complexes sans faire de fortes hypothèses. Cela a conduit au développement d'approches de modélisation basées sur les données, qui permettaient d'extraire des modèles directement à partir des données observées.

En 2016, l'article fondamental intitulé "Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems" de Steven L. Brunton, Joshua L. Proctor et J. Nathan Kutz [12] a été publié dans les Proceedings of the National Academy of Sciences.

Cet article a introduit la méthodologie SINDy en tant que technique puissante pour découvrir des équations régissant un système à partir de données.

Depuis son introduction, SINDy a attiré une attention considérable et a été appliquée dans différents domaines et applications. SINDy a également été étendue et améliorée pour relever des défis tels que les données bruitées, les mesures manquantes et les observations partielles.

Le développement et le succès de SINDy ont contribué à l'avancement des techniques de modélisation basées sur les données et ont ouvert de nouvelles possibilités pour découvrir les équations régissant les systèmes et comprendre les systèmes dynamiques complexes. Ce domaine de recherche est actif, avec des efforts continus pour affiner la méthodologie, développer de nouveaux algorithmes et l'appliquer à des systèmes de plus en plus complexes et diversifiés.

1.4 Applications et Extensions de SINDy : [11]

L'algorithme SINDy a récemment trouvé des applications dans l'identification de systèmes dynamiques de grande dimension, tels que les écoulements de fluides, en se basant sur les coefficients POD (Proper Orthogonal Decomposition) [12, 32, 33]. Par exemple, SINDy a été utilisé pour identifier le modèle de champ moyen généralisé de Noack et al. [38] à partir de données sur l'écoulement autour d'un cylindre, comme illustré dans la figure (Fig. 1.1). De plus, SINDy a été appliquée avec succès à l'identification de modèles en optique non linéaire [50] et en physique des plasmas [20].

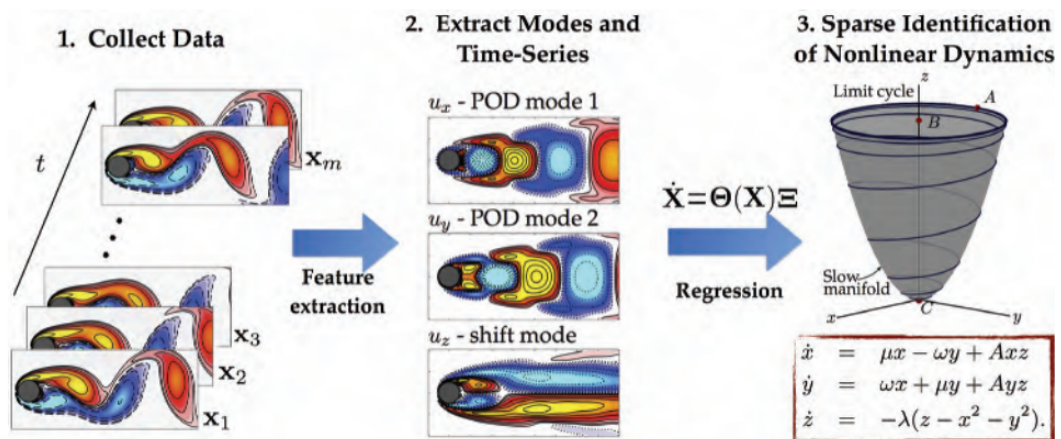


FIGURE 1.1 – Présentation schématique de l'identification de modèles non linéaires à partir de données de grande dimension à l'aide de l'identification parcimonieuse de la dynamique non linéaire (SINDy) [12]. Cette procédure est modulaire, de sorte que différentes techniques peuvent être utilisées pour les étapes d'extraction de caractéristiques et de régression. Dans cet exemple d'écoulement devant un cylindre, SINDy découvre le modèle de Noack et al. [37]. Modifié de Brunton et al. [12].

Une des forces de SINDy est sa formulation en termes de régression linéaire dans une bibliothèque non linéaire, ce qui lui confère une grande extensibilité. Récemment, le cadre SINDy a été étendu par Loiseau et Brunton [32] pour incorporer des contraintes physiques connues et des symétries dans les équations, en utilisant une optimisation contrainte basée sur le seuillage séquentiel des moindres carrés. Par exemple, des contraintes de préservation de l'énergie ont été imposées sur les non-linéarités quadratiques dans les équations de Navier-Stokes pour identifier des systèmes de fluides [32]. Ces contraintes sont connues pour favoriser la stabilité [34, 18, 3].

Cette approche a également montré que les bibliothèques polynomiales sont particulièrement utiles pour construire des modèles d'écoulements de fluides en utilisant les coefficients POD, ce qui permet d'obtenir des modèles interprétables liés à la projection de Galerkin classique [12, 32].

Loiseau et al. [33] ont également démontré la capacité de SINDy à identifier des modèles de systèmes dynamiques de grande dimension, tels que les écoulements de fluides, à partir de quelques mesures physiques provenant de capteurs, comme des mesures de portance et de traînée sur le cylindre présenté dans la figure (Fig. 1.1). Pour les systèmes avec des entrées et un contrôle, SINDy a été généralisé pour inclure ces éléments [13], et ces modèles s'avèrent très efficaces pour la commande prédictive du système [25]. De plus, il est possible d'étendre l'algorithme SINDy pour identifier des dynamiques avec des non-linéarités de fonctions rationnelles, des termes intégraux, ainsi que pour traiter des données fortement corrompues et incomplètes [52]. Des extensions récentes de SINDy incluent également l'incorporation de critères d'information pour la sélection objective de modèles [35], ainsi que l'identification de modèles avec des variables cachées en utilisant des coordonnées de retard [7]. Enfin, le cadre SINDy a été généralisé pour inclure des dérivées partielles, ce qui permet d'identifier des modèles d'équations aux dérivées partielles [43, 44].

De manière plus générale, l'utilisation de méthodes favorisant la parcimonie en dynamique est relativement récente et a donné lieu à diverses approches, telles que la découverte d'équations à partir de séries temporelles, la modélisation sans équation, la modélisation dynamique empirique, la modélisation du comportement émergent, le modèle autorégressif non linéaire avec entrées exogènes (NARMAX), et l'inférence automatisée de dynamiques. Toutes ces techniques relèvent de la catégorie des méthodes d'identification de systèmes, où des méthodes statistiques et d'apprentissage automatique sont utilisées pour identifier des systèmes dynamiques à partir de données. La plupart de ces méthodes impliquent une forme de régression des données sur la dynamique, et la principale distinction entre elles réside dans le degré de contrainte de cette régression.

Il convient également de mentionner des techniques plus récentes, telles que la programmation génétique utilisée par Bongard et Lipson [6] et Schmidt et Lipson [47], qui ont permis d'identifier la structure des dynamiques non linéaires de manière flexible et avec peu de contraintes sur leur forme. En outre, SINDy est étroitement lié à NARMAX [5], une méthode qui identifie la structure des modèles à partir de données de séries temporelles en utilisant une procédure des moindres carrés orthogonaux.

1.5 Les limites de SINDy :

Bien que l'algorithme de SINDy présente de nombreux avantages, il présente également certains défauts potentiels. Voici quelques-uns des défauts couramment associés à l'algorithme de SINDy :

- **Sensibilité aux données bruitées** : Comme de nombreux algorithmes d'apprentissage automatique, l'algorithme de SINDy peut être sensible aux données bruitées. Si les données d'entrée sont affectées par du bruit ou des erreurs de mesure, cela peut entraîner des erreurs dans les modèles identifiés par SINDy.
- **Sensibilité aux échantillonnages temporels inappropriés** : L'efficacité de l'algorithme de SINDy peut être réduite si les données d'entrée sont échantillonnées de manière inappropriée dans le temps. Des échantillons trop espacés ou des lacunes dans les données peuvent entraîner des erreurs dans la modélisation et l'identification des équations différentielles.
- **Problèmes d'identifiabilité** : Dans certains cas, il peut exister plusieurs ensembles d'équations différentielles qui peuvent générer les mêmes observations expérimentales. Cela peut rendre difficile l'identification unique des équations sous-jacentes à partir des données. L'algorithme de SINDy peut produire des modèles qui ne sont pas uniques ou qui ne reflètent pas complètement la dynamique réelle du système.

- **Complexité de modélisation limitée** : Bien que l'algorithme de SINDy soit efficace pour identifier des systèmes dynamiques simples à partir de données, il peut être limité dans sa capacité à modéliser des systèmes complexes ou hautement non linéaires. Il peut avoir du mal à capturer des comportements dynamiques complexes ou des interactions non linéaires fortes entre les variables.
- **Dépendance aux variables d'entrée choisies** : L'algorithme de SINDy repose sur la sélection appropriée des variables d'entrée pour l'identification des équations différentielles. La performance de l'algorithme peut dépendre de la connaissance préalable du système et du choix judicieux des variables d'entrée. Une mauvaise sélection des variables peut conduire à une mauvaise modélisation et à des résultats incorrects.

Il convient de noter que certains de ces défauts peuvent être atténués ou résolus en utilisant des variations de l'algorithme de base de SINDy, en combinant SINDy avec d'autres techniques d'apprentissage automatique ou en prenant des précautions appropriées lors de l'application de l'algorithme.

1.6 Conclusion :

Dans ce chapitre, nous avons présenté les principaux concepts de la méthode SINDy, son historique ainsi que ses applications dans l'identification de systèmes dynamiques de grande dimension et ses extensions. Nous avons également discuté des limites inhérentes à cette méthode.

CHAPITRE 2

L'OPÉRATEUR DE KOOPMAN ET LA DÉCOMPOSITION EN MODE DYNAMIQUE (DMD)

2.1 Introduction :

SINDy, l'opérateur de Koopman et DMD sont des méthodes utilisées pour analyser les systèmes dynamiques à partir de données. SINDy vise à découvrir les équations sous-jacentes régissant le système, tandis que l'opérateur de Koopman représente le système sous forme d'un opérateur linéaire agissant sur des fonctions observables, et DMD extrait les modes dynamiques dominants du système. Ces méthodes sont complémentaires dans l'analyse des données et la compréhension des systèmes, en combinant l'identification des équations, l'analyse des modes et la prédiction du comportement futur.

Dans ce chapitre, nous discuterons du concept et des mathématiques sous-jacentes à l'opérateur de Koopman et à l'algorithme DMD (Dynamic Mode Decomposition). Nous aborderons également la DMD avec contrôle (DMDc), l'EDMD et leur relation avec l'approximation de l'opérateur de Koopman à partir des données de l'espace d'état.

Notre objectif est d'utiliser des méthodes pilotées par les données (data-driven methods) pour approximer l'opérateur de Koopman, qui est un opérateur de dimension infinie qui fait évoluer linéairement l'état d'un système dynamique dans un espace de dimension infinie. Pour cela, nous utiliserons la méthode DMD, qui est une méthode de décomposition en modes dynamiques à partir de données expérimentales.

Plus précisément, la DMD utilise des mesures des états d'un système dynamique pour trouver les modes dominants de ce système à partir des données et reconstruire une représentation linéaire de l'évolution du système.

En effet, l'opérateur de Koopman permet de représenter la dynamique d'un système comme une transformation linéaire dans l'espace des fonctions. La DMD utilise cette représentation pour extraire des modes dynamiques et reconstruire l'évolution temporelle du système à partir de ces modes. Ainsi, la DMD est une méthode pratique pour l'analyse des systèmes dynamiques basée sur l'opérateur de Koopman, et qui permet de traiter efficacement les données expérimentales.

2.2 L'opérateur de Koopman : [11, 49]

2.2.1 Introduction à l'opérateur de Koopman :

La théorie de l'opérateur de Koopman est récemment apparue comme une perspective alternative pour les systèmes dynamiques en termes de l'évolution des mesures. En 1931, Bernard O. Koopman a démontré qu'il était possible de représenter un système dynamique non linéaire en termes d'un opérateur linéaire de dimension infinie agissant sur un espace de Hilbert de fonctions de mesure de l'état du système. Cet opérateur de Koopman est linéaire, et sa décomposition spectrale caractérise complètement le comportement d'un système non linéaire, de manière analogue à $\frac{dx}{dt} = Ax$. Cependant, il est également de dimension infinie, car il y a une infinité de degrés de liberté requis pour décrire l'espace de toutes les fonctions de mesure possibles de l'état. Cela pose de nouveaux défis. Obtenir des approximations matricielles de dimension finie de l'opérateur de Koopman est l'objet d'efforts de recherche intenses et promet de permettre des représentations globalement linéaires de systèmes dynamiques non linéaires. L'expression de la dynamique non linéaire dans un cadre linéaire est attrayante en raison de la richesse des techniques d'estimation et de contrôle optimal disponibles pour les systèmes linéaires et de la capacité à prédire analytiquement l'état futur du système. Obtenir une approximation de dimension finie de l'opérateur de Koopman a été difficile en pratique, car cela implique d'identifier un sous-espace engendré par un sous-ensemble des fonctions propres de l'opérateur de Koopman. L'opérateur de Koopman est souvent calculé à partir de données de l'espace d'état en utilisant des algorithmes de décomposition en modes dynamiques (DMD) ou des variantes telles que la DMDc et l'EDMD.

2.2.2 Les mathématiques derrière l'opérateur Koopman :

Considérons l'évolution d'un système dynamique autonome :

$$\frac{d}{dt}x = f(x) \quad (2.1)$$

Où x est un vecteur, $x \in \mathbb{R}^n$, et f est une fonction Lipschitzienne continue, garantissant l'existence et l'unicité des solutions de (2.1). Pour une formulation plus générale, voir [1].

Nous considérerons également le système dynamique en temps discret :

$$x_{k+1} = F(x_k) \quad (2.2)$$

Où la règle F applique l'espace d'état sur lui-même, c'est-à-dire $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Également connue sous le nom d'**application**, la dynamique en temps discret est plus générale que la formulation en temps continu présentée dans l'équation (2.1), englobant également les systèmes discontinus et hybrides.

Les dynamiques en temps discret peuvent être induites à partir des dynamiques en temps continu, où x_k est obtenu en échantillonnant discrètement la trajectoire de l'équation (2.1) dans le temps, de sorte que $x_k = x(k\Delta t)$. Le propagateur en temps discret $F_{\Delta t}$ est maintenant paramétré par l'étape de temps Δt . Pour un temps arbitraire t , le flot (*flow map*) F_t est définie comme suit :

$$F_t(x(t_0)) = x(t_0) + \int_{t_0}^{t_0+t} f(x(\tau))d\tau. \quad (2.3)$$

La perspective en temps discret est souvent plus naturelle lorsqu'on considère des données expérimentales et un contrôle numérique.

L'opérateur de Koopman fait avancer les fonctions de mesure de l'état avec le flot des dynamiques. Nous considérons des fonctions scalaire de mesures réelles $g : \mathbf{M} \rightarrow \mathbb{R}$, qui sont des éléments d'un espace de Hilbert de dimension infinie. Les fonctions g sont également communément appelées **observables**, cela n'a aucun rapport avec la notion *l'observabilité* de la théorie du contrôle. Typiquement, l'espace de Hilbert est donné par les fonctions carrées sommables sur M ; d'autres choix d'un espace de mesure sont également valables. L'opérateur de Koopman K_t est un opérateur linéaire de dimension infinie qui agit sur les fonctions de mesure g comme suit :

$$\mathcal{K}_t g = g \circ F_t, \quad (2.4)$$

Où \circ est l'opérateur de composition. Pour un système à **temps discret** avec un pas de temps Δt , cela devient :

$$\mathcal{K}_{\Delta t} g(x_k) = g(F_{\Delta t}(x_k)) = g(x_{k+1}), \quad (2.5)$$

Dans l'équation (2.4), l'opérateur de Koopman fait évoluer l'observable par rapport au temps de manière continue, tandis que dans l'équation (2.5), l'opérateur de Koopman fait évoluer la dynamique en temps discret par rapport à Δt , qui est l'intervalle entre k et $k + 1$ dans la série temporelle. Plus de détails sur les liens entre ces deux représentations peuvent être trouvés dans [26, 11, 39, 28, 36].

En d'autres termes, l'opérateur de Koopman définit un système dynamique linéaire en dimension infinie qui fait avancer l'observable $g_k = g(x_k)$ vers l'étape de temps suivante :

$$g(x_{k+1}) = \mathcal{K}_{\Delta t} g(x_k), \quad (2.6)$$

Notez que ceci est vrai pour n'importe quelle fonction observable g et pour n'importe quel état x_k . L'opérateur de Koopman est linéaire, une propriété héritée de la linéarité de l'opération d'addition dans les espaces de fonctions :

$$\mathcal{K}_t(\alpha_1 g_1(x) + \alpha_2 g_2(x)) = \alpha_1 g_1(F_t(x)) + \alpha_2 g_2(F_t(x)) \quad (2.7)$$

$$= \alpha_1 \mathcal{K}_t g_1(x) + \alpha_2 \mathcal{K}_t g_2(x) \quad (2.8)$$

Pour des systèmes dynamiques suffisamment réguliers, il est également possible de définir l'analogue continu dans le temps du système dynamique de Koopman dans l'équation (2.6) :

$$\frac{d}{dt} g = \mathcal{K}g, \quad (2.9)$$

L'opérateur \mathcal{K} est le générateur infinitésimal de la famille à un paramètre de transformations \mathcal{K}_t [1]. Il est défini par son action sur une fonction observable g :

$$\mathcal{K}g = \lim_{t \rightarrow 0} \frac{\mathcal{K}_t g - g}{t} = \lim_{t \rightarrow 0} \frac{g \circ F_t - g}{t} \quad (2.10)$$

Les systèmes dynamiques linéaires dans les équations (2.9) et (2.6) sont analogues aux systèmes dynamiques dans les équations (2.1) et (2.2), respectivement. Il est important de noter que l'état d'origine x peut être **l'observable**, et l'opérateur de dimension infinie \mathcal{K}_t fera avancer cette fonction. Cependant, la représentation simple de l'observable $g = \mathbf{x}$ dans une base choisie pour l'espace de Hilbert peut devenir arbitrairement complexe une fois itérée à travers la dynamique. En d'autres termes, trouver une représentation pour $\mathcal{K}\mathbf{x}$ peut ne pas être simple ou direct.

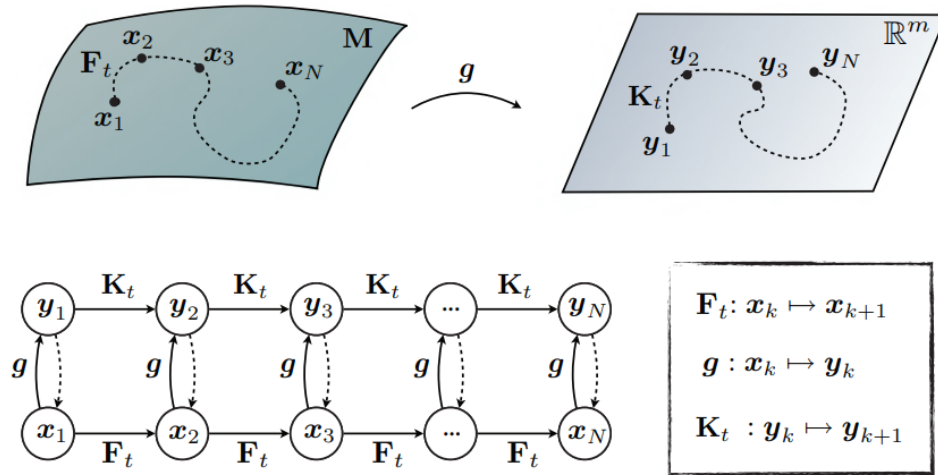


FIGURE 2.1 – Schéma illustrant l'opérateur de Koopman pour les systèmes dynamiques non linéaires, d'après [8]. Les lignes en pointillés de $y_k \rightarrow x_k$ indiquent que nous aimerions pouvoir récupérer l'état d'origine

Cependant, comme le montre la figure (Fig. 2.1), cet opérateur fait progresser la mesure du changement de dynamique au fil du temps [11, 39, 28].

2.3 La décomposition en mode dynamique (DMD) : [11, 49, 40]

2.3.1 Introduction à la DMD :

La Dynamic Mode Decomposition (DMD) est une technique de factorisation et de réduction de dimension pour les séquences de données. Dans sa forme la plus courante, elle traite des mesures séquentielles de grande dimension, extrait des structures cohérentes, isole le comportement dynamique et réduit les processus d'évolution complexes à leurs caractéristiques et composants essentiels. La décomposition est intimement liée à l'analyse de Koopman et, depuis son introduction, a engendré diverses extensions, généralisations et améliorations. Elle a été appliquée aux séquences de données numériques et expérimentales prises à partir de systèmes fluides simples à complexes et a également eu un impact au-delà de la dynamique des fluides dans, par exemple, la surveillance vidéo, l'épidémiologie, la neurobiologie et l'ingénierie financière.

La décomposition en modes dynamiques (DMD) a été développée par Schmid [45, 46], dans la communauté de la dynamique des fluides pour identifier des structures cohérentes spatio-temporelles à partir de données de haute dimension. DMD est basée sur la décomposition en modes orthogonaux propres (Proper orthogonal Decomposition, POD), qui utilise la décomposition en valeurs singulières (SVD) computationnellement efficace, de sorte qu'elle s'adapte bien pour fournir une réduction de dimensionnalité efficace dans les systèmes de grande dimension.

La DMD fournit une décomposition modale dans laquelle chaque mode est composé de structures spatialement corrélées qui ont le même comportement linéaire dans le temps (par exemple, des oscillations à une fréquence donnée avec une croissance ou une décroissance). Ainsi, la DMD ne fournit pas seulement une réduction de la dimensionnalité en termes du nombre de modes nécessaires pour représenter avec précision la dynamique du système, mais elle fournit également une décomposition physiquement significative des données en structures cohérentes spatialement avec des comportements temporels distincts.

Peu après le développement de l'algorithme DMD original [45, 46], Rowley, Mezic et leurs collaborateurs [42] ont établi une connexion importante entre la DMD et la théorie de Koopman. La DMD peut être formulée comme un algorithme permettant d'identifier le système dynamique linéaire le mieux adapté qui fait progresser les mesures de grande dimension vers l'avant dans le temps [54]. De cette manière, la DMD approxime l'opérateur de Koopman restreint à l'ensemble des mesures directes de l'état d'un système de grande dimension.

Enfin, la DMD est une plateforme extrêmement flexible, à la fois mathématiquement et numériquement, facilitant les innovations liées à la compression de données, la théorie de contrôle et les techniques multi-résolutions.

2.3.2 L'algorithme DMD :

Plusieurs algorithmes ont été proposés pour DMD, bien que nous présentions ici le cadre exact de la DMD développé par Tu et al. [54]. La DMD exacte repose sur une décomposition en valeurs singulières efficace et numériquement bien conditionnée, tout comme la formulation originale de Schmid [45].

DMD est intrinsèquement basé sur les données, donc la première étape consiste à collecter un certain nombre de paires de snapshots de l'état d'un système au fur et à mesure de son évolution dans le temps.

DMD repose sur l'hypothèse que l'état d'un système est relié au suivant par :

$$x_{k+1} = Ax_k, \quad (2.11)$$

Ici, $x(t) \in \mathbb{R}^n$ et $A \in \mathbb{R}^{n \times n}$, c'est la matrice décrivant l'évolution de l'état du système de manière continue dans le temps, l'indice k indique l'itération temporelle d'un système dynamique discret.

Les mesures simulées ou expérimentales pour x_k sont ensuite collectées à des intervalles de temps réguliers Δt pour devenir des *snapshots* à utiliser dans un système à temps discret. Le pas de temps Δt est suffisamment petit pour résoudre les fréquences les plus élevées dans la dynamique.

Ces snapshots sont collectés et ensuite organisés en deux matrices de données, X et X' :

$$X = \begin{bmatrix} | & | & \cdots & | \\ x_1 & x_2 & \cdots & x_{m-1} \\ | & | & \cdots & | \end{bmatrix} \quad (2.12)$$

$$X' = \begin{bmatrix} | & | & \cdots & | \\ x_2 & x_3 & \cdots & x_m \\ | & | & \cdots & | \end{bmatrix}$$

- Les lignes de la matrice de données X représentent les mesures par snapshot de temps, et les colonnes représentent les snapshots de temps (qui évoluent dans le temps).

- X' est une grande matrice de données décalée, elle avance d'un pas de temps vers le futur.

L'algorithme DMD cherche la décomposition spectrale principale (c'est-à-dire les valeurs propres et les vecteurs propres) de l'opérateur linéaire A qui correspond le mieux à l'avancement de X vers X' (les deux matrices de snapshots) dans le temps \rightarrow Construire un modèle dynamique linéaire :

$$X' \approx AX \quad (2.13)$$

Si nous supposons un échantillonnage uniforme dans le temps, cela devient :

$$x_{k+1} \approx Ax_k, \quad (2.14)$$

La matrice A la mieux adaptée établit un système dynamique linéaire qui fait avancer approximativement les mesures de snapshots dans le temps, ce qui peut être formulé mathématiquement comme un problème d'optimisation :

$$A = \underset{A}{\operatorname{argmin}} \|X' - AX\|_F = X'X^\dagger \quad (2.15)$$

Où : $\|\cdot\|_F$ est la norme de Frobenius.

Ainsi, par régression :

$$\implies A = X'X^\dagger \quad (2.16)$$

X^\dagger : la pseudo-inverse de X .

La pseudo-inverse peut être calculée à l'aide de la SVD de $X = U\Sigma V^*$ en tant que $X^\dagger = V\Sigma^{-1}U^*$. Où $*$ désigne la transposée conjuguée complexe. Les colonnes de U sont connues sous le nom de modes POD.

Il convient de noter à ce stade que la matrice A dans (2.14) ressemble étroitement à l'opérateur de Koopman dans (2.6), si nous choisissons des mesures linéaires directes de l'état, de sorte que $g(x) = x$. Cette connexion a été établie à l'origine par Rowley, Mezic et leurs collaborateurs [42] et a suscité un intérêt considérable à la fois pour la DMD et la théorie de Koopman.

- L'algorithme exacte de la DMD de Tu et al. [54] est donné par les étapes suivantes :

1. Premièrement, on prend la décomposition en valeurs singulières (SVD) de X :

$$X \approx \tilde{U}\tilde{\Sigma}\tilde{V}^* \quad (2.17)$$

Où : V^* : Transposée conjuguée de V .

$\tilde{U} \in \mathbb{C}^{n \times r}$; $\tilde{\Sigma} \in \mathbb{R}^{r \times r}$ et $\tilde{V} \in \mathbb{C}^{m \times r}$ ($\tilde{V}^* \in \mathbb{C}^{r \times m}$).

r est le rang de l'approximation réduite (SVD) de la matrice de données X , qui peut être soit exact, soit approximatif, $r \leq m$.

En pratique, choisir le rang approximatif r est l'une des étapes les plus importantes et subjectives de la DMD, et de la réduction de la dimensionnalité en général.

Les colonnes de la matrice \tilde{U} sont également connues sous le nom de *modes POD*, et elles satisfont $\tilde{U}^*\tilde{U} = I$. De même, les colonnes de \tilde{V} sont orthonormales et satisfont $\tilde{V}^*\tilde{V} = I$.

La réduction SVD est utilisée à cette étape de l'algorithme pour extraire de la matrice de données les informations les plus importantes (minimiser le rang). Plus précisément, si une structure de basse dimension est présente dans les données, les valeurs singulières diminueront rapidement pour atteindre zéro avec peut-être seulement quelques modes dominants.

2. D'après (2.15), une matrice d'approximation \bar{A} de la matrice complète A peut être calculée en utilisant la pseudo-inverse de X trouvée par la méthode SVD (2.17) :

$$A \approx \bar{A} = X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^* \quad (2.18)$$

Un modèle dynamique du processus peut être construit selon ce qui suit :

$$x_{k+1} = \bar{A}x_k; \quad (2.19)$$

Où x et \bar{A} ont la même dimension que les matrices décrites précédemment dans (2.14).

Cependant, le calcul peut être excessivement coûteux en termes de temps de calcul et de mémoire si $n \gg 1$.

Donc il n'est pas recommandé de calculer directement la matrice d'approximation \bar{A} dans ce cas, car \bar{A} est une matrice de taille $(n \times n)$ et n peut être très grand.

À la place, lorsque $r \ll n$, nous pouvons calculer une approximation de faible rang \tilde{A} , qui est un modèle plus compact et efficace sur le plan computationnel de taille $(r \times r)$ et peut être trouvé en projetant x_k sur un sous-espace linéaire de dimension r .

Cependant, nous sommes seulement intéressés par les r premières valeurs propres et vecteurs propres de A , et nous pouvons donc projeter A sur les modes POD de U en utilisant une transformation de base de cette façon :

$$\tilde{A} = \tilde{U}^* \bar{A} \tilde{U} = \tilde{U}^* X' \tilde{V} \tilde{\Sigma}^{-1}. \quad (2.20)$$

De plus, le calcul est efficace car $\tilde{A} \in \mathbb{R}^{r \times r}$ et $r \ll n$.

La clé de l'observation ici est que malgré sa taille réduite, la matrice réduite \tilde{A} conserve les mêmes valeurs propres non nulles que la matrice complète A , qui codent le comportement à long terme du système. Cet opérateur dynamique réduit peut toujours capturer les dynamiques dominantes du système, car il est obtenu en projetant A ($\bar{A} \approx A$) sur les modes POD dominants (les colonnes de la matrice \tilde{U}), ce qui nous permet de représenter le comportement du système dans un espace de dimension inférieure.

L'opérateur dynamique réduit \tilde{A} est une approximation linéaire de faible rang de l'opérateur dynamique réel A , qui décrit l'évolution des modes POD dans le temps.

La matrice d'ordre réduit \tilde{A} définit un modèle linéaire de basse-dimension pour la dynamique du vecteur de coefficients POD \tilde{x} :

$$\tilde{x}_{k+1} = \tilde{A} \tilde{x}_k \quad (2.21)$$

Il est possible de reconstruire l'état de haute-dimension x (l'état complet) à partir de l'état réduit \tilde{x} en utilisant la matrice \tilde{U} , comme suit :

$$x = \tilde{U} \tilde{x} \quad (2.22)$$

-
3. Calculer la décomposition propre (spectrale) de \tilde{A} :

$$\tilde{A} W = W \Lambda \quad (2.23)$$

Les entrées de la matrice diagonale Λ sont les valeurs propres de DMD, qui correspondent également aux valeurs propres de la matrice complète A : croissance/décroissance, oscillations. Les colonnes de W sont des vecteurs propres de \tilde{A} et fournissent une transformation de coordonnées qui diagonalise la matrice. Ces colonnes peuvent être considérées comme des combinaisons linéaires d'amplitudes de modes POD qui se comportent linéairement avec un seul motif temporel donné par λ .

-
4. Finalement, nous pouvons calculer les modes DMD Φ en utilisant les vecteurs propres de \tilde{A} (c'est-à-dire les colonnes de W) et les corrélations spatiales correspondantes entre les mesures (c'est-à-dire les colonnes de $X' \tilde{V} \tilde{\Sigma}^{-1}$), comme suit :

$$\Phi = X' \tilde{V} \tilde{\Sigma}^{-1} W, \quad (2.24)$$

Ici, X' est la matrice instantanée décalée dans le temps, \tilde{V} et $\tilde{\Sigma}$ sont obtenus à partir de la SVD de X' , et W est la matrice des vecteurs propres du système réduit \tilde{A} . Les colonnes de Φ correspondent aux modes DMD, qui décrivent la structure spatiale du comportement dynamique du système en tant que corrélations spatiales entre les mesures.

Remarquablement, ces modes DMD sont des vecteurs propres de la matrice A de grande dimension correspondant aux valeurs propres dans Λ , comme le montre Tu et al. [54] :

$$A\Phi = (X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^*)\underbrace{(X'\tilde{V}\tilde{\Sigma}^{-1}W)}_{\tilde{A}} \quad (2.25)$$

$$= X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{A}W \quad (2.26)$$

$$= X'\tilde{V}\tilde{\Sigma}^{-1}W\Lambda \quad (2.27)$$

$$= \Phi\Lambda. \quad (2.28)$$

Parce que A est une représentation approximative de l'opérateur de Koopman restreint à un sous-espace de mesures linéaires de dimension finie, nous nous intéressons souvent aux vecteurs propres Φ et aux valeurs propres Λ de A tels que : $A\Phi = \Phi\Lambda$ (2.28).

Donc, cela nous montre que nous pouvons obtenir les valeurs et vecteurs propres de la grande matrice A sans même la calculer réellement, nous travaillons seulement sur \tilde{A} .

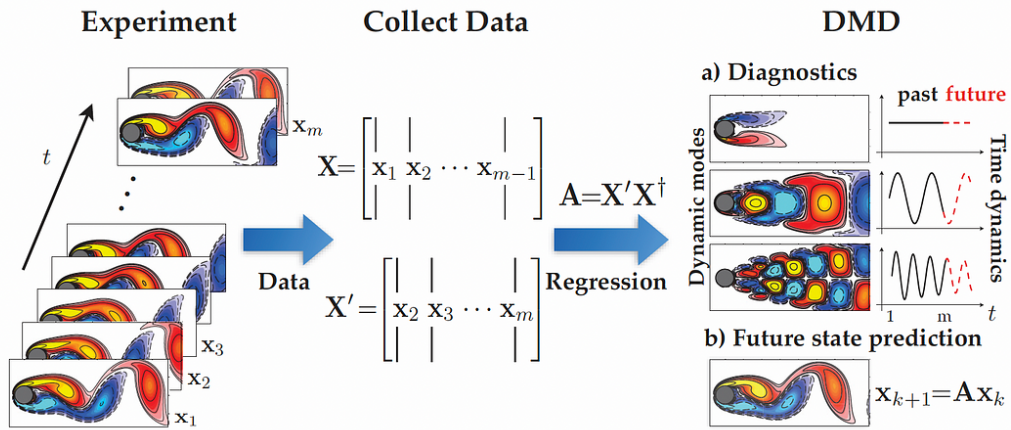


FIGURE 2.2 – Vue d'ensemble de la DMD illustrée sur l'écoulement de fluide autour d'un cylindre circulaire à un nombre de Reynolds de 100. Reproduite à partir de [30].

2.3.3 La décomposition spectrale et le développement DMD :

La Décomposition Spectrale et le développement DMD (DMD Expansion) sont deux techniques utilisées dans l'analyse des systèmes dynamiques.

La Décomposition Spectrale est une méthode utilisée pour décomposer une matrice en ses composantes, afin de mieux comprendre son comportement. La méthode consiste à trouver les valeurs propres et les vecteurs propres de la matrice, qui peuvent être utilisés pour diagonaliser la matrice. La matrice diagonalisée est plus facile à analyser car elle est composée de composantes plus simples, et les valeurs propres et les vecteurs propres peuvent fournir des informations sur le comportement du système.

DMD Expansion, en revanche, est une méthode utilisée pour analyser des données de séries temporelles de systèmes dynamiques. Elle consiste à trouver un ensemble de modes, appelés modes DMD, qui peuvent être utilisés pour approximer le comportement du système dans le temps.

Les modes DMD sont obtenus en appliquant l'algorithme DMD aux données de séries temporelles, ce qui implique de trouver les vecteurs propres et les valeurs propres d'un opérateur linéaire qui décrit l'évolution du système dans le temps. Les modes DMD sont ensuite utilisés pour reconstruire le comportement du système, permettant ainsi la prédiction et le contrôle de son comportement.

La Décomposition Spectrale et le développement DMD sont des outils puissants pour l'analyse des systèmes dynamiques, et peuvent être utilisés ensemble pour obtenir une compréhension plus complète du comportement du système.

L'un des aspects les plus importants de la DMD est la capacité à développer l'état du système en termes d'une décomposition spectrale basée sur les données :

$$x_k = \sum_{j=1}^r \phi_j \lambda_j^{k-1} b_j = \Phi \Lambda^{k-1} \mathbf{b}, \quad (2.29)$$

Où ϕ_j sont les modes de DMD (vecteurs propres de la matrice A), λ_j sont les valeurs propres de la DMD (valeurs propres de la matrice A), et b_j est l'amplitude du mode. Le vecteur \mathbf{b} des amplitudes de mode est généralement calculé comme suit :

$$\mathbf{b} = \Phi^\dagger x_1. \quad (2.30)$$

L'expansion spectrale ci-dessus (2.29) peut également être écrite en temps continu en introduisant les valeurs propres continues $\omega = \log(\lambda)/\Delta t$:

$$x(t) = \sum_{j=1}^r \phi_j e^{w_j t} b_j = \Phi e^{\Omega t} \mathbf{b}. \quad (2.31)$$

À $t=0$: $x(0) = \Phi \mathbf{b} \implies \mathbf{b} = \Phi \backslash x(0)$

Où Ω est une matrice diagonale contenant les valeurs propres en temps continu ω_j .

En résumé, à partir de Φ , nous pouvons reconstruire notre approximation de la dynamique temporelle de $x(t)$ en projetant nos approximations sur une solution future, également connue sous le nom de **prédiction d'état futur**.

Pour les systèmes avec des comportements périodiques ou quasi-périodiques dominants et de grande dimension, DMD donne une très bonne approximation même s'ils sont fortement non linéaires.

2.3.4 La décomposition en mode dynamique avec contrôle (DMDc) : [11, 40]

Un avantage majeur de DMD est sa capacité à décrire des systèmes dynamiques complexes et de grande dimension en termes d'un petit nombre de modes dominants qui représentent des structures cohérentes spatio-temporelles. La réduction de la dimensionnalité du système de n (souvent des millions ou des milliards) à r (dizaines ou centaines) permet une prédiction plus rapide et une estimation à plus faible latence, ce qui se traduit généralement par des contrôleurs avec une performance et une robustesse accrues.

Proctor et al. [40] ont étendu l'algorithme DMD pour inclure l'effet de l'actionnement et du contrôle, dans l'algorithme appelé DMD avec contrôle (DMDc). Il a été observé que l'application naïve de DMD aux données d'un système avec actionnement (actuation) résultait souvent en une dynamique incorrecte, car les effets de la dynamique interne étaient confondus avec les effets de l'actionnement. DMDc a été initialement motivé par le problème de la caractérisation et du contrôle de la propagation de maladies, où il est déraisonnable d'arrêter les efforts d'intervention (par exemple, les vaccinations) simplement pour obtenir une caractérisation de la dynamique non forcée [41]. Au lieu de cela, si le signal d'actionnement est mesuré, une nouvelle régression DMD peut être formulée afin de dissocier l'effet de la dynamique interne de celui de l'actionnement et du contrôle. Par la suite, cette approche a été étendue pour effectuer DMDc sur des mesures fortement sous-échantillonnées ou compressées par Bai et al. [2].

La méthode DMDc cherche à identifier les meilleurs opérateurs linéaires A et B qui satisfont approximativement les dynamiques suivantes sur les données de mesure :

$$x_{k+1} \approx Ax_k + Bu_k \quad (2.32)$$

La variable d'état x et la commande d'actionnement u sont collectées et organisées dans les matrices de données suivantes :

- La matrice des snapshots X et la matrice de snapshots décalés dans le temps X' de (2.12) :

$$X = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_m \\ | & & | \end{bmatrix}, X' = \begin{bmatrix} | & & | \\ x_2 & \cdots & x_{m+1} \\ | & & | \end{bmatrix} \quad (2.33)$$

- Une matrice de l'historique d'entrée d'actionnement (*matrix of the actuation input history*) Υ est assemblée :

$$\Upsilon = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & & | \end{bmatrix} \quad (2.34)$$

Les dynamiques dans (2.32) peuvent être écrites en termes de matrices de données :

$$X' \approx AX + B\Upsilon \quad (2.35)$$

Comme dans l'algorithme DMD (voir Section 2.3.2), les valeurs propres et vecteurs propres principaux de l'opérateur linéaire A sont obtenus par réduction de dimension et régression. Donc, on distingue 2 cas pour la DMDc :

1. **Premier cas** : Si la matrice d'actionnement (*Actuation Matrix*) B est **connue**, il est alors facile de corriger l'effet de l'actionnement et d'identifier la décomposition spectrale de A en remplaçant X' par $X' - B\Upsilon$ dans l'algorithme DMD.

De (2.35), on tire :

$$(X' - B\Upsilon) \approx AX, \quad (2.36)$$

La résolution de la correspondance A peut être effectuée de manière similaire à (2.16) (Par régression).

Avec $(X' - B\Upsilon)$ est le nouveau X' dans $A = X'X^\dagger$.

Encore une fois, la décomposition en valeurs singulières tronquée de X (??) donne la factorisation matricielle $\tilde{U}\tilde{\Sigma}\tilde{V}^*$. Ainsi, \bar{A} l'approximation de A est donnée par la description suivante :

$$A \approx \bar{A} = (X' - B\Upsilon)\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^*. \quad (2.37)$$

Notez que si les *snapshots* de contrôle sont $u_j = 0$, alors la dérivation est équivalente à la méthode DMD.

Un modèle dynamique à la fois du processus calculé et de la matrice d'entrée donnée peut être construit et décrit par ce qui suit :

$$x_{k+1} = \bar{A}x_k + Bu_k, \quad (2.38)$$

Cependant, si $r \ll n$, un modèle plus compact et plus efficace en termes de calcul peut être trouvé en projetant \bar{A} sur les modes POD dominants de U en utilisant la même transformation de base telle que décrite précédemment pour la méthode DMD.

Le modèle d'ordre réduit peut être dérivé de la manière suivante :

$$\tilde{x}_{k+1} = \tilde{U}^* \bar{A} \tilde{U} \tilde{x}_k + \tilde{U}^* B u_k \quad (2.39)$$

$$= \tilde{U}^* (X' - B\Upsilon) \tilde{V} \tilde{\Sigma}^{-1} \tilde{x}_k + \tilde{U}^* B u_k \quad (2.40)$$

$$= \tilde{A} \tilde{x}_k + \tilde{B} \tilde{u}_k \quad (2.41)$$

L'approximation à faible dimension de la matrice A est donnée par ce qui suit :

$$\tilde{A} = \tilde{U}^* (X' - B\Upsilon) \tilde{V} \tilde{\Sigma}^{-1}. \quad (2.42)$$

La décomposition en valeurs propres de \tilde{A} définie par $\tilde{A}W = W\Lambda$ donne des vecteurs propres qui peuvent être utilisés pour trouver les modes dynamiques. Tout comme pour la méthode DMD exacte, les modes dynamiques Φ peuvent être trouvés avec la description suivante :

$$\Phi = (X' - B\Upsilon) \tilde{V} \tilde{\Sigma}^{-1} W. \quad (2.43)$$

2. **Deuxième cas :** Lorsque B est **inconnu**, à la fois A et B doivent être identifiés simultanément en résolvant le problème d'optimisation des moindres carrés suivant :

$$\min_{\begin{bmatrix} A \\ B \end{bmatrix}} \left\| X' - \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X \\ \Upsilon \end{bmatrix} \right\|^2 \quad (2.44)$$

Dans ce cas, la dynamique (2.35) peut être reformulée comme suit :

$$X' \approx \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X \\ \Upsilon \end{bmatrix} = G\Omega. \quad (2.45)$$

Où la solution est donnée par :

$$[A \ B] = X' \begin{bmatrix} X \\ \Upsilon \end{bmatrix}^\dagger. \quad (2.46)$$

On a $G = \begin{bmatrix} A & B \end{bmatrix}$ est obtenue par régression des moindres carrés et $\Omega = \begin{bmatrix} X \\ \Upsilon \end{bmatrix}$, donc (2.46) devient :

$$G \approx X' \Omega^\dagger. \quad (2.47)$$

La matrice $\Omega = \begin{bmatrix} X^* & \Upsilon^* \end{bmatrix}^*$ est généralement une matrice de données de grande dimension, qui peut être approximée à l'aide de la SVD :

$$\Omega = \tilde{U} \tilde{\Sigma} \tilde{V}^* \quad (2.48)$$

Notez que la valeur de troncature de Ω doit être plus grande que celle de X .

De l'équation (2.48) on tire :

$$\Omega^\dagger = \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}^*, \quad (2.49)$$

Cela fournit une approximation de G :

$$G \approx \bar{G} = X' \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}^*. \quad (2.50)$$

La matrice \tilde{U} doit être divisée en deux matrices, $\tilde{U} = \begin{bmatrix} \tilde{U}_1^* & \tilde{U}_2^* \end{bmatrix}^*$, pour fournir des bases pour X et Ω .

Nous pouvons maintenant trouver des approximations des matrices A et B :

$$[A, B] \approx [\bar{A}, \bar{B}] \quad (2.51)$$

$$\approx \begin{bmatrix} X' \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}_1^* & X' \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}_2^* \end{bmatrix} \quad (2.52)$$

Un modèle dynamique utilisant les matrices \bar{A} et \bar{B} peut être construit et décrit par ce qui suit :

$$x_{k+1} = \bar{A}x_k + \bar{B}u_k \quad (2.53)$$

Mais pour un système de grande dimension où $n \gg 1$, cela devient prohibitif en termes de calcul.

Ici, nous cherchons à nouveau un modèle d'ordre réduit de rang $r \ll n$, où une transformation est nécessaire.

Contrairement à la méthode DMD, les vecteurs singuliers gauche tronqués \tilde{U} ne peuvent pas être utilisés pour définir l'espace sur lequel l'état évolue. Pour (2.52), les vecteurs singuliers gauche tronqués de Ω définissent l'espace d'entrée (*Input space*). Pour trouver une transformation linéaire, nous utilisons un sous-espace d'ordre réduit du sous-espace de sortie (*Output subspace*). Cette observation fondamentale permet à la méthode DMDC de découvrir une représentation d'ordre réduit des dynamiques A et de la matrice d'entrée B .

Pour trouver le sous-espace d'ordre réduit du sous-espace de sortie, une deuxième décomposition en valeurs singulières (SVD) est nécessaire pour la matrice de données de l'espace de sortie X' . Ça veut dire que \tilde{U} de (2.48) fournit une base réduite pour l'espace d'entrée (*input space*), tandis que \hat{U} de :

$$X' = \hat{U}\hat{\Sigma}\hat{V}^* \quad (2.54)$$

définit une base réduite pour l'espace de sortie (*output space*). Il est alors possible d'approximer $G = \begin{bmatrix} A & B \end{bmatrix}$ en projetant sur cette base :

$$\tilde{G} = \hat{U}^* G \begin{bmatrix} \hat{U} \\ I \end{bmatrix}. \quad (2.55)$$

Les matrices projetées résultantes \tilde{A} et \tilde{B} dans \tilde{G} sont :

$$\tilde{A} = \hat{U}^* \tilde{A} \hat{U} = \hat{U}^* X' \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}_1^* \hat{U}. \quad (2.56)$$

$$\tilde{B} = \hat{U}^* \tilde{B} = \hat{U}^* X' \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}_2^*. \quad (2.57)$$

Nous pouvons alors former l'équation d'ordre réduit telle que donnée par (2.41) suivante :

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}\tilde{u}_k \quad (2.58)$$

Plus important encore, il est possible de récupérer les vecteurs propres Φ de DMD, de la décomposition propre $\tilde{A}W = W\Lambda$:

$$\Phi = X' \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}_1^* \hat{U} W. \quad (2.59)$$

2.3.5 Décomposition en Mode Dynamique étendu (*Extended DMD, EDMD*) :

Maintenant, nous allons introduire une variante de la méthode DMD appelée "Extended Dynamic Mode Decomposition" (EDMD). La DMD étendue (EDMD) est une extension de la DMD qui permet de modéliser des systèmes fortement non linéaires en identifiant les changements non linéaires de coordonnées nécessaires pour approximer l'opérateur de Koopman.

Dans la méthode DMD traditionnelle, les données sont considérées comme évoluant linéairement entre les mesures (2.11). Cela signifie que les relations entre les différentes variables dans le système sont supposées être linéaires. Cependant, de nombreux systèmes réels présentent des comportements non linéaires, où les relations entre les variables ne peuvent pas être approximées par des fonctions linéaires simples.

L'EDMD remédie à cette limitation en introduisant des transformations non linéaires des caractéristiques dans le processus de décomposition dynamique. Au lieu d'utiliser les données brutes, l'EDMD utilise une fonction de transformation non linéaire pour injecter les données dans un espace de caractéristiques de dimension réduite. Cette fonction de transformation est choisie en fonction de la connaissance du système et des caractéristiques que l'on souhaite capturer.

En effectuant cette transformation non linéaire, l'EDMD permet de capturer les relations non linéaires entre les variables du système. Ainsi, les structures cohérentes et les dynamiques non linéaires peuvent être mieux représentées et identifiées par la décomposition dynamique. Les modes extraits par l'EDMD reflètent alors les comportements non linéaires du système, ce qui permet une meilleure compréhension et prédiction de son évolution.

En résumé, l'approche de la décomposition en mode dynamique étendue (EDMD) a été introduite par Williams et al. [57] pour résoudre la limitation de DMD en introduisant des transformations non linéaires des caractéristiques, ce qui permet de capturer les comportements non linéaires des systèmes et de fournir une meilleure représentation des dynamiques sous-jacentes. Cela rend l'EDMD plus adapté à l'analyse de systèmes complexes et non linéaires par rapport à la méthode DMD traditionnelle.

L'algorithme de DMD étendu [57] est essentiellement le même que celui de DMD standard [54], sauf qu'au lieu de réaliser une régression sur des mesures directes de l'état, une régression DMD linéaire de meilleure qualité (*best-fit*) est réalisée sur un vecteur augmenté contenant des mesures non linéaires de l'état.

Dans l'EDMD, un état augmenté $z \in \mathbb{R}^p$ est construit à partir de mesures non linéaires de l'état \mathbf{x} données par les fonctions θ_k .

$$z = \Theta^T(x) = \begin{bmatrix} \theta_1(x) \\ \theta_2(x) \\ \vdots \\ \theta_p(x) \end{bmatrix}. \quad (2.60)$$

Ici, le vecteur Θ est la collection de mesures du système qui peut contenir l'état d'origine du système \mathbf{x} ainsi que des mesures non linéaires, il arrive souvent que $p \gg n$, où p est le rang de l'état augmenté. Une fois z trouvé, deux matrices de données sont construites, comme dans le cas de la DMD (2.12) :

$$Z = \begin{bmatrix} | & | & & | \\ z_1 & z_2 & \cdots & z_m \\ | & | & & | \end{bmatrix} \quad Z' = \begin{bmatrix} | & | & & | \\ z_2 & z_3 & \cdots & z_{m+1} \\ | & | & & | \end{bmatrix} \quad (2.61)$$

Ici, $z_k = \Theta^T(x_k) = \Theta^T(x(k\Delta t))$, où nous supposons que les données sont échantillonnées à des intervalles réguliers dans le temps, pour simplifier. Comme dans DMD, un opérateur de matrice linéaire de meilleur ajustement (*best fit*) A_Z est construit qui mappe (maps) Z dans Z' :

$$A_Z = \operatorname{argmin}_{A_Z} \| Z' - A_Z Z \|_F = Z' Z^\dagger. \quad (2.62)$$

Cette régression peut être écrite en termes des matrices de données $\Theta(X)$ et $\Theta(X')$:

$$A_Z = \operatorname{argmin}_{A_Z} \| \Theta^T(X') - A_Z \Theta^T(X) \| = \Theta^T(X') (\Theta^T(X))^\dagger. \quad (2.63)$$

Étant donné que le vecteur augmenté \mathbf{z} peut être considérablement plus grand que l'état \mathbf{x} , il est généralement nécessaire d'utiliser des méthodes de noyau pour calculer cette régression [58]. L'utilisation de méthodes de noyau (kernel) pour approximer l'opérateur de Koopman avec DMD est devenue un sujet de recherche important ces dernières années. En principe, la bibliothèque enrichie Θ fournit une base plus grande dans laquelle approximer l'opérateur de Koopman. Il a été démontré récemment que dans la limite des données (snapshots) infinies, l'opérateur DMD étendu converge vers l'opérateur de Koopman projeté sur le sous-espace engendré par Θ [27]. Cependant, si Θ ne couvre pas un sous-espace invariant de Koopman, alors l'opérateur projeté peut ne pas avoir de ressemblance avec l'opérateur de Koopman d'origine, car toutes les valeurs et vecteurs propres peuvent être différents. En fait, il a été démontré que l'opérateur DMD étendu aura des valeurs et vecteurs propres erronés à moins qu'il ne soit représenté en termes d'un sous-espace invariant de Koopman [9].

Il est donc essentiel d'utiliser des techniques de validation et de validation croisée pour s'assurer que les modèles EDMD ne sont pas surajustés et qu'ils sont bien adaptés au système réel, comme discuté ci-dessous. Par exemple, il a été démontré que EDMD ne peut pas contenir l'état d'origine x en tant que mesure et représenter un système qui a plusieurs points fixes, des orbites périodiques ou d'autres attracteurs, car ces systèmes ne peuvent pas être topologiquement conjugués à un système linéaire de dimension finie [9].

2.4 Conclusion :

En conclusion, dans ce chapitre, nous avons exploré le concept de l'opérateur de Koopman et l'algorithme de décomposition en modes dynamiques (DMD). Nous avons également discuté des variantes de la DMD, telles que la DMD avec contrôle (DMDc) et l'EDMD, et de leur relation avec la découverte de l'opérateur de Koopman à partir des données de l'espace d'état.

Au cours de notre étude, nous avons souligné que la recherche directe de l'opérateur de Koopman peut être complexe, même pour des systèmes simples. Cependant, au fur et à mesure que nous approfondissons la recherche de cet opérateur, nous nous confrontons à des systèmes de dimension infinie. Dans de tels cas, l'utilisation d'expansions de Taylor ou de séries de Fourier naïves ne suffit généralement pas à obtenir un bon opérateur de Koopman. Pour surmonter ces difficultés, les méthodes pilotées par les données, telles que la DMD et ses variantes, offrent une alternative précieuse. Elles permettent d'approximer efficacement l'opérateur de Koopman en utilisant les données disponibles, ouvrant ainsi de nouvelles perspectives pour l'analyse et la modélisation des systèmes dynamiques.

Nous avons également constaté que la méthode DMD offre une approche prometteuse pour approximer l'opérateur de Koopman, en particulier pour les systèmes dynamiques non linéaires. En extrayant les modes caractéristiques des systèmes à partir des données, la DMD permet une analyse précise de leur comportement et offre des capacités de modélisation et de prédiction de leur évolution future. Cela constitue un avantage significatif pour de nombreuses applications dans divers domaines.

En résumé, la méthode DMD constitue un outil précieux pour l'analyse et la modélisation des systèmes dynamiques, en particulier lorsque la recherche directe de l'opérateur de Koopman devient complexe. Cependant, des recherches supplémentaires sont nécessaires pour perfectionner ces méthodes et élargir leur applicabilité dans des scénarios plus complexes. En exploitant les données disponibles et en développant de nouvelles techniques, nous pourrions continuer à améliorer notre compréhension des systèmes dynamiques et à exploiter leur potentiel dans diverses applications.

CHAPITRE 3

IDENTIFICATION PARCIMONIEUSE DE LA DYNAMIQUE NON LINÉAIRE 'SINDY'

3.1 Introduction

Ce chapitre se focalise sur la méthode d'identification parcimonieuse de la dynamique non linéaire appelée SINDy (Sparse Identification of Nonlinear Dynamics). Il présente une approche alternative à la modélisation traditionnelle des systèmes dynamiques à partir de données, en utilisant la régression symbolique et la représentation parcimonieuse. La régression symbolique est utilisée pour trouver des fonctions non linéaires décrivant les relations entre les variables mesurées et les dynamiques observées. La méthode de représentation parcimonieuse permet d'identifier de manière efficace les termes pertinents d'un modèle non linéaire en imposant une contrainte de parcimonie. L'algorithme SINDy exploite la dispersion des coefficients du modèle afin de déterminer la structure sous-jacente de la dynamique non linéaire.

Ce chapitre explique en détail l'algorithme SINDy, sa formulation mathématique et les étapes clés de l'identification parcimonieuse. Il présente également un exemple d'application qui met en évidence l'efficacité de SINDy dans la découverte de modèles non linéaires à partir de données. Enfin, l'impact du bruit sur les données est abordé, et un algorithme correspondant est décrit pour traiter ce cas.

3.2 Contexte : [12]

La modélisation des dynamiques à partir de données a une histoire longue et fructueuse, donnant lieu à de puissantes techniques d'identification de systèmes [31]. De nombreuses méthodes ont émergé pour répondre au besoin de comprendre des structures flexibles complexes, comme le télescope spatial Hubble ou la station spatiale internationale. Ces modèles ont été largement utilisés dans pratiquement toutes les branches de l'ingénierie et des mathématiques appliquées, notamment pour le contrôle en boucle fermée basé sur des modèles. Cependant, les méthodes d'identification des systèmes supposent en générale la structure du modèle connue [23].

Ce travail se démarque des travaux fondamentaux sur l'identification de systèmes et s'appuie plutôt sur la régression symbolique et la représentation parcimonieuse. Plus précisément, la régression symbolique est utilisée pour trouver des fonctions non linéaires qui décrivent les relations entre les variables et les dynamiques mesurées (c'est-à-dire les dérivées temporelles). Traditionnellement, la complexité du modèle est équilibrée avec sa capacité de description en utilisant des arguments de parcimonie tels que le front de Pareto. Dans cette étude, nous utilisons la représentation parcimonieuse pour déterminer les termes pertinents du modèle, dans un cadre efficace et évolutif.

3.2.1 Régression symbolique et apprentissage automatique : [12]

La régression symbolique consiste à déterminer une fonction qui relie les données d'entrée et de sortie, et peut être considérée comme une forme d'apprentissage automatique. Habituellement, la fonction est obtenue à l'aide de la programmation génétique, un algorithme évolutionnaire qui construit et teste des fonctions candidates à partir de blocs de base simples [29]. Ces fonctions sont ensuite modifiées selon un ensemble de règles évolutionnaires, et des générations de fonctions sont testées jusqu'à ce qu'un niveau de précision prédéterminé soit atteint.

Récemment, la régression symbolique a été appliquée à des données provenant de systèmes dynamiques, permettant de découvrir des équations différentielles ordinaires à partir de données de mesure [48]. Pour éviter le sur-apprentissage avec la régression symbolique et la programmation génétique, une contrainte de parcimonie doit être imposée. Dans l'étude [48], les auteurs choisissent des équations candidates qui se situent sur le front de Pareto de la complexité, afin de trouver un équilibre entre la complexité du modèle et sa capacité de description.

3.2.2 Représentation parcimonieuse et acquisition comprimée : [12]

Dans de nombreux problèmes de régression, seuls quelques termes dans la régression sont importants, et un mécanisme de sélection de caractéristiques parcimonieuses est nécessaire. Par exemple, considérons des mesures de données y dans R^m qui peuvent être une combinaison linéaire de colonnes d'une bibliothèque de caractéristiques $\Theta \in R^{m \times p}$; la combinaison linéaire des colonnes est donnée par les entrées du vecteur $\xi \in R^p$ de sorte que :

$$y = \Theta \xi \tag{3.1}$$

En effectuant une régression standard pour résoudre le vecteur ξ , on obtient une solution avec des contributions non nulles dans chaque élément.

Cependant, si la parcimonie de ξ est souhaitée, de sorte que la plupart des entrées soient nulles, il est possible d'ajouter un terme de régularisation L1 à la régression, ce qui donne le LASSO [22, 24, 51] :

$$\xi = \underset{\xi'}{\operatorname{argmin}} \|\Theta \xi' - y\|_2 + \lambda \|\xi'\|_1 \quad (3.2)$$

Le paramètre λ pondère la contrainte de parcimonie. Cette formulation est étroitement liée au cadre de l'acquisition comprimée (*compressed sensing*), qui permet de déterminer le vecteur parcimonieux ξ à partir de mesures aléatoires relativement peu nombreuses et incohérentes [21, 15, 16, 14, 4, 53]. La solution parcimonieuse ξ de l'équation (3.1) peut également être utilisée pour des schémas de classification parcimonieuse, tels que la représentation parcimonieuse pour la classification (SRC) [59]. Il est important de noter que, les architectures de l'acquisition comprimée et de la représentation parcimonieuse sont convexes et s'adaptent bien aux problèmes de grande taille.

3.3 Identification parcimonieuse de la dynamique non linéaire 'SINDy' : [11]

L'algorithme d'identification parcimonieuse de la dynamique non linéaire (SINDy) [12] contourne la recherche combinatoire inextricable de toutes les structures de modèles possibles, en exploitant le fait que de nombreux systèmes dynamiques :

$$\frac{d}{dx}x = f(x) \quad (3.3)$$

ont une dynamique f qui ne présente que quelques termes actifs dans l'espace des fonctions possibles du côté droit. Par exemple, les équations de Lorenz dans (3.10) ne contiennent que quelques termes d'interaction linéaire et quadratique par équation. Nous cherchons ensuite à approximer f par un modèle linéaire généralisé :

$$f(x) \approx \sum_{k=1}^p \theta_k(x) \xi_k = \Theta(x) \xi \quad (3.4)$$

en minimisant le nombre de termes non nuls dans ξ . Il est ensuite possible de résoudre les termes pertinents actifs dans la dynamique en utilisant une régression parcimonieuse [51, 60, 22, 24] qui pénalise le nombre de termes dans la dynamique et qui est adaptée aux problèmes de grande taille. Tout d'abord, des données de séries temporelles sont collectées à partir de (3.3) et sont organisées sous forme d'une matrice de données :

$$X = \begin{bmatrix} x(t_1) & x(t_2) & \dots & x(t_m) \end{bmatrix}^T \quad (3.5)$$

Une matrice similaire de dérivées est formée :

$$\dot{X} = \begin{bmatrix} \dot{x}(t_1) & \dot{x}(t_2) & \dots & \dot{x}(t_m) \end{bmatrix}^T \quad (3.6)$$

Dans la pratique, cela peut être calculé directement à partir des données dans X ; pour des données bruitées, la dérivée régularisée par variation totale a tendance à fournir des dérivées numériquement robustes [19]. Une autre approche consiste à formuler l'algorithme SINDy pour des systèmes en temps discret $x_{k+1} = F(x_k)$, similaire à l'algorithme DMD, ce qui permet d'éviter complètement le calcul des dérivées.

Une bibliothèque de fonctions non linéaires candidates $\theta(X)$ peut être construite à partir des données dans X :

$$\Theta(X) = \begin{bmatrix} 1 & X & X^2 & \dots & X^d & \dots & \sin(X) & \dots \end{bmatrix} \quad (3.7)$$

Ici, la matrice X^d représente une matrice dont les vecteurs colonnes sont formés par toutes les séries temporelles possibles de polynômes de degré d de l'état X . En général, cette bibliothèque de fonctions candidates est limitée uniquement par l'imagination de chacun. Le système dynamique dans (3.3) peut maintenant être représenté en termes des matrices de données dans (3.6) et (3.7) comme suit :

$$\dot{X} = \Theta(X)\Xi \quad (3.8)$$

Chaque colonne ξ_k dans la matrice Ξ est un vecteur de coefficients qui détermine les termes actifs dans la k^{eme} ligne de l'équation (3.3). Un modèle parcimonieux offrira un ajustement précis du modèle dans l'équation (3.8) avec le moins de termes possible dans la matrice Ξ . Un tel modèle peut être identifié en utilisant une régression parcimonieuse régularisée convexe avec une régularisation l1.

$$\xi_k = \underset{\xi'_k}{\operatorname{argmin}} \|\dot{X}_k - \Theta(X)\xi'_k\|_2 + \lambda \|\xi'_k\|_1 \quad (3.9)$$

Dans le contexte donné, \dot{X}_k représente la k -ième colonne de \dot{X} , tandis que λ est un paramètre qui favorise la parcimonie. Les techniques de régression parcimonieuse, telles que le LASSO [51] ou l'algorithme de moindres carrés seuillé séquentiel (STLS) utilisé dans SINDy [12], améliorent la robustesse numérique du processus d'identification, notamment pour les problèmes bruités et surdéterminés. Ces méthodes représentent une avancée par rapport aux approches antérieures [56] qui utilisaient des techniques de détection comprimée [21, 14, 16, 15, 17, 4, 53]. Il est conseillé d'utiliser l'algorithme STLS (*Sequentially Trimmed Least Squares*) (Code 3.1) pour sélectionner efficacement les termes actifs dans le modèle.

```

%% compute Sparse regression: sequential least squares
Xi = Theta\dXd; % initial guess: Least-squares

% lambda is our sparsification knob.
for k=1:10
    smallinds = (abs(Xi)<lambda); % find small coefficients
    Xi(smallinds)=0; % and threshold
    for ind = 1:n % n is state dimension
        biginds = ~smallinds(:,ind);
        % Regress dynamics onto remaining terms to find sparse Xi
        Xi(biginds,ind) = Theta(:,biginds)\dXd(:,ind);
    end
end
end

```

FIGURE 3.1 – Moindres carrés seuillés séquentiellement, de [12].

3.3.1 Exemple de compréhension : Équations de Lorenz

Le système de Lorenz est un ensemble d'équations différentielles ordinaires qui décrivent un modèle simplifié de la convection atmosphérique. Les équations de Lorenz sont généralement exprimées comme suit :

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(\rho - z) - y \\ \dot{z} = xy - \beta z \end{cases} \quad (3.10)$$

Ici, x , y et z représentent les variables d'état du système, tandis que σ , ρ et β sont des paramètres constants qui déterminent les caractéristiques du système.

La figure suivante (Fig. 3.2) va expliquer les différentes étapes d'application de l'algorithme de SINDy sur le système de Lorenz :

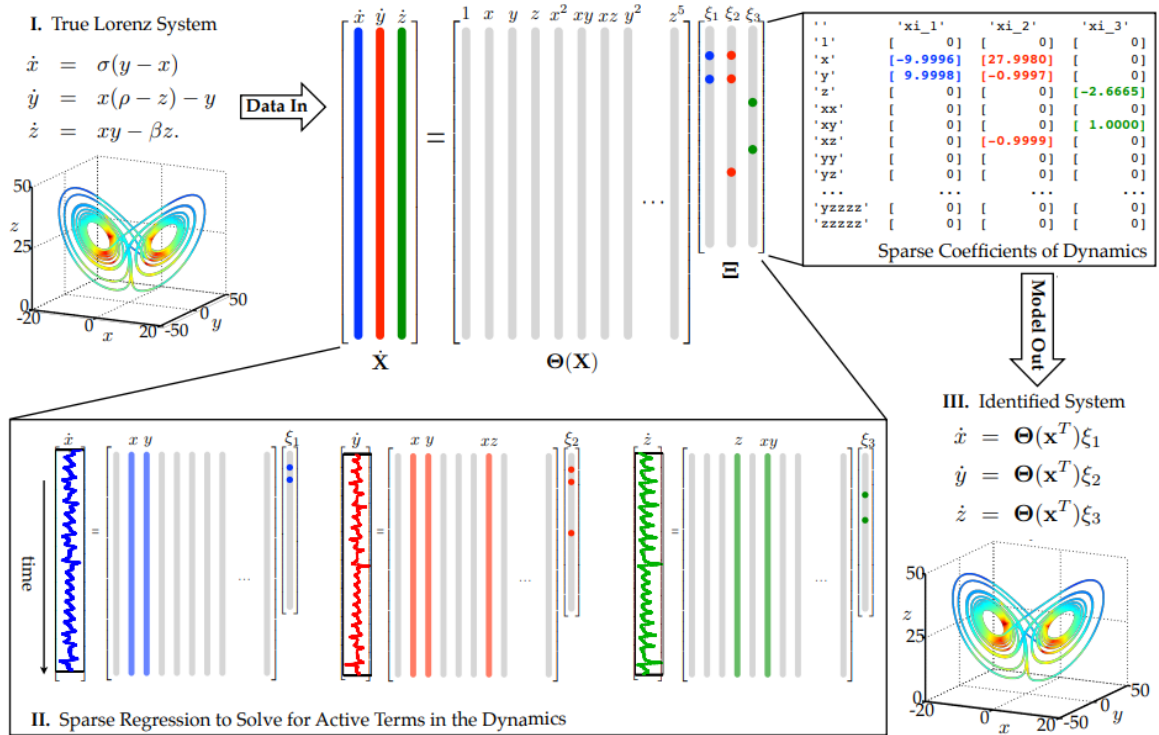


FIGURE 3.2 – Schéma de l'algorithme d'identification parcimonieuse de la dynamique non linéaire, démontré sur les équations de Lorenz. Reproduite de [12].

Les données sont collectées à partir des mesures du système, y compris l'historique temporel des états X et dérivés \dot{X} . Ensuite, une bibliothèque de fonctions non linéaires des états, $\Theta(X)$, est construite. Cette bibliothèque de fonctionnalités non linéaires est utilisée pour trouver le moins de termes nécessaires pour satisfaire $\dot{X} = \Theta(X)\Xi$. Les quelques entrées dans les vecteurs de Ξ , résolus par une régression parcimonieuse, indiquent les termes pertinents dans le côté droit de la dynamique. Les valeurs des paramètres sont $\sigma = 10, \beta = 8/3, \rho = 28, (x_0, y_0, z_0)^T = (-8, 7, 27)^T$. La trajectoire sur l'attracteur de Lorenz est colorée par l'étape de temps adaptative requise, avec le rouge exigeant un *tilmestep* plus petit [12].

Le système de Lorenz (3.10) en injectant les valeurs des paramètres devient :

$$\begin{cases} \dot{x} = 10y - 10x \\ \dot{y} = 28x - xz - y \\ \dot{z} = xy - (8/3)z \end{cases} \quad (3.11)$$

Le système identifié par SINDy est :

$$\begin{cases} \dot{x} = 9.9998y - 9.9996x \\ \dot{y} = 27.9980x - 0.9999xz - 0.9997y \\ \dot{z} = 1.0000xy - 2.6665z \end{cases} \quad (3.12)$$

3.3.2 Algorithme pour la représentation parcimonieuse de la dynamique avec du bruit :

Dans la plupart des cas réalistes, les données X et \dot{X} seront affectées par du bruit, ce qui fait que l'équation (3.8) ne sera pas vérifiée exactement. Dans le cas où X est relativement propre mais les dérivées \dot{X} sont bruitées, l'équation devient :

$$\dot{X} = \Theta(X)\Xi + \eta Z \quad (3.13)$$

Dans cette formulation, nous considérons une matrice Z avec des entrées gaussiennes indépendantes et identiquement distribuées de moyenne nulle, tandis que η représente l'amplitude du bruit. Ainsi, notre objectif est de trouver une solution parcimonieuse pour un système surdéterminé avec du bruit.

Le LASSO [22, 51] en statistiques est efficace pour ce type de données, car il permet d'obtenir une régression parcimonieuse. Cependant, il peut être coûteux en termes de calcul lorsqu'il est appliqué à de grands ensembles de données.

Une alternative consiste à utiliser l'algorithme des moindres carrés seuillés séquentiels présenté dans le code (3.1). Dans cet algorithme, nous commençons par trouver une solution des moindres carrés pour Ξ , puis nous appliquons un seuil à tous les coefficients inférieurs à une valeur de seuil λ . Une fois les indices des coefficients non nuls identifiés, nous obtenons une nouvelle solution des moindres carrés pour Ξ en ne considérant que ces indices. Les nouveaux coefficients sont ensuite soumis à un nouveau seuillage avec λ , et cette procédure est répétée jusqu'à ce que les coefficients non nuls convergent. Cet algorithme est efficace en termes de calcul et converge rapidement vers une solution parcimonieuse en un petit nombre d'itérations. De plus, il est simple à mettre en œuvre, avec un seul paramètre λ qui détermine le degré de parcimonie dans Ξ .

Selon le niveau de bruit, il peut être nécessaire de filtrer les données X et la dérivée \dot{X} avant de résoudre pour Ξ . En particulier, si seules les données X sont disponibles et que \dot{X} doit être obtenue par différenciation, la matrice dérivée résultante peut présenter un bruit important. Pour atténuer cela, nous utilisons la dérivée régularisée par la variation totale [19] pour débruiter la dérivée. Une autre option serait de filtrer les données X et \dot{X} .

3.4 Conclusion :

Dans ce chapitre, nous avons exploré en détail l'algorithme SINDy (Sparse Identification of Nonlinear Dynamics) et sa méthode d'identification parcimonieuse de la dynamique non linéaire. Nous avons commencé par mettre en contexte l'importance de la modélisation des dynamiques à partir de données, ainsi que les limites des méthodes d'identification de systèmes traditionnelles. Ces méthodes sont généralement basées sur des hypothèses concernant la forme des équations régissant ces systèmes.

Nous avons ensuite discuté de deux approches clés utilisées dans SINDy : la régression symbolique et la représentation parcimonieuse. La régression symbolique permet de trouver des fonctions non linéaires qui décrivent les relations entre les variables et les dérivées temporelles mesurées. La représentation parcimonieuse, quant à elle, vise à sélectionner les termes pertinents du modèle en imposant une contrainte de parcimonie, c'est-à-dire en favorisant un petit nombre de termes non nuls.

Ensuite, nous avons présenté l'algorithme SINDy et ses étapes clés. L'algorithme contourne la recherche combinatoire de toutes les structures de modèles possibles en exploitant le fait que de nombreux systèmes dynamiques ont une dynamique qui ne présente que quelques termes actifs dans l'espace des fonctions possibles. Ainsi, SINDy cherche à approximer la dynamique par un modèle linéaire généralisé en minimisant le nombre de termes non nuls.

Nous avons également discuté de l'application de SINDy aux équations de Lorenz, un système dynamique bien connu décrivant la convection atmosphérique. À travers cet exemple, nous avons illustré les différentes étapes de l'application de SINDy, de la collecte des données à l'identification des termes pertinents du modèle.

En outre, nous avons introduit dans ce chapitre un algorithme permettant de modéliser la dynamique non linéaire en présence de bruit. Cette extension de la méthode SINDy combine la recherche de la représentation parcimonieuse avec des techniques de réduction du bruit, ce qui permet d'obtenir des modèles plus robustes et précis. En prenant en compte le bruit dans les données, cette approche permet de mieux modéliser la complexité inhérente aux systèmes dynamiques non linéaires.

En conclusion, SINDy offre une approche puissante et efficace pour découvrir des modèles non linéaires à partir de données. En exploitant la régression symbolique et la représentation parcimonieuse, SINDy permet d'identifier les termes pertinents du modèle tout en maintenant la simplicité et l'interprétabilité. Cette approche présente de nombreuses applications potentielles dans divers domaines, tels que la physique, la biologie, l'économie et l'ingénierie, où la modélisation des dynamiques non linéaires à partir de données est essentielle.

CHAPITRE 4

APPLICATION DE LA MÉTHODE DE IDENTIFICATION PARCIMONIEUSE DE LA DYNAMIQUE NON LINÉAIRE 'SINDY'

4.1 Introduction :

Dans ce chapitre, nous allons appliquer la méthode pilotée par les données appelée "SINDy" (Sparse Identification of Nonlinear Dynamics) pour l'identification parcimonieuse de la dynamique non linéaire. Nous allons également évaluer l'efficacité de cette méthode dans l'identification des systèmes. Nous discuterons et analyserons les résultats obtenus, ainsi que les limites auxquelles nous avons été confrontés lors de l'évaluation de l'efficacité de cet algorithme pour trouver un modèle non linéaire, et non seulement pour l'estimation des paramètres du modèle déjà choisi. Pour cette section, nous utiliserons des ensembles de données provenant de différents systèmes étudiés à l'aide d'expériences virtuelles.

Remarque :

Dans les cas réalistes, les données recueillies sont souvent affectées par du bruit. Lorsque nous travaillons avec de telles données bruitées et effectuons des opérations de dérivation, les erreurs et les fluctuations se propagent et s'amplifient dans les dérivées. En conséquence, les dérivées peuvent être bruitées et ne pas refléter fidèlement le véritable comportement du système.

Pour simuler cette situation, nous ajoutons un bruit aux dérivées, reproduisant ainsi l'amplification du bruit lors de l'opération de dérivation. Cette approche vise à mieux modéliser les équations dynamiques dans des conditions réelles où les données sont bruitées. Il est important de noter que l'ajout de bruit supplémentaire aux dérivées est une simplification et peut ne pas correspondre précisément à la réalité du système. Cependant, il est crucial de choisir attentivement l'amplitude et le type de bruit ajouté pour qu'ils reflètent au mieux les caractéristiques réelles du bruit présent dans les données.

Dans les exemples qui suivent, nous ajoutons un bruit gaussien d'amplitude 0.05 aux dérivées lors de l'application de SINDy. Cela permet d'illustrer comment l'algorithme peut mieux s'adapter à des conditions bruitées et fournir une estimation plus robuste des équations dynamiques.

Le bruit est ajouté selon les équations suivantes :

$$dx = dx + v; \quad (4.1)$$

$$v = \text{eps} \times \text{randn}(\text{size}(dx)); \quad (4.2)$$

Où \mathbf{dx} représente la matrice des dérivées de données.

La fonction `randn(size(dx))` génère une matrice de nombres pseudo-aléatoires suivant une distribution normale (bruit gaussien), et son amplitude est définie par le paramètre `eps`, qui est fixé à 0.05.

4.2 Exemple 1 : Système linéaire

Dans cet exemple, nous avons étudié un système linéaire tridimensionnel, représenté par les équations suivantes :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -0.3 & 4 & 0 \\ -2 & -0.2 & 0 \\ 0 & 0 & -0.2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.3)$$

4.2.1 Résultat de simulation de système original (Prenant une condition initiale $x_0 = [2; 0; 1]$) :

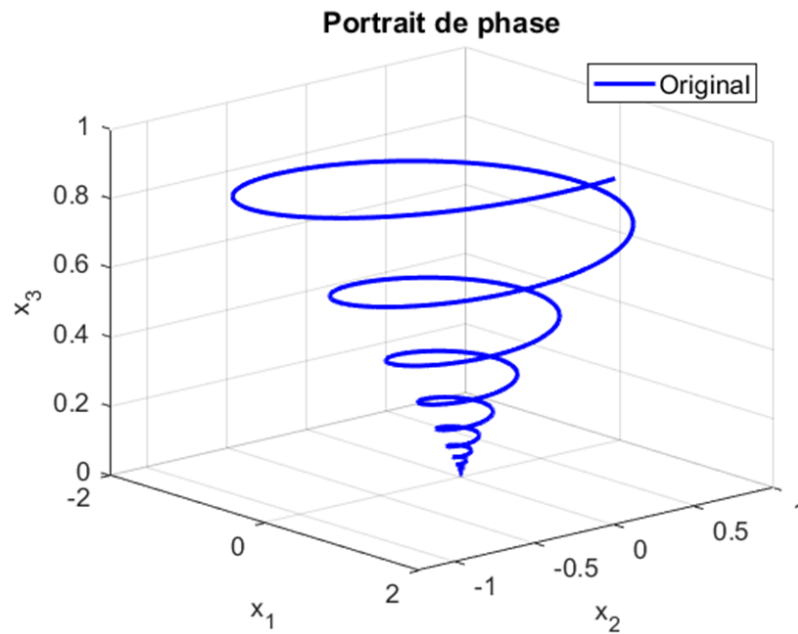
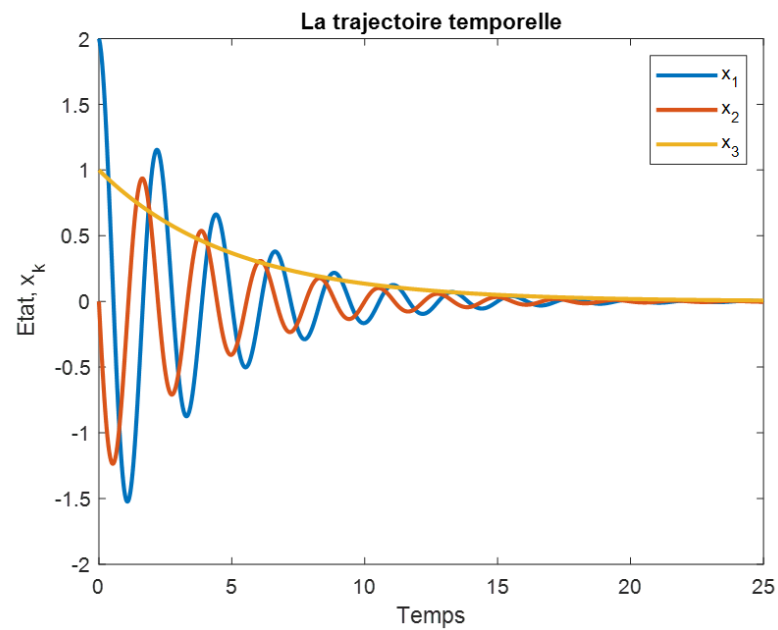


FIGURE 4.1 – Le portrait de phase du système étudié

FIGURE 4.2 – L'évolution temporelle du système étudié X_{data} .

4.2.2 Résultat de l'identification :

La bibliothèque des fonctions non linéaires candidates $\theta(X)$ utilisée est :

$$\Theta(X) = \begin{bmatrix} 1 & x & y & z & xx & xy & xz & yy & yz & zz \end{bmatrix} \quad (4.4)$$

L'utilisation de la méthode SINDy avec la bibliothèque Θ (4.4), en ajoutant un bruit gaussien à la dérivée des états \dot{X} comme indiqué dans les équations (4.1) et (4.2), et en utilisant la méthode d'optimisation des moindres carrés seuillés séquentiellement, nous permet d'approximer le système original (4.3) par le système linéaire suivant :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -0.3008 & 4.0004 & 0 \\ -2.0002 & -0.1967 & 0 \\ 0 & 0 & -0.1969 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.5)$$

L'évolution du modèle dynamique résultant :

Nous pouvons observer le résultat de l'évolution du modèle dynamique résultant de l'application de SINDy en comparant ce dernier, noté X_{sindy} , avec celui du système original, noté X_{data} (voir Fig. 4.2), en les affichant sur la même figure :

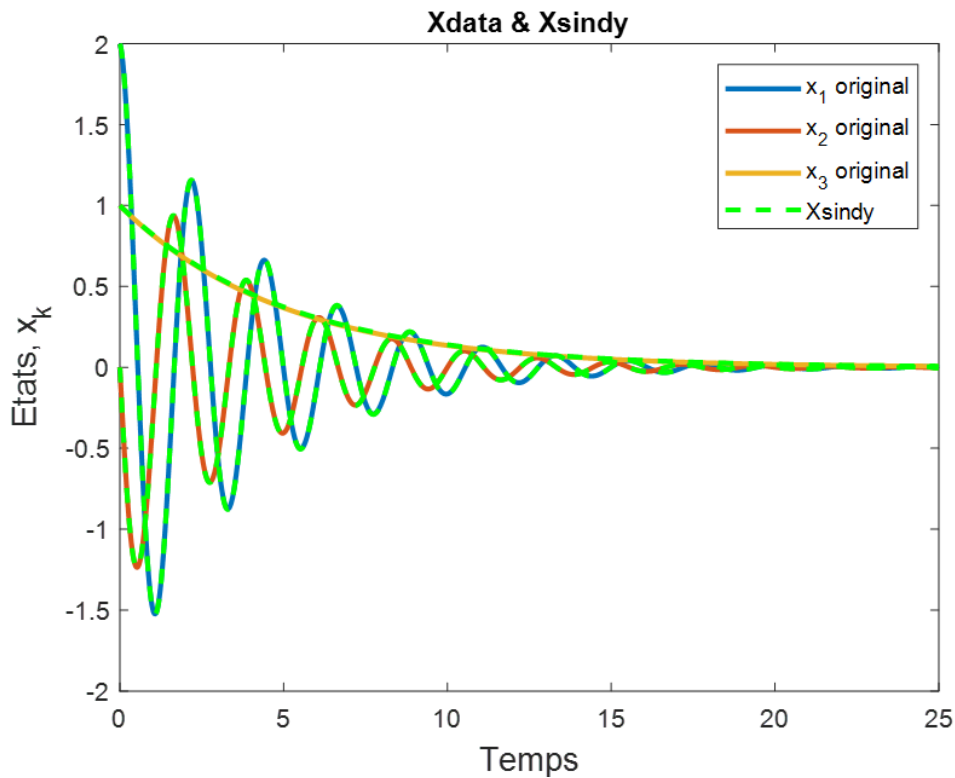


FIGURE 4.3 – Comparaison des données reconstruites par la méthode SINDy X_{sindy} (en vert) avec les données du système étudié X_{data} (en bleu, marron et orange).

Erreur de reconstruction des états par SINDy :

Nous remarquons que SINDy reconstruit parfaitement les états du système étudié avec une faible erreur définie comme $err = X_{data} - X_{sindy}$. Cette observation est illustrée dans la figure suivante :

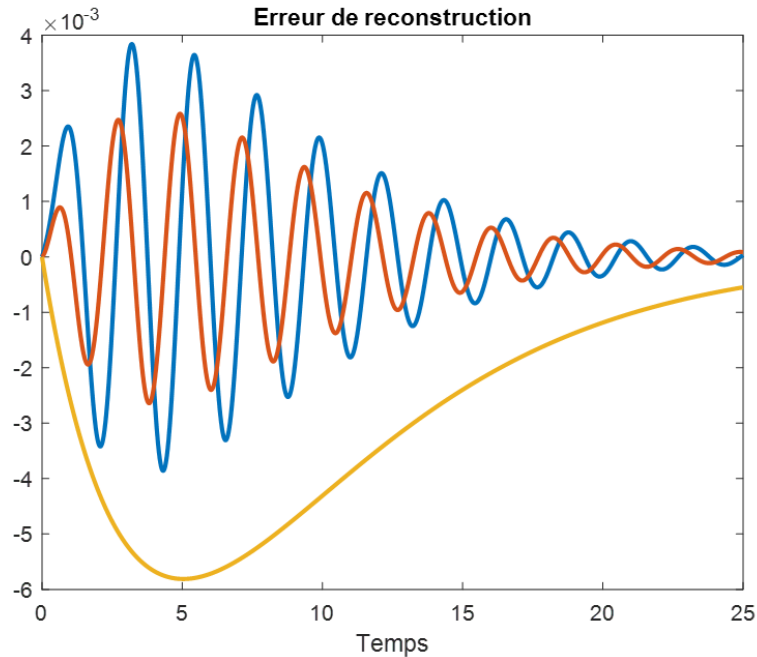


FIGURE 4.4 – Erreur de reconstruction des états par SINDy : Différence entre les données reconstruites et les données originales.

4.2.3 Interprétation :

Dans cet exemple, la méthode SINDy parvient à obtenir une forme linéaire précise avec une précision de 10^{-3} , grâce à la linéarité initiale du système. Malgré la présence de termes non linéaires dans la bibliothèque augmentée Θ (4.4), la méthode SINDy sélectionne uniquement les termes linéaires pour identifier le système linéaire.

4.3 Exemple 2 : Équation de Duffing

Le système d'oscillation de Duffing est un exemple simple souvent utilisé pour illustrer l'application de la théorie d'identification parcimonieuse de la dynamique non linéaire 'SINDy'. L'équation de Duffing est représenté dans l'espace d'état par $\ddot{x} = x - x^3$ [10].

Représentation d'état :

L'équation de Duffing peut être représentée dans l'espace d'état par :

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1 - x_1^3 \end{cases} \quad (4.6)$$

Où x_1 représente la position de l'oscillateur et x_2 sa vitesse. Cette représentation permet de décrire l'évolution du système au cours du temps en fonction de ses états précédents et des entrées externes.

4.3.1 Résultat de simulation de système original : (Prenant une condition initiale $x_0 = [2; -2]$)

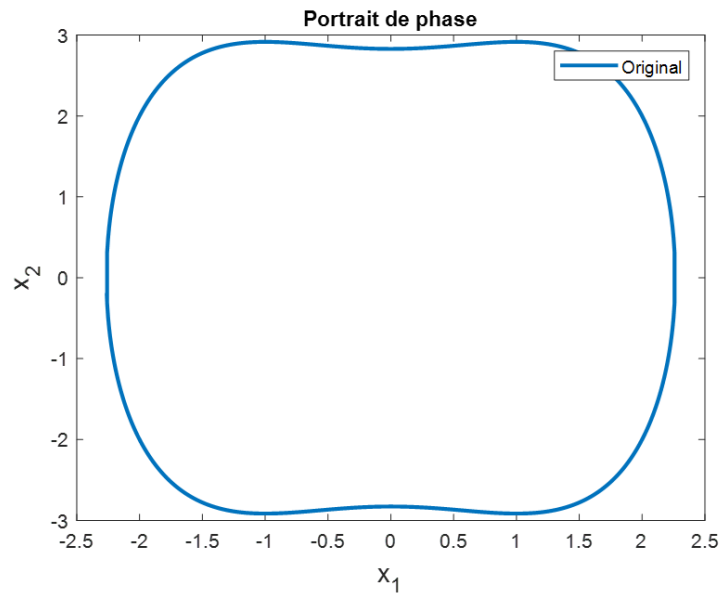
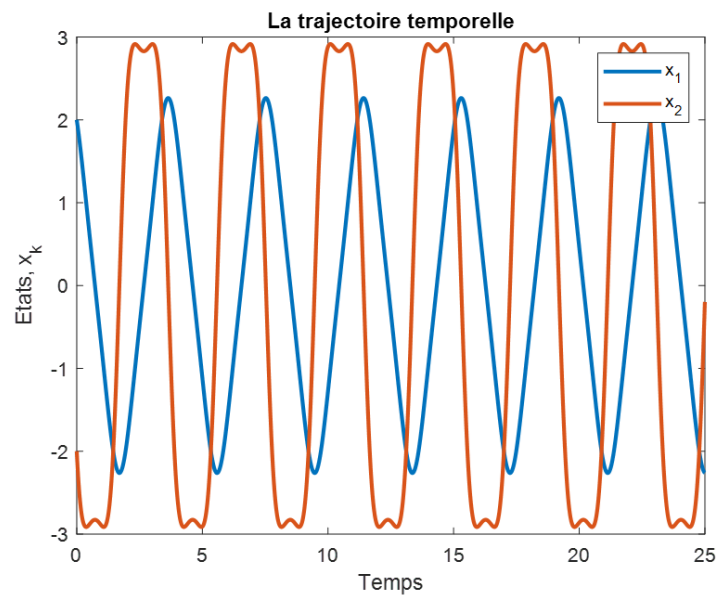


FIGURE 4.5 – Le portrait de phase du système étudié

FIGURE 4.6 – L'évolution temporelle du système étudié X_{data} .

4.3.2 Résultat de l'identification :

La bibliothèque des fonctions non linéaires candidates $\theta(X)$ utilisée est :

$$\Theta(X) = [1 \ X_1 \ X_2 \ X_1^2 \ X_1X_2 \ X_2^2 \ X_1^3 \ X_1^2X_2 \ X_1X_2^2 \ X_2^3 \ X_1^4 \ X_1^3X_2 \ X_1^2X_2^2 \ X_1X_2^3 \ X_2^4] \quad (4.7)$$

L'utilisation de la méthode SINDy avec la bibliothèque Θ (4.7), en ajoutant un bruit gaussien à la dérivée des états \dot{X} comme indiqué dans les équations (4.1) et (4.2), et en appliquant la méthode d'optimisation des moindres carrés seuillés séquentiellement, nous permet d'approximer le système original (4.6) par le système non linéaire suivant :

$$\begin{cases} \dot{x}_1 &= 1.0001x_2 \\ \dot{x}_2 &= 1.0009x_1 - 1.0002x_1^3 \end{cases} \quad (4.8)$$

L'évolution du modèle dynamique résultant :

Nous pouvons observer le résultat de l'évolution du modèle dynamique résultant de l'application de SINDy en comparant ce dernier, noté X_{sindy} , avec celui du système original, noté X_{data} (Voir Fig. 4.6), en les affichant sur la même figure :

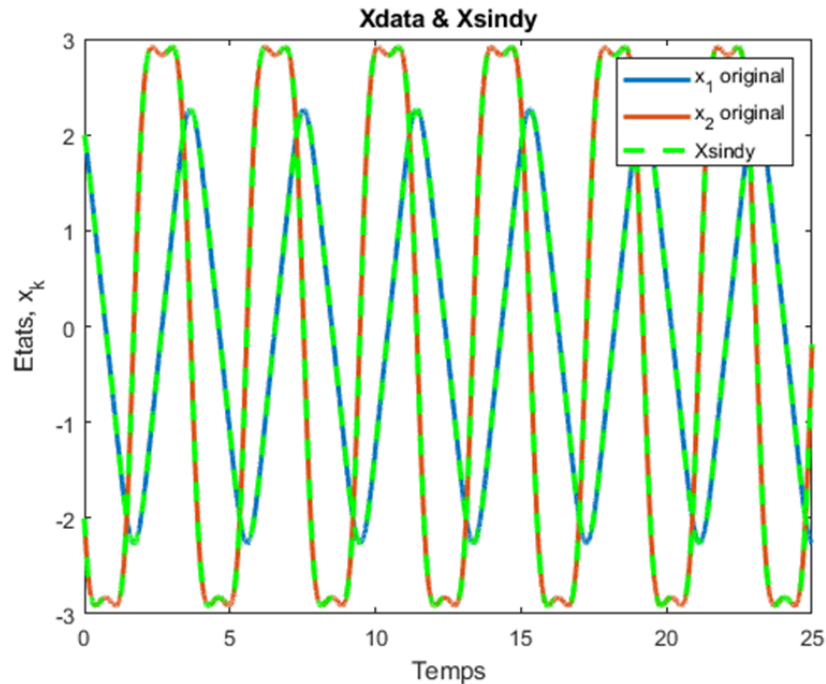


FIGURE 4.7 – Comparaison des données reconstruites par la sindy X_{sindy} (en vert) avec les données du système étudié X_{data} (en bleu et marron).

Erreur de reconstruction des états par SINDy :

Nous remarquons que SINDy reconstruit parfaitement les états du système étudié avec une faible erreur définie comme $err = X_{data} - X_{sindy}$. Cette observation est illustrée dans la figure suivante :

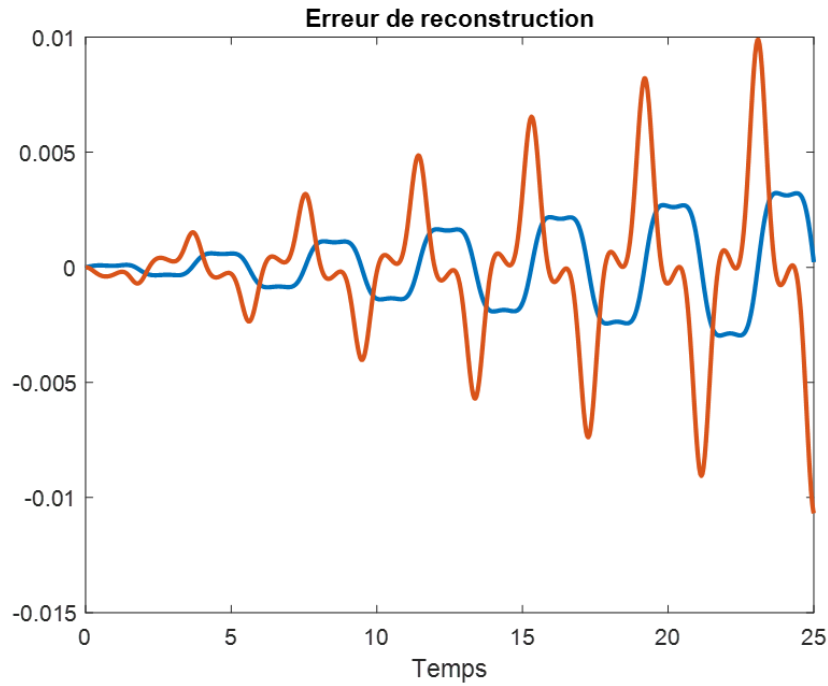


FIGURE 4.8 – Erreur de reconstruction des états par SINDy : Différence entre les données reconstruites et les données originales.

4.3.3 Interprétation :

Dans l'exemple du système de Duffing, la bibliothèque non linéaire augmentée Θ (4.7) comprend des polynômes en X jusqu'à l'ordre quatre. Malgré cela, on peut observer que la forme correcte de la non-linéarité est obtenue, puisque la non-linéarité était initialement polynomiale. Cela est clairement mis en évidence lors de la comparaison entre le modèle non linéaire résultant de SINDy (4.8) et le modèle du système dynamique original (4.6).

4.4 Exemple 3 : Pendule simple

4.4.1 Description du système :

Le pendule simple est un système physique constitué d'un objet de masse \mathbf{m} suspendu à une corde de longueur \mathbf{l} , qui oscille librement sous l'effet de la gravité. C'est l'un des systèmes les plus simples pour étudier les oscillations.

la dynamique d'un pendule simple avec une entrée de contrôle peut être décrite par l'équation différentielle suivante :

$$\ddot{\theta} + (k/ml^2)\dot{\theta} + (g/l)\sin(\theta) = (1/ml^2)u \quad (4.9)$$

Où :

- θ est l'angle (la position angulaire) que la corde du pendule fait avec la verticale.
- $\dot{\theta}$ représente la dérivée de θ par rapport au temps, c'est-à-dire la vitesse angulaire du pendule.
- $\ddot{\theta}$ représente l'accélération angulaire.
- \mathbf{g} est l'accélération due à la gravité (accélération gravitationnelle).
- \mathbf{k} Coefficient de frottement visqueux.
- \mathbf{l} est la longueur de la corde (la tige).
- \mathbf{m} est la masse du pendule (l'objet).
- \mathbf{u} est l'entrée de contrôle, cette variable de commande peut être utilisée pour modifier le comportement du système.

L'équation du pendule simple est une équation non linéaire en raison de la présence du terme $\sin(\theta)$. Cependant, pour de petites oscillations où θ est proche de zéro, on peut approximer $\sin(\theta)$ par θ en utilisant la formule de l'angle petit.

Ainsi, pour de petites oscillations, l'équation du pendule simple linéaire devient :

$$\ddot{\theta} + (k/ml^2)\dot{\theta} + (g/l)\theta = (1/ml^2)u \quad (4.10)$$

4.4.2 La représentation d'état :

La représentation d'état de ce système peut être obtenue en introduisant deux variables d'état, $x_1 = \theta$ et $x_2 = \dot{\theta}$, et une variable de commande, u .

Donc les équations d'état (*les dynamiques*) de pendule simple avec entrée de contrôle sont comme suit :

$$\dot{x} = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -(g/l)\sin(x_1) - (k/ml^2)x_2 + (1/ml^2)u \end{cases} \quad (4.11)$$

Sous la forme d'un système non linéaire affine en commande :

$$\dot{x} = f(x) + g(x)u \quad (4.12)$$

La représentation donnée permet de modéliser, simuler et contrôler le comportement d'un pendule en tenant compte de la gravité, de l'amortissement et d'une entrée de contrôle externe u . Ce système présente des comportements non linéaires.

4.4.3 Résultats de simulation de système original (Prenant une condition initiale $x_0 = [2; 2]$) :

Les valeurs des paramètres utilisés dans cet exemple sont données comme suit :

- Longueur du pendule : $l = 0.5 \text{ m}$
- Masse du pendule : $m = 0.1 \text{ Kg}$
- Accélération gravitationnelle : $g = 9.8 \text{ m.s}^{-2}$
- Coefficient de frottement visqueux : $k = 25\text{e-}3 \text{ Kg.s}^{-1}$.

Pour simuler notre système, nous allons utiliser une commande par retour d'état de la forme $u = -k_1x_1 - k_2x_2$, où nous choisissons les valeurs de $k_1 = 0.3333$ et $k_2 = 0.6666$ pour stabiliser le système. Cela nous permettra de ramener le système (4.11) à la forme suivante :

$$\dot{x} = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -(g/l)\sin(x_1) - (k/ml^2)x_2 + (1/ml^2)(-k_1x_1 - k_2x_2) \end{cases} \quad (4.13)$$

Nous allons identifier le système en boucle fermée.

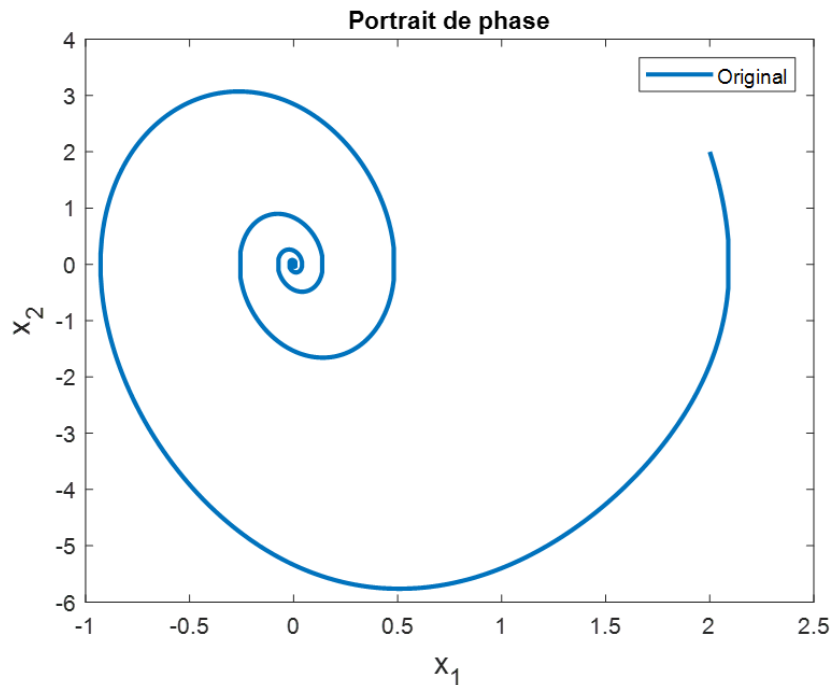


FIGURE 4.9 – Le portrait de phase du système étudié.

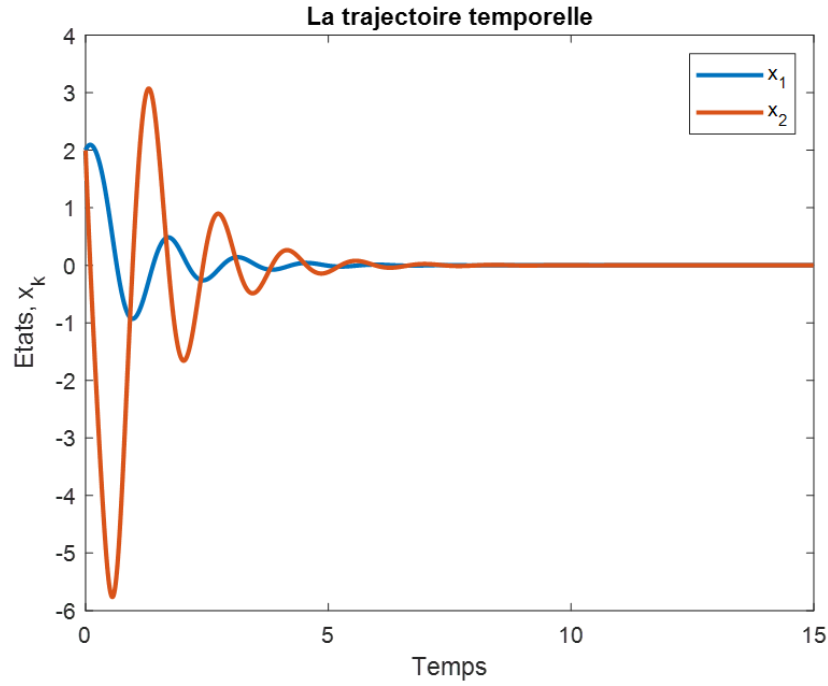


FIGURE 4.10 – L'évolution temporelle du système étudié Xdata.

4.4.4 Résultat de l'identification :

La bibliothèque utilise de fonctions non linéaires candidates $\theta(X)$:

$$\Theta(X) = [1 \ X_1 \ X_2 \ X_1^2 \ X_1 X_2 \ X_2^2 \ X_1^3 \ X_1^2 X_2 \ X_1 X_2^2 \ X_2^3 \ X_1^4 \ X_1^3 X_2 \ X_1^2 X_2^2 \ X_1 X_2^3 \ X_2^4 \ X_1^5 \ X_1^4 X_2 \ X_1^3 X_2^2 \ X_1^2 X_2^3 \ X_1 X_2^4 \ X_2^5] \quad (4.14)$$

L'utilisation de la méthode SINDy avec la bibliothèque Θ (4.14), en ajoutant un bruit gaussien à la dérivée des états \dot{X} comme indiqué dans les équations (4.1) et (4.2), et en appliquant la méthode d'optimisation des moindres carrés seuillés séquentiellement, nous permet d'approximer le système original bouclé par le retour d'état appliqué (4.13) par le système non linéaire suivant :

$$\begin{cases} \dot{x}_1 &= 0.9999x_2 \\ \dot{x}_2 &= -1.729x_2 - 20.4137x_1 + 3.2183x_1^3 - 0.1374x_1^5 \end{cases} \quad (4.15)$$

4.4.5 L'évolution du modèle dynamique résultant :

Nous pouvons observer le résultat de l'évolution du modèle dynamique résultant de l'application de SINDy en comparant ce dernier, noté X_{sindy} , avec celui du système original, noté X_{data} (voir Fig. 4.10), en les affichant sur la même figure :

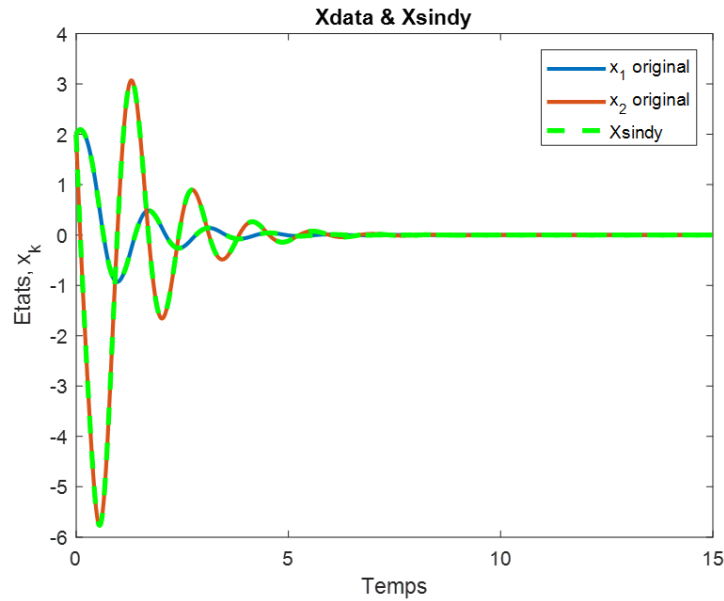


FIGURE 4.11 – Comparaison des données reconstruites par la sindy X_{sindy} (en vert) avec les données du système étudié X_{data} (en bleu et marron).

4.4.6 Erreur de reconstruction des états par SINDy :

Nous remarquons que SINDy reconstruit parfaitement les états du système étudié avec une faible erreur définie comme $err = X_{data} - X_{sindy}$. Cette observation est illustrée dans la figure suivante :

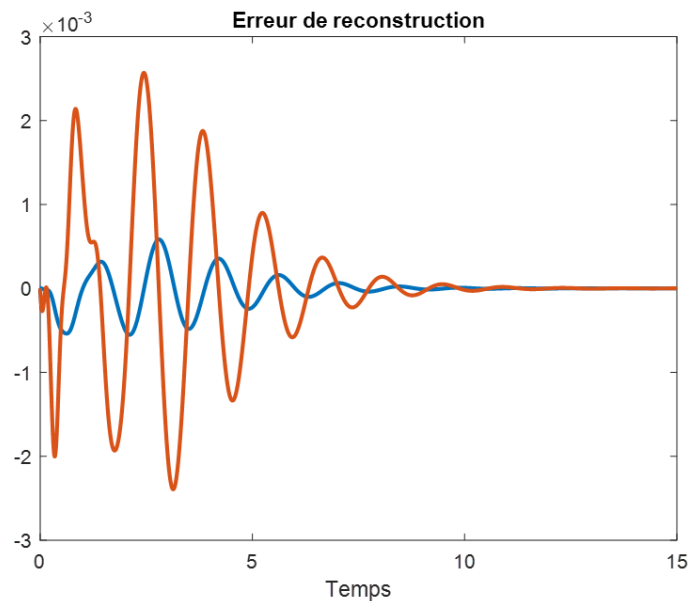


FIGURE 4.12 – Erreur de reconstruction des états par SINDy : Différence entre les données reconstruites et les données originales.

4.4.7 Interprétation :

L'application de la méthode SINDy dans le système de pendule simple a conduit à la construction d'un modèle non linéaire différent de celui du système original, mais qui présente une évolution similaire avec une erreur d'environ 10^{-3} . Cette disparité est attribuée à l'absence directe de la fonction sinus dans la bibliothèque de fonctions candidates.

Cependant, les fonctions candidates disponibles dans la bibliothèque sont suffisamment riches pour extraire le développement limité de la fonction sinus. Ainsi, SINDy peut utiliser les autres fonctions candidates pour approximer le sinus en utilisant son développement limité ($\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$).

L'efficacité de la méthode SINDy dépend de la précision de cette approximation ainsi que du nombre de termes du développement limité pris en compte dans la bibliothèque. En ajustant correctement la bibliothèque des fonctions candidates, il est possible d'améliorer la précision de l'approximation du système non linéaire, réduisant ainsi l'erreur par rapport au système original.

4.5 Exemple 4 : Équation de Van der pol (oscillateur non forcé)

Le système de Van der Pol est un exemple de système dynamique non linéaire qui décrit les oscillations auto-entretenues. Il a été introduit par le physicien néerlandais Balthasar van der Pol en 1927 [55]. L'équation différentielle de Van der Pol est présentée comme suit :

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0, \quad (4.16)$$

Où x est la position et μ est un paramètre qui détermine la force de non-linéarité.

Le système de Van der Pol peut être utilisé pour modéliser de nombreux phénomènes physiques, tels que les oscillations électriques dans les circuits électroniques, les vibrations dans les systèmes mécaniques et les oscillations cardiaques. Ce système possède des propriétés intéressantes, notamment la capacité à générer des oscillations périodiques complexes et à entrer en résonance avec d'autres systèmes.

Représentation d'état :

Pour le système de Van der Pol, la représentation d'état peut être obtenue en introduisant deux variables d'état, x_1 et x_2 . Donc les équations d'état pour le système de Van der Pol sont comme suit :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \mu(1 - x_1^2)x_2 - x_1 \end{cases} \quad (4.17)$$

Où x_1 représente la position et x_2 représente la vitesse, le vecteur d'état est : $x := [x_1, x_2]^T$.

4.5.1 Résultats de simulation de système original (Prenant $\mu = 1$ et une condition initiale $x_0 = [2; 3]$) :

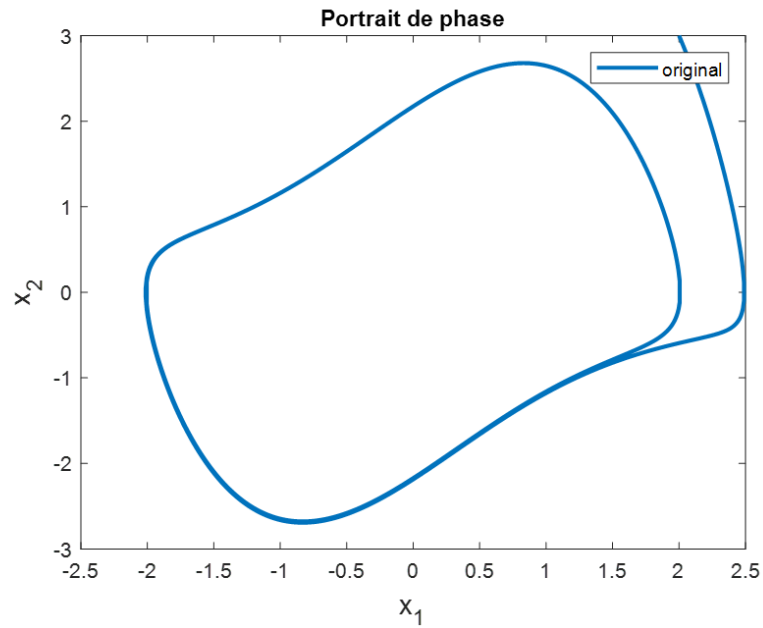


FIGURE 4.13 – Le portrait de phase du système original de Van Der Pol.

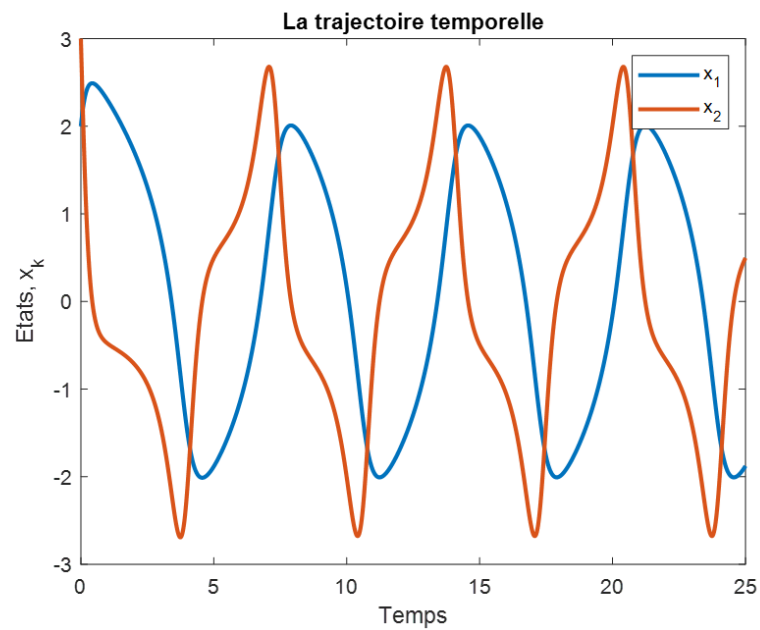


FIGURE 4.14 – L'évolution du système original de Van Der Pol

4.5.2 Résultat de l'identification :

La bibliothèque des fonctions non linéaires candidates $\theta(X)$ utilisée est :

$$\Theta(X) = [1 \ X_1 \ X_2 \ X_1^2 \ X_1X_2 \ X_2^2 \ X_1^3 \ X_1^2X_2 \ X_1X_2^2 \ X_2^3 \ X_1^4 \ X_1^3X_2 \ X_1^2X_2^2 \ X_1X_2^3 \ X_2^4] \quad (4.18)$$

L'utilisation de la méthode SINDy avec la bibliothèque Θ (4.18), en ajoutant un bruit gaussien à la dérivée des états \dot{X} comme indiqué dans les équations (4.1) et (4.2), et en appliquant la méthode d'optimisation des moindres carrés seuillés séquentiellement, nous permet d'approximer le système original (4.17) par le système non linéaire suivant :

$$\begin{cases} \dot{x}_1 = 1.0004x_2 \\ \dot{x}_2 = -1.0003x_1 + 0.9998x_2 - 0.9999x_1^2x_2 \end{cases} \quad (4.19)$$

4.5.3 L'évolution du modèle dynamique résultant :

Nous pouvons observer le résultat de l'évolution du modèle dynamique résultant de l'application de SINDy en comparant ce dernier, noté X_{sindy} , avec celui du système original, noté X_{data} (voir Fig. 4.14), en les affichant sur la même figure :

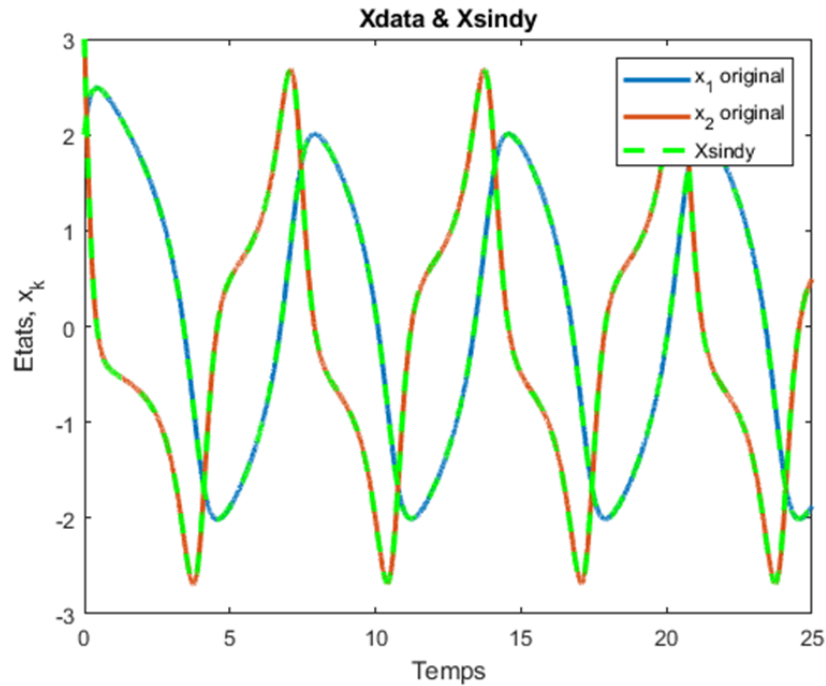


FIGURE 4.15 – Comparaison des données reconstruites par la sindy X_{sindy} (en vert) avec les données du système étudié X_{data} (en bleu et marron).

4.5.4 Erreur de reconstruction des états par SINDy :

Nous remarquons que SINDy reconstruit parfaitement les états du système étudié avec une faible erreur définie comme $err = X_{data} - X_{sindy}$. Cette observation est illustrée dans la figure suivante :

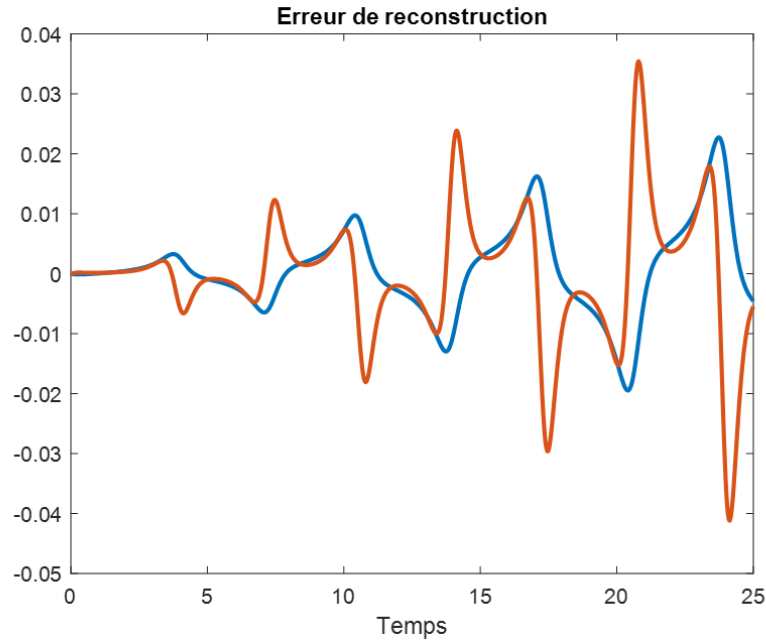


FIGURE 4.16 – Erreur de reconstruction des états par SINDy avec bruit : Différence entre les données reconstruites et les données originales.

4.5.5 Interprétation des résultats :

Dans l'exemple du système de Van der Pol, la bibliothèque non linéaire augmentée Θ (4.18) comprend des polynômes en X jusqu'à l'ordre quatre. Malgré cela, on peut observer que la forme correcte de la non-linéarité est obtenue, puisque la non-linéarité était initialement polynomiale. Cela est clairement mis en évidence lors de la comparaison entre le modèle non linéaire résultant de SINDy (4.19) et le modèle du système dynamique original (4.17).

4.5.6 Élimination du bruit :

Dans cette partie de l'exemple du Van der Pol, nous avons supprimé l'ajout de bruit à la dérivée de l'état \dot{X} afin d'étudier l'impact du bruit sur la méthode SINDy. Le système approximatif obtenu est donné par :

$$\begin{cases} \dot{x}_1 &= 1x_2 \\ \dot{x}_2 &= -1x_1 + 1x_2 - 1x_1^2x_2 \end{cases} \quad (4.20)$$

Dans ce cas, l'erreur de reconstruction diminue considérablement et atteint un niveau de l'ordre de 10^{-14} , comme le montre la figure (Fig. 4.17). Cette valeur est nettement plus faible que dans le cas précédent où du bruit était présent. En comparant le modèle obtenu (4.20) avec le système non linéaire original (4.17), on peut constater une identification exacte. Cela confirme que l'absence de bruit permet d'obtenir une reconstruction précise du système, ce qui est important car la méthode SINDy est sensible au bruit.

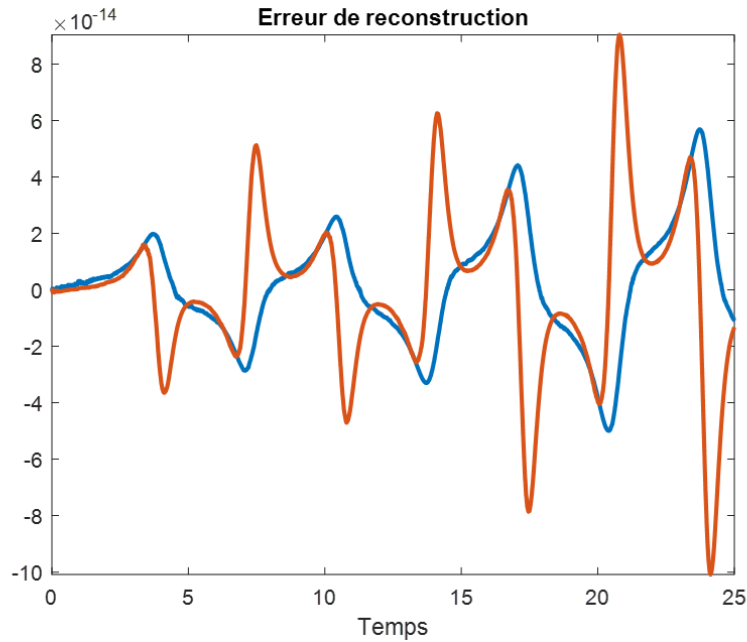


FIGURE 4.17 – Erreur de reconstruction des états par SINDy sans bruit : Différence entre les données reconstruites et les données originales non bruitées.

4.6 Conclusion :

Dans ce chapitre consacré à l'application de la méthode SINDy, nous avons utilisé cette approche sur un système linéaire ainsi que sur trois systèmes dynamiques non linéaires spécifiques : le système de Duffing, le système de pendule simple et le système de Van der Pol.

Tout d'abord, dans l'exemple du système linéaire, la méthode SINDy parvient à obtenir une reconstruction précise sous une forme linéaire, avec une précision élevée. Malgré la présence de termes non linéaires dans la bibliothèque augmentée, la méthode SINDy sélectionne uniquement les termes linéaires pour identifier le système linéaire. Cela démontre l'efficacité de la méthode pour capturer la linéarité intrinsèque des systèmes.

Ensuite, dans les exemples de systèmes non linéaires, la méthode SINDy réussit à reconstruire avec succès la forme correcte des non-linéarités, même si la bibliothèque utilisée ne contient que des termes polynomiaux jusqu'à un certain ordre. Cela est clairement illustré par la comparaison entre le modèle non linéaire résultant de SINDy et le modèle du système dynamique original. La méthode parvient à extraire les non-linéarités et à capturer les comportements non linéaires du système, confirmant ainsi sa capacité à identifier les systèmes non linéaires.

De plus, l'impact du bruit sur la méthode SINDy a été étudié. Il a été observé que l'ajout de bruit dans les données peut affecter la précision de la reconstruction du système. L'erreur de reconstruction augmente en présence de bruit, soulignant ainsi la sensibilité de la méthode SINDy au bruit. Cependant, en éliminant le bruit des données, la méthode parvient à reconstruire le système avec une précision considérablement améliorée, réduisant l'erreur de reconstruction à des niveaux très faibles. Cela met en évidence l'importance de minimiser le bruit pour obtenir des résultats précis avec la méthode SINDy.

En conclusion, la méthode SINDy se révèle être une approche puissante pour l'identification de systèmes, qu'ils soient linéaires ou non linéaires. Elle permet de reconstruire avec précision la forme du système, en capturant les comportements linéaires ou non linéaires, selon le cas. Toutefois, il est essentiel de prendre en compte le bruit dans les données et de sélectionner une bibliothèque de fonctions candidates appropriée afin d'obtenir des résultats fiables et précis.

CONCLUSION GÉNÉRALE

Dans notre projet de fin d'étude, nous nous sommes concentrés sur la méthode SINDy (Sparse Identification of Nonlinear Dynamics) et son application dans l'analyse des systèmes dynamiques à partir de données. Nous avons exploré en profondeur les principes fondamentaux de SINDy, son algorithme et ses applications dans l'identification de systèmes dynamiques de grande dimension. Nous avons également brièvement abordé les méthodes complémentaires de SINDy, à savoir l'opérateur de Koopman et la décomposition en mode dynamique (DMD).

Dans notre premier chapitre, nous avons présenté un aperçu général de SINDy, mettant en évidence son utilisation pour découvrir les équations sous-jacentes régissant le système, ainsi que son historique. Nous avons également souligné les limites inhérentes à cette méthode.

Notre deuxième chapitre est consacré à la compréhension de l'opérateur de Koopman et de la décomposition en mode dynamique (DMD), en mettant en avant leur complémentarité avec SINDy dans l'analyse des données et la compréhension des systèmes. Nous avons exploré l'algorithme de la DMD ainsi que ses différentes variantes, telles que la DMD avec contrôle (DMDc) et l'EDMD, en les reliant à la découverte de l'opérateur de Koopman à partir des données de l'espace d'état. Nous avons souligné l'efficacité de la DMD pour approximer l'opérateur de Koopman, notamment dans le cas des systèmes dynamiques non linéaires.

Le troisième chapitre a été consacré à l'identification parcimonieuse de la dynamique non linéaire avec SINDy. Nous avons examiné en détail l'algorithme SINDy, en mettant l'accent sur l'utilisation de la régression symbolique et de la représentation parcimonieuse. Nous avons expliqué comment SINDy approxime la dynamique des systèmes par un modèle non linéaire généralisé tout en recherchant la parcimonie, c'est-à-dire en minimisant le nombre de termes non nuls dans le modèle. Cette approche permet d'obtenir des modèles simplifiés et interprétables, en identifiant les termes les plus significatifs pour décrire la dynamique du système. Nous avons également présenté une extension de SINDy qui intègre la prise en compte du bruit dans les données, permettant ainsi de mieux modéliser les systèmes dynamiques non linéaires complexes.

Enfin, dans notre quatrième chapitre, nous avons appliqué la méthode SINDy à un système linéaire et à trois systèmes dynamiques non linéaires spécifiques : le système de Duffing, le système de Van der Pol et le système de pendule simple. Nous avons démontré l'efficacité de SINDy pour identifier et modéliser les dynamiques non linéaires de ces systèmes, même en présence de bruit ou d'incertitudes dans les données. Cependant, nous avons souligné que l'élimination du bruit permet d'obtenir des résultats beaucoup plus précis et une erreur très faible. Ces résultats ont mis en évidence l'importance du choix des fonctions candidates dans le processus d'identification.

En résumé, notre projet de fin d'étude a approfondi la méthode SINDy et son utilisation dans l'analyse des systèmes dynamiques. Nous avons exploré ses liens avec l'opérateur de Koopman et la DMD, ainsi que son application pratique à travers l'identification et la modélisation de systèmes dynamiques spécifiques. Ces travaux ouvrent la voie à de nombreuses possibilités de recherche future dans le domaine de l'analyse des systèmes dynamiques à partir de données.

BIBLIOGRAPHIE

- [1] Ralph Abraham, Jerrold E. Marsden, and Tudor Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75 of *Applied Mathematical Sciences*. Springer-Verlag, 1988.
- [2] Z Bai, E Kaiser, JL Proctor, JN Kutz, and SL Brunton. Dynamic mode decomposition for compressive system identification. *arXiv preprint arXiv :1710.07737*, 2017.
- [3] Maciej J. Balajewicz, Earl H. Dowell, and Bernd R. Noack. Low-dimensional modelling of high-reynolds-number shear flows incorporating constraints from the navier–stokes equation. *Journal of Fluid Mechanics*, 729 :285–308, 2013.
- [4] Richard G. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4) :118–120, 2007.
- [5] Stephen A. Billings. *Nonlinear System Identification : NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. John Wiley & Sons, 2013.
- [6] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24) :9943–9948, 2007.
- [7] Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, Eurika Kaiser, and J. Nathan Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8(1), May 2017.
- [8] Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLOS ONE*, 11(2) :e0150171, Feb 2016.
- [9] Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2) :e0150171, 2016.
- [10] Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern koopman theory for dynamical systems. *arXiv preprint arXiv :2102.12086*, 2021.
- [11] Steven L Brunton and J Nathan Kutz. *Data-Driven Science and Engineering : Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.
- [12] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. *Proceedings of the National Academy of Sciences*, 113(15) :3932–3937, Mar 2016.
- [13] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Sparse identification of nonlinear dynamics with control (sindyc). In *IFAC NOLCOS*, volume 49, pages 710–715, 2016.
- [14] Emmanuel J Candès. Compressive sensing. In *Proceedings of the International Congress of Mathematics*, 2006.

-
- [15] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles : exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2) :489–509, 2006.
- [16] Emmanuel J Candès, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications in Pure and Applied Mathematics*, 59(8) :1207–1223, 2006.
- [17] Emmanuel J Candès and Terence Tao. Near optimal signal recovery from random projections : Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12) :5406–5425, 2006.
- [18] Kevin Carlberg, Matthew Barone, and Harbir Antil. Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330 :693–734, 2017.
- [19] Rick Chartrand. Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, 2011.
- [20] Magnus Dam, Morten Brøns, Jens Juul Rasmussen, Volker Naulin, and Jan S. Hesthaven. Sparse identification of a predator-prey system from simulation data of a convection model. *Physics of Plasmas*, 24(2) :022310, 2017.
- [21] David L Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4) :1289–1306, 2006.
- [22] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The Elements of Statistical Learning*, volume 2. Springer, 2009.
- [23] Philip Holmes and John Guckenheimer. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42 of *Applied Mathematical Sciences*. Springer-Verlag, Berlin, 1983.
- [24] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer, 2013.
- [25] Eurika Kaiser, J. Nathan Kutz, and Steven L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *arXiv preprint*, 2017.
- [26] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of koopman eigenfunctions for control. *Machine Learning : Science and Technology*, 2(3) :035023, 2021.
- [27] Milan Korda and Igor Mezic. On convergence of extended dynamic mode decomposition to the koopman operator. *Journal of Nonlinear Science*, 28(2) :687–710, 2018.
- [28] Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems : Koopman operator meets model predictive control. *Automatica*, 93 :149–160, Jul 2018.
- [29] John R Koza. *Genetic programming : on the programming of computers by means of natural selection*, volume 1. MIT Press, 1992.
- [30] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic Mode Decomposition : Data-Driven Modeling of Complex Systems*. SIAM-Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2016.
- [31] Lennart Ljung. *System Identification : Theory for the User*. Prentice Hall, 1999.
- [32] Jean-Charles Loiseau and Steven L. Brunton. Constrained sparse galerkin regression. *Journal of Fluid Mechanics*, 838 :42–67, 2018.
- [33] Jean-Charles Loiseau, Bernd R. Noack, and Steven L. Brunton. Sparse reduced-order modeling : sensor-based dynamics to full-state estimation. *Journal of Fluid Mechanics*, 844 :459–490, 2018.
-

-
- [34] Andrew J. Majda and John Harlim. Physics constrained nonlinear regression models for time series. *Nonlinearity*, 26(1) :201, 2012.
- [35] Niall M. Mangan, J. Nathan Kutz, Steven L. Brunton, and Joshua L. Proctor. Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A*, 473(2204) :1–16, 2017.
- [36] Alexandre Mauroy, Igor Mezic, and Yoshihiko Susuki. *The Koopman Operator in Systems and Control Concepts, Methodologies, and Applications : Concepts, Methodologies, and Applications*. 01 2020.
- [37] Bernd R Noack, Konstantin Afanasiev, Marek Morzynski, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497 :335–363, 2003.
- [38] Bernd R. Noack, Konstantin Afanasiev, Marek Morzynski, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497 :335–363, 2003.
- [39] Joshua Proctor, Steven Brunton, and J. Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(2) :147–167, 2016.
- [40] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1) :142–161, 2016.
- [41] Joshua L Proctor and Philip A Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International Health*, 7(2) :139–145, 2015.
- [42] Clarence Rowley, Igor Mezic, Shervin Bagheri, Philipp Schlatter, and Dan Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641 :115–127, 2009.
- [43] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(e1602614), 2017.
- [44] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. In *Proc. R. Soc. A*, volume 473, page 20160446. The Royal Society, 2017.
- [45] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656 :5–28, 2010.
- [46] Peter J Schmid and Jörn Sesterhenn. Dynamic mode decomposition of numerical and experimental data. In *61st Annual Meeting of the APS Division of Fluid Dynamics*. American Physical Society, November 2008.
- [47] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923) :81–85, 2009.
- [48] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923) :81–85, 2009.
- [49] Gregory A. Snyder. A thesis submitted to the graduate division of the university of hawai‘i at mānoa in partial fulfillment of the requirements for the degree of master of science in mechanical engineering. Master’s thesis, University of Hawai‘i at Mānoa, December 2021.
- [50] Mariia Sorokina, Stylianos Sygletos, and Sergei Turitsyn. Sparse identification for nonlinear optical communication systems : Sino method. *Optics Express*, 24(26) :30433–30443, 2016.
- [51] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, pages 267–288, 1996.
- [52] Giang Tran and Rachel Ward. Exact recovery of chaotic systems from highly corrupted data. *Multiscale Modeling & Simulation*, 15(3) :1108–1129, 2017.
-

- [53] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12) :4655–4666, 2007.
- [54] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition : Theory and applications. *Journal of Computational Dynamics*, 1(2) :391–421, 2014.
- [55] B. van der Pol. The nonlinear theory of electric oscillations. *Proceedings of the Institute of Radio Engineers*, 22(9) :1051–1086, Sept. 1934.
- [56] Wei-Xing Wang, Rui Yang, Ying-Cheng Lai, Vassilios Kovanis, and Celso Grebogi. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical Review Letters*, 106 :154101–1–154101–4, 2011.
- [57] Matthew O Williams and Ioannis G Kevrekidis. A data-driven approximation of the koopman operator : Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6) :1307–1346, 2015.
- [58] Matthew O Williams, Clarence W Rowley, and Ioannis G Kevrekidis. A kernel approach to data-driven koopman spectral analysis. *Journal of Computational Dynamics*, 2(2) :247–265, 2015.
- [59] John Wright, Allen Yang, Arvind Ganesh, Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2) :210–227, 2009.
- [60] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(2) :301–320, 2005.