Dimensionality Reduction and Sparsity Promotion for Complex Dynamical Systems

Seth Michael Hirsh

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

J. Nathan Kutz, Chair

Bingni W. Brunton

Arka Majumdar

Program Authorized to Offer Degree:
Physics

University of Washington

Abstract

Dimensionality Reduction and Sparsity Promotion for Complex Dynamical Systems

Seth Michael Hirsh

Chair of the Supervisory Committee:
Robert Bolles and Yasuko Endo Professor J. Nathan Kutz
Applied Mathematics

This thesis provides several contributions to data-driven techniques for modeling spatio-temporal data, both developing the theory of existing methods and proposing new techniques. These methods leverage underlying structures, namely low dimensional and sparse representations to extract meaningful representations for a variety of dynamical systems.

We first focus on the dynamic mode decomposition (DMD), a popular dimensionality reduction technique, which assumes the data is governed by linear dynamics. We show that mean subtraction improves the performance of the method, explicitly characterizing the behavior theoretically and corroborating the results with real world datasets. This contribution is particularly significant as previous results have shown that mean subtraction has undesirable effects. We next transition to modeling nonlinear dynamics, and propose a dimensionality reduction method for time-varying linear dynamics. Our method, the spatiotemporal temporal intrinsic mode decomposition (STIMD), leverages spatial correlations to decompose data into a linear combination of intrinsic mode functions (IMFs). IMFs have the beneficial property that they have well-defined instantaneous frequencies. This method robustly models nonstationary signals, while still preserving much of the interpretibility of linear methods. We then unify results from differential geometry, time delay embeddings, and dimensionality reduction to show that nonlinear dynamical systems may be decomposed into a sparse and structured linear dynamical model plus forcing. We use this perspective to explain observed behavior in the recently developed Hankel Alternative View of Koopman (HAVOK) algorithm. We further propose modifications which not only more closely align with theory, but also improve model stability and reconstruction. Finally, we propose a new framework, uncertainty quantification for sparse identification of nonlinear dynamics (UQ-SINDy), which utilizes compressed sensing and Bayesian statistics to directly extract governing equations from data, providing explicit quantification of uncertainty in the model. This method is particularly powerful for systems which have sparse nonlinear representations and yields state of the art accuracy.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Number Page

ACKNOWLEDGMENTS

I would like to thank all of the people who have been supportive throughout the PhD. First, I would like to thank Nathan Kutz and Bing Brunton for being outstanding advisors, providing support, guidance, and patience. I have learned a great deal over the past four years from them and am truly grateful for the opportunity to work with them.

Additionally, I would like to thank all of my colleagues and co-authors, in particular Kameron Decker-Harris, Steve Brunton, David Barajas-Solano, and Sara Ichinaga. It has been a pleasure working with each of them on papers as co-authors and hearing their thoughts and perspectives. I would like to thank the members of my supervisory committee, Arka Majumdar, Jason Detwiler, Armita Nourmohammad, and Bruce Darling, for their thoughtful feedback throughout this process. I would like to thank the past and current members of the Brunton and Kutz labs. In particular, I would like to thank Satpreet Singh, Nancy Wang, Steve Peterson, Alice Schwarze, Charlie Fieseler, Emily Clark, Daniel Dylewsky, Chang Sun, and Zhe Bai for all of their insights and fruitful discussions.

Outside of academia, I had the opportunity to work with Qing Chao, Michael Hall, and Chiao Liu at Facebook Reality Labs and Arman Kaljahi, Gautam Kedia, and Ido Bright at Lyft. I am very grateful for their mentorship, from which I learned a great deal and broadened my perspectives.

Personally, I would like to thank all my family and friends who have been supportive throughout this process. In particular, I would like to thank my parents, my sister, and Xing. I am very thankful for their support and patience, listening to my struggles and providing helpful perspectives and advice.

Chapter 1

INTRODUCTION

The analysis of spatiotemporal signals is of critical importance for characterizing emerging large-scale measurements in a wide variety of scientific and engineering applications. Significant advances in sensor cost, data storage, and processing power have led to a rapidly increasing availability of data in domains including neuroscience, atmospheric sciences, and finance, to name a few. For the majority of these applications, the underlying dynamics are nonlinear, non-stationary, and the governing equations are poorly known at best. Therefore, data-driven modeling tools have become increasingly central, and the ability to extract interpretable structure and provide physical insights are crucial for advancing the field. This has motivated the study of dynamical systems with the goal of identifying and analyzing these systems. One key area of focus has been to model these systems, which allows for characterization, prediction, and control. Although many spatiotemporal datasets are very high dimensional, these systems can typically be well-represented by a small number of coordinates, which has increased interest in the development of dimensionality reduction and sparsity promoting techniques. Although these methods have shown great performance on data, much of their theory is not fully understood. The body of this work is divided into four chapters, in which we contribute to the theory of existing methods and propose both new methods and modifications to improve performance in modeling these systems.

1.1 Data Driven Discovery

In recent decades, there has been a rapid growth in measurements of systems generated by complex, nonlinear dynamics. Discovering meaningful representations of these dynamics are a long sought after goal, allowing for better characterization, prediction, and control. With decreases in sensor cost and increases in computing power, focus has increasingly turned towards data-driven techniques for discovering these meaningful representations in which the governing dynamics are easily identifiable.

Utilizing data-driven methods for physics discovery is nothing new. As a prime example, in the early 1600s, by studying data from the orbits of Mars [197], Johannes Kepler proposed three governing equations to described planetary motion. Additionally, in the early 1800s Gauss utilized the novel least squares method to accurately estimate the trajectory of the dwarf planet Ceres with limited data even when the problem was believed to be intractable. With recent advances in computing power and computational techniques, data-driven methods have become increasingly popular tools for analyzing these high dimensional systems. These methods have shown state of the art performance, for tasks such as forecasting [192], and control [11].

Although a promising approach, performing data driven discovery in a principled way presents many challenges, as meaningful representations of data are typically system dependent. Further, many of these methods are empirical in nature, thus making it difficult to determine their performance on unseen data. In this work we will primarily focus on two sets of techniques, modal decompositions and sparse identification of governing equations, exploring the theory of these methods and providing modifications in cases where their performance is limited.

1.2 Modal Decompositions

Modal decompositions are dimensionality reduction techniques in which the goal is to decompose data into a linear combination of interpretable states. These methods are prevalent throughout physics. Examples include in classical mechanics decomposing coupled oscillators into sums of normal modes [9, 70, 204, 108, 170, 189], in quantum mechanics decomposing the hydrogen atom in a linear combination of eigenstates [34, 45, 185, 176], and in electrodynamics decomposing the fields in a waveguide into a combination of TE or TM modes [76, 153, 188].

More formally, we can think of our data as a matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$. If our data lies in an r dimensional space (and therefore consists of r modes), our goal is to learn a decomposition

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^{\intercal}, \tag{1.1}$$

where $\boldsymbol{U}$ is an r dimensional basis of the system and $\boldsymbol{V}$ are the associated coordinates in this basis. For example, in the case of a spring mass system, the columns of $\boldsymbol{U}$ provide information about the relative orientations of the masses, while the columns of $\boldsymbol{V}$ describe the temporal evolution of these spatial modes. We define r to be the rank of the data.

This problem of choosing $\boldsymbol{U}$ and $\boldsymbol{V}$ is highly underdetermined. There are in general, many solutions for $\boldsymbol{U}$ and $\boldsymbol{V}$ that satisfy $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^{T}$, and different assumptions may be incorporated depending on the application. In this section we will discuss several recent methods which play key roles in the following chapters.

1.2.1 Linear Dynamical Systems

One of the most popular modal decompositions for dynamical systems is the dynamic mode decomposition. This method originated in the field of fluid dynamics starting with Peter Schmid [180, 181], and has since been applied in a wide range of fields, including neuroscience [29], the stock market [141], and climate science [124].

For this method we assume that we have a set of snapshots $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^m$, which are governed by some nonlinear dynamics $\boldsymbol{x}_{j+1} = \boldsymbol{F}(\boldsymbol{x}_j)$, where $\boldsymbol{F}$ is unknown [213]. The goal of DMD is to determine a best linear approximation

$$\boldsymbol{x}_{j+1} \approx \boldsymbol{A}\boldsymbol{x}_j. \tag{1.2}$$

With the linear dynamics matrix $\boldsymbol{A}$, we can then compute its eigendecomposition. The resulting eigenvectors $\boldsymbol{v}_i$, also known as the DMD modes, are the spatial modes of the system while the eigenvalues λ_i correspond to the frequencies of the system. Most notably, using the eigendecomposition we may reexpress the data in the form of (1.1), where $\boldsymbol{v}_i$ are the columns of $\boldsymbol{U}$ and the columns of $\boldsymbol{V}$ are of the form $\exp i\omega_i t$, where $\omega_i = \ln(\lambda_i/\Delta t)$.

In general, m may be very large and solving for $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ may be computationally prohibitive. Many techniques have been developed, including the companion matrix approach [180], Exact DMD [213], and optimized DMD [12]. Furthermore, there have been a wide variety of modifications to improve performance for specific applications, including control [163] and uncertainty quantification [201]. DMD is particularly attractive due to its computational simplicity and its interpretability. This has allowed for explicit characterizations of many properties such as the effects of noise [50, 84].

From a physical perspective, DMD is very similar to the extraction of normal modes. However, there is one key difference. Namely, when computing normal modes we choose to linearize about a fixed point, while in DMD we do not. From a data standpoint this is very similar to mean subtraction and is a common preprocessing step for several methods, such as PCA [228] and artificial neural networks. However, the effects of mean subtraction on DMD remain unknown. We will explore this relationship further in chapter 2.

1.2.2 Time Varying Linear Dynamics

When modeling dynamics with linear methods, such as the dynamic mode decomposition and the Fourier transform, we typically assume that our data consists of a linear combination of complex exponentials, or in the case of stable dynamics, sinusoids $x(t) = a\cos(\omega t)$. Data of this form satisfies $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$. As a first step towards modeling nonlinear dynamics we can relax these assumptions so that $\boldsymbol{A}$ is a slowly varying function of time [95]. This results in dynamics of the form $x(t) = a(t)\cos(\omega(t)t)$, which corresponds to a sinusoid with frequency modulation and amplitude modulation. Signals of this form are defined as intrinsic mode functions [98]. Decomposing signals into this form are particularly popular since IMFs have well-defined instantaneous frequencies $\omega(t)$.

One of the most popular techniques for decomposing data of this form is the empirical mode decomposition EMD [98, 167, 230]. This method assumes that a time series may be represented by a sum of intrinsic mode functions. This method has been empirically successful, with applications to geology [220], neural recordings [158], and finance [99]. However, its theoretical foundation is not well understood and its performance on unseen data is difficult to predict.

Several modifications have been proposed to improve robustness, and alternative methods with stronger theoretical foundations have since been proposed [49, 55, 69]. Most notably, Hou and Shi developed the nonlinear matching pursuit (NMP) method, which leveraged matching pursuit to extract IMFs from time series [93] . This method has several beneficial properties, including computational efficiency and convergence guarantees [96].

Although these methods have been successful at extracting IMFs, they are limited to one dimensional signals. Although multivariate extensions exist, incorporating IMFs into a modal decomposition framework remains a challenge.

1.2.3 Time Delay Embeddings

An alternative method for modeling nonlinear dynamics is focused on determining an effective set of coordinates in which the governing equations can be easily identified. For example, by discovering observables which exhibit linear dynamics, we can apply well understood methods such as DMD, which then utilize existing methods for modeling, prediction and control [227]. Discovering this set of observables is the key goal of Koopman Theory [33, 121], which has been a central motivation in the development of DMD [183].

Significant efforts have been focused on discovering such a set of observables [123, 131, 135, 151, 202, 227]. Although these methods have shown strong performance on data, their performance is typically system dependent. In addition, in many cases the measurements may not span the phase space, making it difficult to recover and model the dynamics.

One approach to remedy this issue has been to use time delay coordinates. In particular, to construct the state at time t, we stack $x(t)$ with copies of its history $x(t-\tau)$ delayed by some time τ. This technique gained significant attention with the development of Taken's theorem [203], which showed that with sufficient time delays even a single time series measurement could be used to reconstruct an attractor. Additionally, more recently it has been shown that high dimensional time delay embeddings have very good low rank approximations, making them promising tools for discovering low dimensional representations. Further results have related dynamics in these coordinates to the Koopman operator [7] and the Fourier transform [27]. Empirically, time delay embeddings are a core component in DMD, and other linear methods including, the eigensystem realization algorithm (ERA) [114], singular spectrum analysis (SSA) [28], and nonlinear Laplacian spectrum analysis [67]. In addition, this structure has also been incorporated into neural network architectures [217].

One recent development is the Hankel alternative view of Koopman (HAVOK) method [30], which leverages time delay embeddings to decompose chaotic dynamics into linear dynamics with intermittent forcing. This model was successfully applied to several systems, including measles outbreaks, Earth's magnetic field, and double pendulum data. Although empirically successful, the theoretical foundation for the HAVOK method is not well-understood and is an active area of research [118]. One surprising result is that for specific systems the linear model takes a specific structure, namely it is antisymmetric with nonzero elements above the diagonal. Understanding this structure and the theory of this method is key for improving the method and understanding its behavior on unseen data.

1.3 Sparse Identification

For many nonlinear dynamical systems, the governing equations may be represented by a few nonlinear terms. By directly identifying these nonlinear terms allows us to both accurately model a wider class of dynamics, while constructing meaningful representations. Towards this goal, there have been several approaches to directly discovering governing equations from data. One approach that has gained increased interest is on developing sparse representations of the dynamics, as these parsimonious representations can accommodate complex dynamics while maintining generalizibility and interpretibility [16, 32, 31, 136, 152, 162, 210, 221].

Here we focus on the Sparse Identification for Nonlinear Dynamics (SINDy) method, which leverages dictionary learning and compressed sensing to determine the governing equations [31]. In particular this method, constructs a library of possible terms in the model and then employs sparse regression to identify relevant terms. This approach has been successful in modeling a wide array of fields, including chemistry [90], optics [195], epidemiology [91], and plasma physics [47]. Furthermore, there have been a variety of modifications, including improved robustness to noise [42, 115], generalizations to partial differential equations [165, 175, 174], boundary conditions [190], and libraries of rational functions [140, 117].

One limitation of this method is the fact that it is difficult to how the method will generalize to unknown datasets. In particular, many datasets contain measurement noise or complex dynamics, which are highly sensitive to initial conditions. The SINDy method provides only a single candidate model of the data making it difficult to determine the variability in the results and how the model will generalize to new data.

1.4 Organization and Contributions

This thesis is organized into six chapters. In chapter 2, we show how mean-subtraction improves the dynamic mode decomposition, both from a theoretical perspective and through several real world examples, contrary to previous results. In chapter 3, we develop a new modal decomposition technique, the spatiotemporal intrinsic mode decomposition (STIMD), which leverages spatial correlations to model data with time varying linear dynamics. In chapter 4, we unify results from time delay embeddings, dimensionality reductions, and differential geometry, and show how these three fields can be combined to elicit the structured linear dynamics recently observed in the Hankel alternative view of Koopman (HAVOK) method. In chapter 5, we propose an uncertainty quantification for sparse identification of nonlinear dynamics (UQ-SINDy) framework, which uses sparse bayesian inference to determine governing equations. In chapter 6, we summarize these results and contributions and propose areas of future research. Summaries of chapters 2 through 5 follow below.

In chapter 2, we show that DMD with centered data is equivalent to incorporating an affine term in the dynamic model and is not equivalent to computing a discrete Fourier transform. Importantly, we show that DMD with centering can always be used to model affine dynamics. However, in many cases DMD without centering cannot model these dynamics, most notably if the data have full effective rank. Additionally, we generalize the notion of centering to extracting arbitrary, but known, fixed frequencies from the data. We corroborate these theoretical results numerically on three nonlinear examples: the Lorenz system, a surveillance video, and brain recordings. Since centering the data is simple and computationally efficient, we recommend it as a preprocessing step before DMD; furthermore, we suggest that it can be readily used in conjunction with many other popular implementations of the DMD algorithm.

In chapter 3, we propose a new modal decomposition technique that factors mixed time-series signals into a sum of spatiotemporal modes, with the constraint that the temporal components are intrinsic mode functions (IMFs). The key motivation is that IMFs allow the computation of meaningful Hilbert transforms of non-stationary data, from which instantaneous time-frequency representations may be derived. Our spatiotemporal intrinsic mode decomposition (STIMD) method leverages spatial correlations to generalize the extraction of IMFs from one-dimensional signals, commonly performed using the empirical mode decomposition (EMD), to multi-dimensional signals. Further, this data-driven method enables future-state prediction. We demonstrate STIMD on several synthetic examples, comparing it to common matrix factorization techniques, namely singular value decomposition (SVD), independent component analysis (ICA), and dynamic mode decomposition (DMD). We show that STIMD outperforms these methods at reconstruction and extracting interpretable modes. Next, we apply STIMD to analyze two real-world datasets, gravitational wave data and neural recordings from the rodent hippocampus.

In chapter 4, we unify properties of the Frenet-Serret frame with time delay embeddings and dimensionality reduction techniques to show that a dynamical system may be decomposed into a sparse linear dynamical model with forcing. In particular, the linear dynamical model is antisymmetric with nonzero elements exclusively along the sub- and super-diagonals and the forcing term is nonzero only in the last component. The nonzero elements of this model have a clear geometric meaning; namely they are the curvatures of the system. This is the first unification of these three fields to the best of our knowledge. This work is motivated by the recently developed Hankel Alternative View of Koopman (HAVOK) method, which combines time delay embeddings and dimensionality reduction techniques to construct a linear dynamical model with an additional forcing term. With this new perspective we explore its limitations and requirements and propose how small modifications to the HAVOK method induce this structure. These modifications improve model reconstruction and model stability, which we illustrate on several synthetic and real-world datasets, including measurements of measles outbreaks and double pendulum data.

In chapter 5, we propose a uncertainty quantification framework for directly extracting the governing equations for nonlinear dynamics. Our uncertainty quantification for sparse identification of nonlinear dynamics (UQ-SINDy) method combines advances in compressed sensing and dictionary learning with Bayesian statistics to determine an explicit parsiminous representation of the dynamics with estimates of both (1) the uncertainty in parameter values and (2) the inclusion probabilities of different terms in the model. We demonstrate this on two synthetic examples and real world dataset of lynx and hare populations. We find that the UQ-SINDy framework identifies the governing equations, even for highly noisy and sparsely sampled data and provides highly precise forecasts.

Bibliographic Notes

This dissertation is based upon the following publications:

1. Hirsh, S. M., Brunton, B. W., & Kutz, J. N. (2020). Data-driven spatiotemporal modal decomposition for time frequency analysis. Applied and Computational Harmonic Analysis, 49(3), 771-790.

2. Hirsh, S. M., Harris, K. D., Kutz, J. N., & Brunton, B. W. (2020). *Centering data improves the dynamic mode decomposition.* SIAM Journal on Applied Dynamical Systems, 19(3), 1920-1955.

3. Hirsh, S. M., Ichinaga, S. M., Brunton, S. L., Kutz, J. N., & Brunton, B. W. *Structured Time-Delay Models for Dynamical Systems with Connections to Frenet-Serret Frame.* In preparation.

4. Hirsh, S. M., Barajas-Solano, D., & Kutz, J. N. *Sparsifying Priors for Bayesian Uncertainty Quantification in System Identification.* In preparation.

Chapter 2

CENTERING IMPROVES THE DYNAMIC MODE DECOMPOSITION

2.1 Introduction

One popular method for modeling dynamical systems is the *dynamic mode decomposition* (DMD) [183, 181, 173, 143, 213, 123]. Like principal component analysis (PCA) [110, 228] and independent component analysis (ICA) [102], DMD is a dimensionality reduction technique that decomposes data into a set of spatial and temporal modes. Unlike PCA and ICA, DMD makes the additional assumption that the data are observations from an underlying dynamical system. In particular, the dynamics are assumed to be approximately linear, and the data are decomposed into pairs of interpretable spatial and temporal modes. DMD has been successfully applied in a wide variety of disciplines, including fluid dynamics [182], neuroscience [29], disease modeling [164], finance [141], and computer vision [77]. In addition, several extensions and variations to the DMD algorithm have been developed (see [113, 227, 124, 235, 163, 112, 12], among many others).

For many systems of interest, the dynamics we want to model are perturbations about equilibria. To name a few specific examples, in hydrodynamics we may model motion of a fluid about a base flow [149, 200]; in video processing we may extract the foreground from a static background [194]; and in climate science we may analyze anomalies that depart from long-term averages [82, 60]. Further, linearizing about equilibria provides key information on the stability of the system about these fixed points. If the data are oscillating about the fixed point, the mean of the measurement data is a natural estimate of an unknown equilibrium point; therefore, it is natural to apply DMD on mean-subtracted data.

In a complementary perspective, we may think of DMD computed over a short time

		Data Generation	
		Linear System $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$	Affine System $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$
Method	DMD w/o centering	✓ (Theorem 2.4.6)	Sometimes (Theorem 2.5.3)
	DMD w/ centering	✓ (Theorem 2.5.2)	✓ (Theorem 2.4.8)

Table 2.1: Comparison of performance of DMD with and without centering. A ✓ indicates that the method does correctly extract the spectrum and modes of the system in each column.

window as a multivariate Taylor expansion of the dynamics, discarding quadratic and higher order terms. It follows that the model should include an affine, or bias, term (Fig. 2.1), which is usually not a part of the DMD model; if DMD is computed on centered data, then this affine term is expected to be small (in fact, one of our results is that it will be zero).

In this chapter, we show that centering data improves the performance of DMD. Previous work has suggested that computing the DMD of centered data may be restrictive and have undesirable consequences [43]. In particular, Chen *et al.* [43] show that DMD on mean-subtracted data is equivalent to a temporal discrete Fourier transform (DFT), restricting the frequencies extracted to be independent of the dataset. This argument hinged on the mean-subtracted data being full rank; however, here we show that, in linear systems that contain a nonzero fixed point, mean-subtracted data will always have linearly dependent columns. Therefore, DMD on centered data does not converge to the DFT. Furthermore, our proposed method of centering the data successfully extracts the equilibrium and dynamics about this equilibrium.

In Section 2.2 we review the DMD algorithm, focusing on comparing the SVD-based approach to the companion matrix approach. We propose centering the data in Section 2.3, showing that it is equivalent to incorporating an affine term in the DMD model. Section 2.4 concerns the uniqueness of the DMD modes and whether the DMD problem is well-posed, generalizing previous results to the case where data may be low rank. Section 2.5 compares DMD with and without centering, including theory and numerical examples. We find that, in the case of linear dynamics about an equilibrium point, DMD with centering can always extract the correct dynamics. However, DMD without centering sometimes produces an inaccurate model. These results are summarized in Table 2.1. The work by Chen *et al.* [43] is discussed in detail in Section 2.6, where we argue that DMD with centering is not equivalent to a DFT. This notion of data centering is generalized in Section 2.7 to extract dynamics while subtracting any known fixed frequencies. Section 2.8 demonstrates DMD with centering and fixed frequency subtraction on three nonlinear examples, the Lorenz system, background-foreground separation of a video, and brain recordings. Many of the proofs in these sections assume the data are state observations of a linear system. Applications of DMD do not necessarily satisfy this assumption. As a practical recommendation, we suggest centering data as a preprocessing step in DMD.

2.2 Background

Initially developed in the fluid dynamics community, dynamic mode decomposition (DMD) has become a popular tool for analyzing large-scale dynamical systems in many different application domains [123, 182]. In this section we briefly review two formulations of this problem.

Consider a set of $T+1$ measurement snapshots $\boldsymbol{x}_j \in \mathbb{R}^n$ for $j = 1, \dots, T+1$, which are generated by linear dynamics,

$$\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j. \tag{2.1}$$

Data which exactly satisfy (2.1) form a Krylov sequence [211].

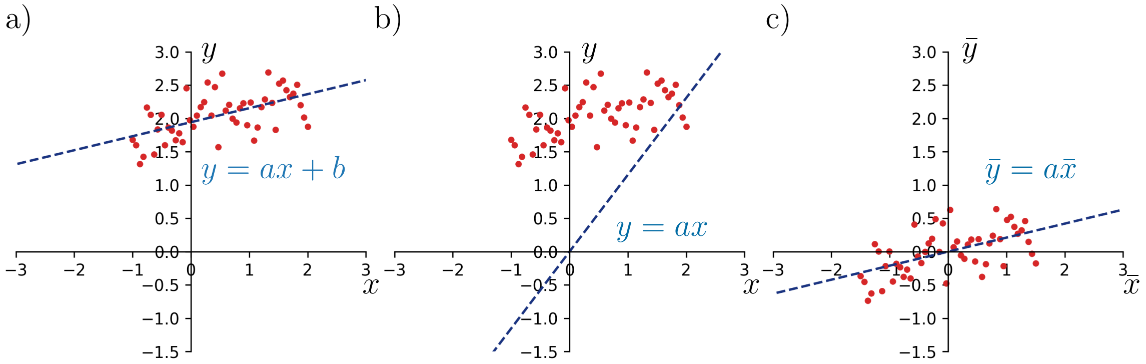


Figure 2.1: An illustration of the benefit of centering for one-dimensional regression, where the data (x_j, y_j) is generated by an affine model with noise. a) Data fit to affine model $y = ax + b$ yields a good fit. b) Data fit to linear model $y = ax$ yields a poor fit. c) Centered data $(\bar{x}_j \bar{y}_j)$ fit to linear model $\bar{y} = a\bar{x}$ yields a good fit.

The goal of DMD is to characterize the dynamics of the system by the eigendecomposition of the linear operator $\boldsymbol{A} \in \mathbb{R}^{n \times n}$:

$$\boldsymbol{A}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i \quad \text{for} \quad i = 1, \ldots, n. \tag{2.2}$$

The eigenvectors $\boldsymbol{v}_i$ are typically refered to as the DMD modes. For our theoretical results, we typically assume that the eigenvalues $\lambda_i \neq 0$ are distinct. For many systems of interest, the true dynamics may be nonlinear and/or stochastic. In addition, observations may contain measurement noise. Where the measurements deviate from true linear dynamics, the goal of DMD is to find the best linear approximation.

Many alternative methods have been developed to compute these eigenvalue/eigenvector pairs, including the Arnoldi method, or in the case where $\boldsymbol{A}$ is symmetric the Lanczos algorithm [10, 126]. However, these methods typically involve explicitly computing $\boldsymbol{A}$ which may be computationally prohibitive if n is large [181].

2.2.1 SVD-based DMD

We first summarize the most commonly used formulation of DMD, the SVD-based approached also known as exact DMD [213]. First, let us define the pair of snapshot matrices containing the measurement vectors

$$\boldsymbol{X}_1 = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_T \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \boldsymbol{X}_2 = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_2 & \boldsymbol{x}_3 & \cdots & \boldsymbol{x}_{T+1} \\ | & | & & | \end{bmatrix}. \tag{2.3}$$

If the snapshots satisfy (2.1), then we have that

$$\boldsymbol{X}_2 = \boldsymbol{A}\boldsymbol{X}_1. \tag{2.4}$$

Otherwise, we hope to discover the "best" $\boldsymbol{A}$ which approximately satisfies this equation.

One solution to (2.4) is obtained by regression with least squares minimization; we define this solution to be $\hat{\boldsymbol{A}}$. In general, (2.4) may be consistent (having at least one solution), or inconsistent (having no solution). With these two cases, the corresponding minimization problem takes the form,

$$\hat{\boldsymbol{A}} = \begin{cases} \min_{\boldsymbol{A}} \|\boldsymbol{A}\|_F \text{ such that } \boldsymbol{X}_2 = \boldsymbol{A}\boldsymbol{X}_1 & \text{if (2.4) is consistent} \\ \min_{\boldsymbol{A}} \|\boldsymbol{X}_2 - \boldsymbol{A}\boldsymbol{X}_1\|_F & \text{if (2.4) is inconsistent} \end{cases}. \tag{2.5}$$

The solution in either case is given by the least squares fit,

$$\hat{\boldsymbol{A}} := \boldsymbol{X}_2\boldsymbol{X}_1^{\dagger}, \tag{2.6}$$

where $\boldsymbol{X}_1^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of $\boldsymbol{X}_1$ [156]. The DMD modes and eigenvalues in the SVD approach are the eigenvectors and eigenvalues of $\hat{\boldsymbol{A}}$, respectively.

When n is large, it may not be practical to compute $\hat{\boldsymbol{A}} \in \mathbb{R}^{n \times n}$ and its eigendecomposition directly. If $\boldsymbol{X}_1$ is low rank or approximately low rank, we may project the dynamics to a lower dimensional basis. In particular, if $\boldsymbol{X}_1$ has rank r, we may compute the reduced SVD,

$$\boldsymbol{X}_1^r := \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^{\intercal}, \tag{2.7}$$

where the left singular vectors $\boldsymbol{U}_r \in \mathbb{R}^{n\times r}$ and right singular vectors $\boldsymbol{V}_r \in \mathbb{R}^{T\times r}$ are orthogonal matrices and $\boldsymbol{\Sigma}_r$ is a real positive diagonal matrix [73]. When measurement noise is present, we define r to be the effective rank of the system (discussed in detail in Section 2.2.3).

We can then define the matrix,

$$\tilde{\boldsymbol{A}} \coloneqq \boldsymbol{U}_r^{\intercal}\boldsymbol{X}_2\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1},$$

where $\tilde{\boldsymbol{A}} \in \mathbb{R}^{r\times r}$ is much smaller in size than $\boldsymbol{A}$. Importantly, Tu *et al.* showed that the eigenvalues of $\tilde{\boldsymbol{A}}$ are precisely the nonzero eigenvalues of $\boldsymbol{A}$ [213]. The corresponding eigenvectors $\boldsymbol{\phi}_i$ of $\boldsymbol{A}$ can be found by first computing the eigenvectors $\boldsymbol{w}_i$ of $\tilde{\boldsymbol{A}}$,

$$\tilde{\boldsymbol{A}}\boldsymbol{w}_i = \lambda_i\boldsymbol{w}_i,$$

and then projecting into the original measurement space,

$$\boldsymbol{\phi}_i = \frac{1}{\lambda_i}\boldsymbol{X}_2\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\boldsymbol{w}_i. \tag{2.8}$$

In the case where the ranges of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are equal, (2.8) reduces to $\boldsymbol{\phi}_i = \boldsymbol{U}_r\boldsymbol{w}_i$.

2.2.2 Companion Matrix Approach

An alternative formulation of DMD focuses on the computation of a so-called companion matrix. Although it is less commonly used in practice, this original formulation by Schmid [181] is analytically simpler and has been used in some key theoretical work [43, 7].

We again consider $T+1$ snapshots $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1} \in \mathbb{R}^n$ which satisfy (2.1). We may express the last snapshot $T+1$ as a linear combination of the first T states and a residual $\boldsymbol{r} \in \mathbb{R}^n$ which is orthogonal to these T states,

$$\boldsymbol{x}_{T+1} = \sum_{j=1}^{T} c_j\boldsymbol{x}_j + \boldsymbol{r} \text{ such that } \boldsymbol{r} \perp \operatorname{span}\{\boldsymbol{x}_1, \ldots \boldsymbol{x}_T\},$$

where $c_j \in \mathbb{R}$. Equivalently, we may write in matrix form,

$$\boldsymbol{X}_2 = \boldsymbol{X}_1\boldsymbol{C} + \boldsymbol{r}\boldsymbol{e}_T^{\intercal}, \tag{2.9}$$

where $\boldsymbol{e}_T = [0, \ldots, 0, 1]^\intercal$, and

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & \cdots & 0 & c_1 \\ 1 & 0 & \cdots & 0 & c_2 \\ 0 & 1 & & 0 & c_3 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & c_T \end{bmatrix} \tag{2.10}$$

is called the companion matrix. The least squares solution for $\boldsymbol{c} = [c_1, \ldots, c_T]$ is then given by $\boldsymbol{c} = \boldsymbol{X}_1^\dagger \boldsymbol{x}_{T+1}$. Note that all of the residual error in the model is placed on the last time snapshot. The least squares solution $\boldsymbol{C}$ to (2.9) is unique if and only if $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ are linearly independent [43]. If $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ are linearly independent, then $\boldsymbol{C}$ must also equal the least squares solution,

$$\boldsymbol{C} = \boldsymbol{X}_1^\dagger \boldsymbol{X}_2.$$

In some cases, the DMD modes (eigenvalues and eigenvectors of $\boldsymbol{A}$, assuming (2.1)) are related to the eigenvalues and eigenvectors of the companion matrix $\boldsymbol{C}$ [181], but these eigenvalues are, in general, not equal. In particular, the eigenvalues are only guaranteed to be equal if the columns of $\boldsymbol{X}_1$ are linearly independent [43, 181].

2.2.3 Rank vs. Effective Rank

If $\boldsymbol{X}_1$ has full column rank, then the companion matrix approach described in Section 2.2.2 is equivalent to computing the DMD modes as in (2.2). In the presence of measurement noise, $\boldsymbol{X}_1$ will almost surely have full column rank even in the case where $\boldsymbol{A}$ is low-rank ($r < T < n$). In that case, even though the companion matrix approach (2.10) has a well-posed solution, it yields the wrong number of eigenvalues. Specifically, the companion matrix approach yields T modes while there are only r signal modes masked by noise. On the other hand, the SVD-based approach (2.7) can filter out these noise modes with a good estimate of rank($\boldsymbol{A}$). Formally, we define the effective rank as follows:

Definition 2.2.1. *Given a set of noisy measurements* $\boldsymbol{Y} = \boldsymbol{X} + \eta \boldsymbol{Z}$, *where* $\boldsymbol{X}$ *is low rank and elements of* $\boldsymbol{Z}$ *are drawn independently from a random distribution with zero mean and finite variance, we define the* ***effective rank*** *of* $\boldsymbol{Y}$ *to be the rank of* $\boldsymbol{X}$*.*

In other words, the effective rank of $\boldsymbol{Y}$ is the rank of the data with no measurement noise ($\eta = 0$). In general, the effective rank of the data is unknown. However, it may be estimated from the SVD spectrum [62, 218]. We now claim (and later show, in Section 2.6) that the companion matrix approach yields the DMD modes if and only if $\boldsymbol{X}_1$ not only has full column rank but also *full effective column rank*. Although subtle, this distinction will play an important role in Section 2.6.

2.3 Centering Data

DMD as defined in (2.3) and (2.4) can be thought of as a multivariate regression of the dynamics. If the mean of $\boldsymbol{X}$ is not zero, as would occur with data measured about a non-zero equilibrium or data acquired over a short time interval, then the DMD model would be improved with an additional affine term:

$$\boldsymbol{X}_2 = \boldsymbol{A}\boldsymbol{X}_1 + \boldsymbol{b}\mathbf{1}^\intercal, \tag{2.11}$$

where $\boldsymbol{b} \in \mathbb{R}^n$ and $\mathbf{1}$ is a vector of length T whose elements are all one. The corresponding minimization problem to find $\boldsymbol{A}$ and $\boldsymbol{b}$ is given by

$$\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{b}} = \begin{cases} \operatorname{argmin}_{\boldsymbol{A},\boldsymbol{b}} \|\boldsymbol{A}\|_F \text{ s.t. } \boldsymbol{A}\boldsymbol{X}_1 + \boldsymbol{b}\mathbf{1}^\intercal = \boldsymbol{X}_2 & \text{if (2.11) is consistent} \\ \operatorname{argmin}_{\boldsymbol{A},\boldsymbol{b}} \|\boldsymbol{A}\boldsymbol{X}_1 + \boldsymbol{b}\mathbf{1}^\intercal - \boldsymbol{X}_2\|_F^2 & \text{if (2.11) is inconsistent.} \end{cases} \tag{2.12}$$

As illustrated in Fig. 2.1, the incorporation of an affine term in the one-dimensional regression model is equivalent to centering x_j and y_j in the data. For high-dimensional data, we compute the means of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ as

$$\boldsymbol{\mu}_1 = \frac{\boldsymbol{X}_1\mathbf{1}}{\mathbf{1}^\intercal\mathbf{1}} \quad \text{and} \quad \boldsymbol{\mu}_2 = \frac{\boldsymbol{X}_2\mathbf{1}}{\mathbf{1}^\intercal\mathbf{1}}.$$

The corresponding mean-subtracted or *centered* data matrices are

$$\bar{\boldsymbol{X}}_1 = \boldsymbol{X}_1 - \boldsymbol{\mu}_1 \mathbf{1}^\intercal \quad \text{and} \quad \bar{\boldsymbol{X}}_2 = \boldsymbol{X}_2 - \boldsymbol{\mu}_2 \mathbf{1}^\intercal,$$

and we now solve the unbiased regression problem

$$\bar{\boldsymbol{X}}_2 = \bar{\boldsymbol{A}} \bar{\boldsymbol{X}}_1. \tag{2.13}$$

The least squares solution to (2.13) is given by

$$\bar{\boldsymbol{A}} = \begin{cases} \operatorname{argmin}_{\boldsymbol{A}} \|\boldsymbol{A}\|_F \ \text{ s.t. } \boldsymbol{A}\bar{\boldsymbol{X}}_1 = \bar{\boldsymbol{X}}_2 & \text{if (2.13) is consistent} \\ \operatorname{argmin}_{\boldsymbol{A}} \left\|\boldsymbol{A}\bar{\boldsymbol{X}}_1 - \bar{\boldsymbol{X}}_2\right\|_F^2 & \text{if (2.13) is inconsistent} \end{cases}. \tag{2.14}$$

Importantly, the minimization problem (2.14) is simpler to solve than the one in (2.12). We show in Proposition 2.3.1 that they are equivalent, yielding $\tilde{\boldsymbol{A}} = \bar{\boldsymbol{A}}$. The following Proposition, which we include for completeness, is well-known among statisticians in the setting of multivariate regression:

Proposition 2.3.1. *Let $\boldsymbol{X}_1$ and $\boldsymbol{X}_2 \in \mathbb{R}^{n\times T}$ be arbitrary matrices. The minimization problems* (2.12) *and* (2.14) *are equivalent, with solutions $\tilde{\boldsymbol{A}} = \bar{\boldsymbol{A}}$ and $\tilde{\boldsymbol{b}} = \boldsymbol{\mu}_2 - \bar{\boldsymbol{A}}\boldsymbol{\mu}_1$.*

Proof. We have two cases to consider, depending on whether the affine system of equations (2.11) is linearly consistent (has at least one solution) or inconsistent (has no solution). We will show that system (2.11) is consistent if and only if (2.13) is consistent as well.

Case 1: Consistent

When (2.11) is consistent, the affine problem (2.12) is in the constrained (at least one solution for $\boldsymbol{A}$) case. Note that we do not minimize over the norm of $\boldsymbol{b}$. Multiplying the constraint by $\frac{\mathbf{1}}{\mathbf{1}^\intercal \mathbf{1}}$ yields,

$$\boldsymbol{A}\frac{\boldsymbol{X}_1 \mathbf{1}}{\mathbf{1}^\intercal \mathbf{1}} + \boldsymbol{b}\frac{\mathbf{1}^\intercal \mathbf{1}}{\mathbf{1}^\intercal \mathbf{1}} = \frac{\boldsymbol{X}_2 \mathbf{1}}{\mathbf{1}^\intercal \mathbf{1}},$$

which can be rearranged to find $\tilde{\boldsymbol{b}} = \boldsymbol{\mu}_2 - \boldsymbol{A}\boldsymbol{\mu}_1$. Thus we can write (2.12) as

$$\min_{\boldsymbol{A}} \|\boldsymbol{A}\|_F \ \text{ such that } \bar{\boldsymbol{X}}_2 = \boldsymbol{A}\bar{\boldsymbol{X}}_1,$$

which is precisely (2.14). Note that, since we assumed the constraint is satisfiable, this implies that the centered system of equations (2.13) is consistent.

Case 2: Inconsistent

If no solution to (2.11) exists, then we minimize the residual error without constraints. Taking the gradient with respect to $\boldsymbol{b}$ and setting it equal to 0 yields

$$\mathbf{1}^{\intercal}\boldsymbol{X}_1^{\intercal}\boldsymbol{A}^{\intercal} + \boldsymbol{b}^{\intercal}\mathbf{1}^{\intercal}\mathbf{1} = \mathbf{1}^{\intercal}\boldsymbol{X}_2^{\intercal}$$

and rearranging, we again find that $\tilde{\boldsymbol{b}} = \boldsymbol{\mu}_2 - \boldsymbol{A}\boldsymbol{\mu}_1$. Plugging this into (2.12), yields the minimization problem

$$\begin{aligned}\min_{\boldsymbol{A}} \left\|\boldsymbol{A}\boldsymbol{X}_1 + (\boldsymbol{\mu}_2 - \boldsymbol{A}\boldsymbol{\mu}_1)\,\mathbf{1}^{\intercal} - \boldsymbol{X}_2\right\|_F^2 &= \min_{\boldsymbol{A}} \left\|\boldsymbol{A}\left(\boldsymbol{X}_1 - \boldsymbol{\mu}_1\mathbf{1}^{\intercal}\right) - \left(\boldsymbol{X}_2 - \boldsymbol{\mu}_2\mathbf{1}^{\intercal}\right)\right\|_F^2 \\ &= \min_{\boldsymbol{A}} \left\|\boldsymbol{A}\bar{\boldsymbol{X}}_1 - \bar{\boldsymbol{X}}_2\right\|_F^2,\end{aligned}$$

which is precisely (2.14). Note that this also must be inconsistent, otherwise the affine problem would be consistent, and we would obtain a contradiction.

□

Remark 2.3.2. *We make no assumptions about the matrices* $\boldsymbol{X}_1$ *and* $\boldsymbol{X}_2$ *in Proposition 2.3.1. Therefore, this result does not depend on the system being linear or being generated by a dynamical system, and thus it is applicable in all regression settings.*

Instead of centering $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ individually, we may also choose to subtract the overall mean $\boldsymbol{\mu} = \frac{1}{T+1}\sum_{j=1}^{T+1}\boldsymbol{x}_j$ from the data. Mean-subtraction of data and normalization of variance is standard in matrix factorization algorithms such as PCA [228] and ICA [102]. In many cases, $\boldsymbol{\mu}$ is very similar to $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. In particular, $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$, and $\boldsymbol{\mu}$ are all approximately equal in the case of neutral dynamics (all of the DMD eigenvalues lie near the unit circle). Consequently, in this case subtracting the overall mean should yield similar results to using $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. However, in the presence of transients or unstable behavior, these three values may be very different.

2.4 Uniqueness of Modes

The remainder of this paper compares DMD modes and eigenvalues computed with and without centering. To perform such a comparison, it is necessary that we first establish uniqueness of the DMD modes and corresponding eigenvalues for a linear system (Section 2.4.1). We then follow with a similar proof for the uniqueness of modes data generated by an affine linear system (Section 2.4.2). This is key for showing that the modes from DMD with centering are well-defined.

2.4.1 Uniqueness of Dynamic Mode Decomposition

Following (2.1), (2.3), and (2.4), assume we have sequential snapshots of data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1} \in \mathbb{R}^n$ that are generated by linear dynamics (2.1). In general, there may be infinitely many matrices $\boldsymbol{A}'$ that satisfy

$$\boldsymbol{x}_{j+1} = \boldsymbol{A}' \boldsymbol{x}_j.$$

Chen *et al.* show, using the companion matrix approach, that although $\boldsymbol{A}'$ is not unique, the corresponding eigenvectors and eigenvalues of $\boldsymbol{A}'$ are, under specific conditions:

Theorem 2.4.1 (Chen et al. [43], Theorem 1 (rephrased))**.** *For the case of $T = n$, the choice of eigenvalues $\lambda_1, \ldots, \lambda_n$ and corresponding eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ are unique up to a reordering in j, if and only if $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ are linearly independent and $\lambda_1, \ldots, \lambda_n$ are distinct.*

In other words, even though $\boldsymbol{A}'$ is not unique, all n eigenvalues and eigenvectors of $\boldsymbol{A}'$ are unique if and only if $\boldsymbol{X}$ has full column rank and the eigenvalues are distinct. In the case of low-rank data, $\boldsymbol{X}$ will not have full column rank and the eigenvalues of $\boldsymbol{A}'$ will not be distinct, since $\boldsymbol{A}'$ will have a zero eigenvalue with multiplicity greater than 1. Consequently, this theorem does not provide much relevant information about uniqueness in the case of low-rank dynamics. To remedy this, we generalize this result to the case of low-rank data and prove that the nonzero eigenvalues and corresponding eigenvectors are unique. We first establish two useful lemmas:

Lemma 2.4.2. *Consider the* $(p+1)\times q$ rectangular *Vandermonde matrix*

$$\boldsymbol{\Lambda} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_q \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_1^p & \lambda_2^p & \cdots & \lambda_q^p \end{bmatrix}.$$

Then the q *columns of* $\boldsymbol{\Lambda}$ *are linearly independent (*$\boldsymbol{\Lambda}$ *has full column rank) if and only if* $q \leq p+1$ *and* $\lambda_1, \lambda_2, \ldots, \lambda_q$ *are distinct.*

Proof. Assume $q \leq p+1$ (if not, $\operatorname{rank}(\boldsymbol{\Lambda}) \leq p+1 < q$). We form the $q \times q$ submatrix

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_r \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_1^{q-1} & \lambda_2^{q-1} & \cdots & \lambda_q^{q-1} \end{bmatrix},$$

which has nonzero determinant [214] if and only if the eigenvalues are distinct. $\square$

Lemma 2.4.3. *Suppose we have sequential time series snapshots* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1}$ *such that* $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$ *for* $j = 1, \ldots, T$. *Assume that* $\boldsymbol{A}$ *has precisely* r *distinct, nonzero eigenvalues and is the matrix with the smallest rank such that this holds.*

- *If* $\operatorname{range}(\boldsymbol{X}_1) = \operatorname{range}(\boldsymbol{X}_2)$, *then* $\boldsymbol{X}$ *may be expressed as*

$$\boldsymbol{X} = \underbrace{\begin{bmatrix} | & | & & | \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_r \\ | & | & & | \end{bmatrix}}_{\boldsymbol{V}} \underbrace{\begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^T \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^T \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_r & \lambda_r^2 & \cdots & \lambda_r^T \end{bmatrix}}_{\boldsymbol{\Lambda}^{\intercal}}, \tag{2.15}$$

  where $\lambda_1, \ldots, \lambda_r$ *and* $\boldsymbol{v}_1, \ldots \boldsymbol{v}_r$ *are distinct nonzero eigenvalues and eigenvectors of* $\boldsymbol{A}$, *respectively.*

- *If* range$(\boldsymbol{X}_1) \neq$ range$(\boldsymbol{X}_2)$*, then* $\boldsymbol{X}$ *may be expressed as*

$$\boldsymbol{X} = \underbrace{\begin{bmatrix} | & | & | & & | \\ \boldsymbol{v}_0 & \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_r \\ | & | & | & & | \end{bmatrix}}_{\boldsymbol{V}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^T \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^T \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_r & \lambda_r^2 & \cdots & \lambda_r^T \end{bmatrix}}_{\boldsymbol{\Lambda}^\intercal}, \tag{2.16}$$

  where $\boldsymbol{v}_0 \in \mathrm{Null}(\boldsymbol{A})$*.*

Remark 2.4.4. *The condition on minimal rank is equivalent to requiring that* $\mathrm{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r\} = \mathrm{range}(\boldsymbol{X}_2)$*. If one considers another* $\boldsymbol{A}'$ *that the data are linearly consistent with but with larger rank, this will lead to another column in* $\boldsymbol{V}$ *and a corresponding column of zeros in* $\boldsymbol{\Lambda}$*.*

Proof. Assume range$(\boldsymbol{X}_1) =$ range$(\boldsymbol{X}_2)$. Since $\boldsymbol{X}_2 = \boldsymbol{A}\boldsymbol{X}_1$, $\boldsymbol{x}_1 \in$ range$(\boldsymbol{X}_1)$, and range$(\boldsymbol{X}_1) =$ range$(\boldsymbol{X}_2)$, then $\boldsymbol{x}_1 \in$ range$(\boldsymbol{X}_2) \subseteq$ range$(\boldsymbol{A})$. Since the eigenvectors corresponding to the r nonzero eigenvalues form a basis for these ranges, we may express $\boldsymbol{x}_1$ as a linear combination of these vectors, scaled appropriately, so that

$$\boldsymbol{x}_1 = \sum_{i=1}^{r} \boldsymbol{v}_i.$$

We note that eigenvalues corresponding to the $\boldsymbol{v}_i$'s are distinct. Otherwise, they can be summed together in the initial condition $\boldsymbol{x}_1$. Recursively applying $\boldsymbol{A}$ to $\boldsymbol{x}_1$,

$$\boldsymbol{x}_2 = \boldsymbol{A}\boldsymbol{x}_1 = \boldsymbol{A}\sum_{i=1}^{r} \boldsymbol{v}_i = \sum_{i=1}^{r} \lambda_i \boldsymbol{v}_r,$$
$$\boldsymbol{x}_3 = \boldsymbol{A}^2\boldsymbol{x}_1 = \sum_{i=1}^{r} \lambda_i^2 \boldsymbol{v}_i,$$

and in general,

$$\boldsymbol{x}_k = \boldsymbol{A}^{k-1}\boldsymbol{x}_1 = \sum_{i=1}^{r} \lambda_i^{k-1} \boldsymbol{v}_i.$$

Putting this in matrix form yields (2.15).

If range($\boldsymbol{X}_1$) $\neq$ range($\boldsymbol{X}_2$), then $\boldsymbol{x}_1$ has a component $\boldsymbol{v}_0$ which lies in Null($\boldsymbol{A}$). We scale $\boldsymbol{v}_0$ so that $\boldsymbol{x}_1 = \sum_{i=0}^{r} \boldsymbol{v}_i$. Since $\boldsymbol{v}_0$ has an associated eigenvalue $\lambda_0 = 0$, for $k \geq 1$,

$$\boldsymbol{x}_k = \boldsymbol{A}^{k-1}\boldsymbol{x}_1 = \sum_{i=0}^{r} \lambda_i^{k-1}\boldsymbol{v}_i = \sum_{i=1}^{r} \lambda_i^{k-1}\boldsymbol{v}_i,$$

which yields the decomposition in (2.16). □

Note that since the columns of $\boldsymbol{V}$ correspond to eigenvectors of distinct eigenvalues of $\boldsymbol{A}$, $\boldsymbol{V}$ has full column rank. We now introduce a definition of well-posedness for the DMD problem.

Definition 2.4.5. *Suppose we have sequential time series snapshots* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1}$ *such that* $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$. *Let* $\boldsymbol{A}$ *have* r *nonzero and distinct eigenvalues* $\lambda_1, \ldots, \lambda_r$ *and corresponding eigenvectors* $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$. *We say that the* **DMD problem is well-posed** *if the conditions*

1. $\boldsymbol{x}_1$ *is not orthogonal to any* $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$, *and either*

2a. $T \geq r$ *and* $\boldsymbol{X}_1$ *and* $\boldsymbol{X}_2$ *share the same range, or*

2b. $T \geq r + 1$,

are satisfied.

We are primarily interested in recovering dynamics for a linear system. It is unclear the meaning of resulting DMD modes for a nonlinear dynamical system even if the data are, for example, linearly consistent. If we assume that the data come from a linear system, then whether or not we can recover that dynamical system depends on the properties of $\boldsymbol{A}$. These assumptions of well-posedness give sufficient conditions for recovering $\boldsymbol{A}$. The DMD problem being well-posed depends on the assumptions made about the underlying system. In particular, it depends on the rank of the underlying system r. In theory, r is unknown, but we can estimate it using results such as [62].

In the following sections we apply different methods (such as DMD with centering and DMD without centering) to various synthetic and real datasets. To compare their performances it is important to establish a notion of uniqueness. In particular, in the following sections we plot the resulting spectra for these different methods. By showing that the modes for a given linear system are unique, then we establish that there is a true set of modes for the linear system. By comparing the spectra extracted using each of these methods to the true system we can assert whether a method can accurately model the dynamics of that system.

We now prove our main uniqueness theorem.

Theorem 2.4.6 (Uniqueness)**.** *Suppose we have sequential time series snapshots* $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{T+1}$ *such that* $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$ *for* $j = 1, \dots, T$*, where* $\boldsymbol{A}$ *has* r *nonzero and distinct eigenvalues* $\lambda_1, \dots, \lambda_r$ *and corresponding eigenvectors,* $\boldsymbol{v}_1, \dots, \boldsymbol{v}_r$*. Let* $\boldsymbol{A}'$ *be any other rank* r *matrix which satisfies* $\boldsymbol{x}_{j+1} = \boldsymbol{A}'\boldsymbol{x}_j$*. If the DMD problem is well-posed, then* $\boldsymbol{A}'$ *has the same* r *nonzero eigenvalues* $\lambda_1, \dots, \lambda_r$ *and corresponding eigenvectors* $\boldsymbol{v}_1, \dots, \boldsymbol{v}_r$ *as* $\boldsymbol{A}$*, and these are unique up to scaling.*

Proof. First, suppose range($\boldsymbol{X}_1$) = range($\boldsymbol{X}_2$). Since $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$, if we define $\boldsymbol{X}$ to be the matrix containing all $T+1$ snapshots, then by Lemma 2.4.3 we can factor $\boldsymbol{X}$ as follows,

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{T+1} \\ | & | & & | \end{bmatrix} = \underbrace{\begin{bmatrix} | & | & & | \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_r \\ | & | & & | \end{bmatrix}}_{\boldsymbol{V}} \underbrace{\begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^T \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^T \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_r & \lambda_r^2 & \cdots & \lambda_r^T \end{bmatrix}}_{\boldsymbol{\Lambda}^T}, \tag{2.17}$$

where $\boldsymbol{v}_j$ are the eigenvectors of $\boldsymbol{A}$ scaled appropriately so that $\boldsymbol{x}_1 = \sum_{i=1}^{r} \boldsymbol{v}_j$. We denote these matrices $\boldsymbol{V}$ and $\boldsymbol{\Lambda}$ and note that $\boldsymbol{V}$ and $\boldsymbol{\Lambda}$ have full column rank (Lemma 2.4.2).

Suppose there exists another solution with corresponding eigenvalues $\lambda_1', \dots \lambda_r'$ and eigenvectors $\boldsymbol{v}'_1, \ldots, \boldsymbol{v}'_r$. We construct another factorization,

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{v}'_1 & \boldsymbol{v}'_2 & \cdots & \boldsymbol{v}'_r \\ | & | & & | \end{bmatrix} \begin{bmatrix} 1 & \lambda'_1 & \lambda'^2_1 & \cdots & \lambda'^T_1 \\ 1 & \lambda'_2 & \lambda'^2_2 & \cdots & \lambda'^T_2 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda'_r & \lambda'^2_r & \cdots & \lambda'^T_r \end{bmatrix}.$$

For the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{A'}$ to be different, there must exist some λ'_i which does not equal $\lambda_1, \ldots, \lambda_r$. Since $\left\{[1,\ \lambda_i,\ \lambda_i^2, \cdots,\ \lambda_i^T] \text{ for } i = 1, \ldots, r\right\}$ spans the row space of $\boldsymbol{X}_2$, then $[1\ \lambda'_i\ \lambda'^2_i \cdots \lambda'^T_i]$ must lie in this row space. Hence the $(r+1) \times (T+1)$ matrix,

$$\begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^T \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^T \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_r & \lambda_r^2 & \cdots & \lambda_r^T \\ 1 & \lambda'_i & \lambda'^2_i & \cdots & \lambda'^T_i \end{bmatrix} \tag{2.18}$$

must have low row rank. However, since $r + 1 \leq T + 1$ by assumption, and all the λ_j's and λ'_i are all distinct, by Lemma 2.4.2 (2.18) must have full row rank. With this contradiction we conclude that the nonzero eigenvalues of $\boldsymbol{A}$ are unique.

For the eigenvectors of $\boldsymbol{A}$, recall from Lemma 2.4.3 that $\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Lambda}^{\intercal}$. Since $\boldsymbol{\Lambda}$ has full column-rank there is a unique solution for $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{\Lambda}^{\intercal^\dagger}$. Thus, the r eigenvectors corresponding to nonzero eigenvalues of $\boldsymbol{A}$ must be unique up to a scaling. Note that since $\boldsymbol{X}$ and $\boldsymbol{\Lambda}$ have rank r, $\boldsymbol{V}$ must also have rank r and thus have full column rank.

Note that we assumed that $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ share the same range. If they do not, since $\boldsymbol{X}_2 = \boldsymbol{A}\boldsymbol{X}_1$, there must be a component of $\boldsymbol{x}_1$ which is in the nullspace of $\boldsymbol{A}$. This results in appending an extra column $\boldsymbol{v_0}$, which is in the nullspace of $\boldsymbol{A}$, to $\boldsymbol{V}$ and an extra row $[1\ 0 \cdots 0]$ to $\boldsymbol{\Lambda}$ as in (2.16). Following the same method, we find that $T \geq r$ must be replaced with $T \geq r + 1$. $\square$

2.4.2 Uniqueness of Affine Linear Model

As we have seen in Section 2.3, DMD with centering is equivalent to an additional affine term. Thus, in addition to proving that the DMD modes are unique it also important to show uniqueness of the modes for an affine dynamical system of the form $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$. In the following Theorem, we assume that the matrix $\boldsymbol{A}$ does not contain an eigenvalue equal to 1. If 1 is an eigenvalue of $\boldsymbol{A}$, then there is an inherent ambiguity in whether this mode is an eigenvector of $\boldsymbol{A}$ or incorporated into $\boldsymbol{b}$. First, we again define some conditions for the problem to be well-posed:

Definition 2.4.7. *Suppose we have sequential time series snapshots* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1}$ *such that* $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$*. Let* $\boldsymbol{A}$ *have* r *nonzero and distinct eigenvalues,* $\lambda_1, \ldots, \lambda_r$ *and corresponding eigenvectors* $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$*. We say that the* **affine DMD problem is well-posed** *if the conditions*

1. $\boldsymbol{A}$ *does not have an eigenvalue equal to* 1*,*

2. $\boldsymbol{x}_1 - \boldsymbol{c}$ *is not orthogonal to* $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$*, and either*

3a. $T \geq r + 1$ *and* $\boldsymbol{X}_1 - \boldsymbol{c}\boldsymbol{1}^\intercal$ *and* $\boldsymbol{X}_2 - \boldsymbol{c}\boldsymbol{1}^\intercal$ *share the same range, or*

3b. $T \geq r + 2$*,*

are satisfied, where $\boldsymbol{c} = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b}$*.*

Theorem 2.4.8 (Uniqueness of Affine DMD)**.** *Suppose we have sequential time series snapshots* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1}$ *such that* $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$ *for* $j = 1, \ldots, T$*, where* $\boldsymbol{A}$ *has* r *nonzero and distinct eigenvalues,* $\lambda_1, \ldots, \lambda_r$ *and corresponding eigenvectors* $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$*. Let* $\boldsymbol{A}'$ *and* $\boldsymbol{b}'$ *be any other rank* r *matrix and vector which satisfy* $\boldsymbol{x}_{j+1} = \boldsymbol{A}'\boldsymbol{x}_j + \boldsymbol{b}'$*. If the affine DMD problem is well-posed, then* $\boldsymbol{b}' = \boldsymbol{b}$ *and* $\boldsymbol{A}'$ *has the same* r *nonzero eigenvalues* $\lambda_1, \ldots, \lambda_r$ *and corresponding eigenvectors* $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$ *as* $\boldsymbol{A}$*, and these are unique up to scaling.*

Proof. Since $\boldsymbol{A}$ does not have an eigenvalue of 1, then $\boldsymbol{I} - \boldsymbol{A}$ is invertible. Therefore, we can shift the origin in order to express $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$ as $\boldsymbol{x}_{j+1} - \boldsymbol{c} = \boldsymbol{A}(\boldsymbol{x}_j - \boldsymbol{c})$, where $\boldsymbol{c} = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b}$. By Lemma 2.4.3, we may express $\boldsymbol{X} - \boldsymbol{c}\mathbf{1}^\intercal$ as

$$\boldsymbol{X} - \boldsymbol{c}\mathbf{1}^\intercal = \boldsymbol{V}\boldsymbol{\Lambda}^\intercal. \tag{2.19}$$

Similarly for $\boldsymbol{A}'$ and $\boldsymbol{b}'$, $\boldsymbol{X} - \boldsymbol{c}'\mathbf{1}^\intercal = \boldsymbol{V}'\boldsymbol{\Lambda}'^\intercal$. Taking the difference in these equations,

$$\boldsymbol{V}\boldsymbol{\Lambda}^\intercal = \boldsymbol{V}'\boldsymbol{\Lambda}'^\intercal + (\boldsymbol{c} - \boldsymbol{c}')\mathbf{1}^\intercal.$$

First, assume that both $\boldsymbol{c} \neq \boldsymbol{c}'$ and that $\boldsymbol{A}$ and $\boldsymbol{A}'$ do not share all of their eigenvalues. (We will show that this yields a contradiction.) Without loss of generality, let λ be an eigenvalue of $\boldsymbol{A}$ but not $\boldsymbol{A}'$. Hence, $\boldsymbol{\lambda} = \begin{bmatrix}1 & \lambda & \cdots & \lambda^T\end{bmatrix}^\intercal$ is a column of $\boldsymbol{\Lambda}$ but not $\boldsymbol{\Lambda}'$. Defining, $\tilde{\boldsymbol{\Lambda}} = [\boldsymbol{\Lambda}' \ \mathbf{1}]$, then

$$\boldsymbol{V}\boldsymbol{\Lambda}^T = \begin{bmatrix}\boldsymbol{V}' & \boldsymbol{c} - \boldsymbol{c}'\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}'^\intercal \\ \mathbf{1}^\intercal\end{bmatrix} = \begin{bmatrix}\boldsymbol{V}' & \boldsymbol{c} - \boldsymbol{c}'\end{bmatrix}\tilde{\boldsymbol{\Lambda}}^\intercal.$$

Since $\boldsymbol{V}$ has full column rank, applying $\boldsymbol{V}^\dagger$ on the left to both sides yields,

$$\boldsymbol{\Lambda}^\intercal = \boldsymbol{V}^\dagger\begin{bmatrix}\boldsymbol{V}' & \boldsymbol{c} - \boldsymbol{c}'\end{bmatrix}\tilde{\boldsymbol{\Lambda}}^\intercal.$$

If we apply the orthogonal projection $\boldsymbol{I} - \tilde{\boldsymbol{\Lambda}}^{\intercal\dagger}\tilde{\boldsymbol{\Lambda}}^\intercal$ on the right to both sides, we see that the right hand side is $\mathbf{0}$. However, $\boldsymbol{\Lambda}^\intercal\left(\boldsymbol{I} - \tilde{\boldsymbol{\Lambda}}^{\intercal\dagger}\tilde{\boldsymbol{\Lambda}}^\intercal\right)$ cannot be $\mathbf{0}$ since, by assumption, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}'$ have different column spaces. To see this, consider the solution $\boldsymbol{Y}$ to $\tilde{\boldsymbol{\Lambda}}\boldsymbol{Y} = \boldsymbol{\Lambda}$. In particular, consider a single column of this equation:

$$\tilde{\boldsymbol{\Lambda}}\boldsymbol{y} = \boldsymbol{\lambda}. \tag{2.20}$$

If $\text{range}(\boldsymbol{X}_1 - \boldsymbol{c}\mathbf{1}^\intercal) = \text{range}(\boldsymbol{X}_2 - \boldsymbol{c}\mathbf{1}^\intercal)$, then using Lemma 2.4.3 we have

$$\underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ \lambda_1' & \lambda_2' & \cdots & \lambda_r' & 1 \\ \lambda_1'^{\,2} & \lambda_2'^{\,2} & \cdots & \lambda_r^{2\prime} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda_1'^{\,T} & \lambda_2'^{\,T} & \cdots & \lambda_r'^{\,T} & 1 \end{bmatrix}}_{\tilde{\boldsymbol{\Lambda}}} \boldsymbol{y} = \begin{bmatrix} 1 \\ \lambda \\ \lambda^2 \\ \vdots \\ \lambda^T \end{bmatrix}.$$

By Lemma 2.4.2, if $T+1 \geq r+2$, then since $\lambda, \lambda'_1, \ldots, \lambda'_r, 1$ are all distinct, $\boldsymbol{\lambda}$ cannot be expressed as a linear combination of the columns of $\tilde{\boldsymbol{\Lambda}}^\intercal$. If $\text{range}(\boldsymbol{X}_1 - \boldsymbol{c}\boldsymbol{1}^\intercal) \neq \text{range}(\boldsymbol{X}_2 - \boldsymbol{c}\boldsymbol{1}^\intercal)$, then we must append an extra column to $\tilde{\boldsymbol{\Lambda}}$, and the condition in this case is $T+1 \geq r+3$. Thus, there does not exist a solution for $\boldsymbol{y}$ in (2.20), and consequently there does not exist a solution for $\boldsymbol{Y}$. Thus, $\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{\Lambda}}^\dagger\boldsymbol{\Lambda} \neq \boldsymbol{\Lambda}$. Taking the transpose and rearranging we have

$$\boldsymbol{\Lambda}^\intercal \left(\boldsymbol{I} - \tilde{\boldsymbol{\Lambda}}^{\intercal^\dagger}\tilde{\boldsymbol{\Lambda}}^\intercal\right) \neq \boldsymbol{0}.$$

This yields a contradiction, from which we conclude that either $\boldsymbol{c} = \boldsymbol{c}'$, or $\boldsymbol{A}$ and $\boldsymbol{A}'$ have the same nonzero eigenvalues, or both.

First, consider the case where $\boldsymbol{c} = \boldsymbol{c}'$. This implies that $\boldsymbol{V}'\boldsymbol{\Lambda}'^\intercal = \boldsymbol{V}\boldsymbol{\Lambda}^\intercal$, and using the same logic as Theorem 2.4.6, then the nonzero eigenvalues of $\boldsymbol{A}'$ and $\boldsymbol{A}$ are equal.

Next, suppose that the nonzero eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{A}'$ are equal. Thus, $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}'$ and

$$\left(\boldsymbol{V} - \boldsymbol{V}'\right)\boldsymbol{\Lambda}^\intercal = \left(\boldsymbol{c} - \boldsymbol{c}'\right)\boldsymbol{1}^\intercal.$$

Applying $\boldsymbol{I} - \boldsymbol{\Lambda}^{\intercal^\dagger}\boldsymbol{\Lambda}^\intercal$ to both sides,

$$\boldsymbol{0} = \left(\boldsymbol{c} - \boldsymbol{c}'\right)\boldsymbol{1}^\intercal\left(\boldsymbol{I} - \boldsymbol{\Lambda}^{\intercal^\dagger}\boldsymbol{\Lambda}^\intercal\right),$$

and note that the right hand side is an $n \times (T+1)$ rank-1 matrix. Now, $\boldsymbol{1}^\intercal\left(\boldsymbol{I} - \boldsymbol{\Lambda}^{\intercal^\dagger}\boldsymbol{\Lambda}^\intercal\right) \neq \boldsymbol{0}$, since $\boldsymbol{1}$ is not in the column space of $\boldsymbol{\Lambda}$. Thus, $\boldsymbol{c} = \boldsymbol{c}'$.

In either case we have that both $\boldsymbol{c} = \boldsymbol{c}'$ and $\boldsymbol{\Lambda}' = \boldsymbol{\Lambda}$. From $\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Lambda}^\intercal + \boldsymbol{c}\boldsymbol{1}^\intercal$, we see that $\boldsymbol{V}$ is the unique solution, $\boldsymbol{V} = \left(\boldsymbol{X} - \boldsymbol{c}\boldsymbol{1}^\intercal\right)\boldsymbol{\Lambda}^{\intercal^\dagger}$. This implies that the eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$ of $\boldsymbol{A}$, which correspond to the columns of $\boldsymbol{V}$, are unique up to scaling. ☐

2.5 Comparison of DMD with Centering to DMD without Centering

In this section, we show that, for linear systems (dynamics generated by $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$), both DMD with centering and without centering can be used to compute the modes of $\boldsymbol{A}$. In particular, DMD with and without centering will yield the same modes, except for the background mode. For DMD without centering, this background mode corresponds to an eigenvalue equal to 1, while for DMD with centering, this is replaced by an eigenvalue equal to 0 (see Theorem 2.5.2).

For affine systems (dynamics generated by $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$), DMD with centering can be used to extract $\boldsymbol{b}$ and the modes of $\boldsymbol{A}$. In some cases, DMD without centering can also be used to compute the modes of $\boldsymbol{A}$ and model the dynamics of the system, but in many cases it cannot, most notably if $\boldsymbol{A}$ is full-rank. Here we provide necessary and sufficient conditions for when DMD will and will not be able to successfully model the dynamics.

We then illustrate these results with synthetic examples in Section 2.5.3, and show that these results generalize to the case of measurement noise in Section 2.5.4.

2.5.1 Linear Systems without Bias

Consider a set of snapshots $\boldsymbol{x}_j$ which satisfy $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$. From Theorem 2.4.6, we know that DMD without centering can be used to extract the nonzero eigenvalues and eigenvectors of $\boldsymbol{A}$. We now show that DMD with centering can also be used to compute the same modes. In particular, if 1 is not an eigenvalue of $\boldsymbol{A}$, DMD with and without centering yield the same DMD modes (Theorem 2.5.2). If one is an eigenvalue of $\boldsymbol{A}$, then DMD with and without centering will share the same modes, except for the background mode. For DMD without centering this mode corresponds to an eigenvalue equal to one, while for DMD with centering this is replaced with an eigenvalue equal to zero.

We first prove a useful lemma.

Lemma 2.5.1. *Suppose we have sequential time series such that* $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$*, and the DMD problem is well-posed. Then* $\boldsymbol{A}$ *has an eigenvalue equal to* 1 *if and only if* $\mathbf{1}^\intercal \boldsymbol{X}_1^\dagger \boldsymbol{X} = \mathbf{1}^\intercal$.

Proof. Let $\boldsymbol{X}_1$ have rank r.[1] By Lemma 2.4.3, we may decompose $\boldsymbol{X}_1$ into the product of two matrices $\boldsymbol{V}$ and $\boldsymbol{\Lambda}$ which have full column rank.

---

[1] In general, we define r to be the rank of $\boldsymbol{A}$. In many cases, assuming the DMD problem is well-posed, $\mathrm{rank}(\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{X}_1)$. However, if $\boldsymbol{x}_1$ has a component in the nullspace of $\boldsymbol{A}$, then $\mathrm{rank}(\boldsymbol{X}_1) = \mathrm{rank}(\boldsymbol{A}) + 1$ and one of the λ_j's will be 0.

$$\boldsymbol{X}_1 = \underbrace{\begin{bmatrix} | & | & & | \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_r \\ | & | & & | \end{bmatrix}}_{\boldsymbol{V}} \underbrace{\begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_r^{T-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{T-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_r & \lambda_r^2 & \cdots & \lambda_r^{T-1} \end{bmatrix}}_{\boldsymbol{\Lambda}^\intercal} \tag{2.21}$$

Thus, $\boldsymbol{V}^\dagger \boldsymbol{V} = \boldsymbol{I}$ and

$$\boldsymbol{X}_1^\dagger \boldsymbol{X}_1 = \boldsymbol{\Lambda}^{\dagger\intercal} \boldsymbol{V}^\dagger \boldsymbol{V} \boldsymbol{\Lambda}^\intercal = \boldsymbol{\Lambda}^{\dagger\intercal} \boldsymbol{\Lambda}^\intercal = \boldsymbol{\Lambda}(\boldsymbol{\Lambda}^\intercal \boldsymbol{\Lambda})^{-1} \boldsymbol{\Lambda}^\intercal = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\dagger.$$

First, suppose that 1 is an eigenvalue of $\boldsymbol{A}$. Without loss of generality, let $\lambda_1 = 1$. Consider the solution $\boldsymbol{\beta}$ to the equation $\boldsymbol{\Lambda}\boldsymbol{\beta} = \mathbf{1}$, or more explicitly,

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \lambda_2 & \cdots & \lambda_r \\ 1 & \lambda_2^2 & \cdots & \lambda_r^2 \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_2^{T-1} & \cdots & \lambda_k^{T-1} \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{2.22}$$

Clearly, there exists a solution for $\boldsymbol{\beta}$, namely $\boldsymbol{\beta} = [1\ 0 \cdots 0]^\intercal$. Since the DMD problem is well posed, the columns of $\boldsymbol{\Lambda}$ are linearly independent. This implies that the solution for $\boldsymbol{\beta}$ is unique, and hence $\boldsymbol{\beta} = \boldsymbol{\Lambda}^\dagger \mathbf{1} = [1\ 0 \cdots 0]^\intercal$ and $\boldsymbol{X}_1^\dagger \boldsymbol{X}_1 \mathbf{1} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\dagger \mathbf{1} = \mathbf{1}$. Since $\boldsymbol{X}_1^\dagger \boldsymbol{X}_1$ is symmetric $\mathbf{1}^\intercal = \mathbf{1}^\intercal \boldsymbol{X}_1^\dagger \boldsymbol{X}_1$. To conclude the proof for this direction we must show that $\mathbf{1}^\intercal \boldsymbol{X}_1^\dagger \boldsymbol{x}_{T+1} = 1$. We can express $\boldsymbol{x}_{T+1}$ as

$$\boldsymbol{x}_{T+1} = \begin{bmatrix} | & | & & | \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_r \\ | & | & & | \end{bmatrix} \underbrace{\begin{bmatrix} 1 \\ \lambda_2^T \\ \vdots \\ \lambda_r^T \end{bmatrix}}_{\boldsymbol{\lambda}_T}.$$

Plugging this in,

$$\mathbf{1}^\intercal \boldsymbol{X}_1^\dagger \boldsymbol{x}_{T+1} = \mathbf{1}^\intercal \boldsymbol{\Lambda}^{\dagger\intercal} \boldsymbol{V}^\dagger \boldsymbol{V} \boldsymbol{\lambda}_T = \mathbf{1}^\intercal \boldsymbol{\Lambda}^{\dagger\intercal} \boldsymbol{\lambda}_T = \boldsymbol{\beta}^\intercal \boldsymbol{\lambda}_{T+1} = 1.$$

Combining this fact with $\mathbf{1}^\intercal = \mathbf{1}^\intercal \boldsymbol{X}_1^\dagger \boldsymbol{X}_1$ implies that $\mathbf{1}^\intercal = \mathbf{1}^\intercal \boldsymbol{X}_1^\dagger \boldsymbol{X}$.

Now suppose 1 is not an eigenvalue of $\boldsymbol{A}$. Similar to (2.22), we consider the equation,

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_r \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_r^2 \\ \vdots & \vdots & \cdots & \vdots \\ \lambda_1^{T-1} & \lambda_2^{T-1} & \cdots & \lambda_r^{T-1} \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

In this case, all of the λ_i's are distinct and not equal to 1. By Lemma 2.4.2, since $T-1 \geq r$, $\mathbf{1}$ is not in the span of the columns of $\boldsymbol{\Lambda}$ and hence $\boldsymbol{\Lambda}\boldsymbol{\beta} \neq \mathbf{1}$ for any value of $\boldsymbol{\beta}$. Thus, $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\dagger \mathbf{1} \neq \mathbf{1}$ and therefore $\mathbf{1}^\intercal \boldsymbol{X}_1 \boldsymbol{X}_1^\dagger \neq \mathbf{1}^\intercal$. □

Theorem 2.5.2 (DMD with and without Centering for Linear Systems)**.** *Suppose we have sequential time series snapshots $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1}$ such that $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$ for $j = 1, \ldots, T$. If the DMD problem is well-posed, then the following holds for $\hat{\boldsymbol{A}} = \boldsymbol{X}_2 \boldsymbol{X}_1^\dagger$:*

1. *If $\hat{\boldsymbol{A}}$ has an eigenvalue equal to 1, then $\bar{\boldsymbol{A}} = \bar{\boldsymbol{X}}_2 \bar{\boldsymbol{X}}_1^\dagger$ will have the same eigenvalues and corresponding eigenvectors as $\hat{\boldsymbol{A}}$, except the 1 eigenvalue is replaced with a 0 eigenvalue.*
2. *If $\hat{\boldsymbol{A}}$ does not have an eigenvalue equal to 1, then $\bar{\boldsymbol{A}}$ will have the same eigenvalues and corresponding eigenvectors as $\hat{\boldsymbol{A}}$.*

Proof. DMD with centering obtains the centered matrix

$$\bar{\boldsymbol{A}} = \bar{\boldsymbol{X}}_2 \bar{\boldsymbol{X}}_1^\dagger = (\boldsymbol{X}_2 - \boldsymbol{\mu}_2 \mathbf{1}^\intercal)(\boldsymbol{X}_1 - \boldsymbol{\mu}_1 \mathbf{1}^\intercal)^\dagger. \tag{2.23}$$

For part 1, suppose that $\hat{\boldsymbol{A}}$ has an eigenvalue equal to 1. Then we have that $(\boldsymbol{I} - \boldsymbol{X}_1^\dagger \boldsymbol{X}_1)^\intercal \mathbf{1} = \mathbf{0}$ by Lemma 2.5.1. Applying the rank one update formula in [157], which is a generalization of the Sherman-Morrison-Woodbury formula [191] to the case of non-invertible matrices, then

$$(\boldsymbol{X}_1 - \boldsymbol{\mu}_1 \mathbf{1}^\intercal)^\dagger = \boldsymbol{X}_1^\dagger \left(\boldsymbol{I} - \frac{\boldsymbol{n}\boldsymbol{n}^\intercal}{\boldsymbol{n}^\intercal \boldsymbol{n}} \right), \tag{2.24}$$

where $\boldsymbol{n} = \boldsymbol{X}_1^{\dagger\intercal}\boldsymbol{1}$ (see Appendix A.1). Plugging (2.24) into (2.23)

$$\begin{aligned}
\bar{\boldsymbol{A}} &= (\boldsymbol{X}_2 - \boldsymbol{\mu}_2\boldsymbol{1}^\intercal)\,\boldsymbol{X}_1^\dagger\left(\boldsymbol{I} - \frac{\boldsymbol{n}\boldsymbol{n}^\intercal}{\boldsymbol{n}^\intercal\boldsymbol{n}}\right) \\
&= \boldsymbol{X}_2\boldsymbol{X}_1^\dagger - \boldsymbol{\mu}_2\boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger - \boldsymbol{X}_2\boldsymbol{X}_1^\dagger\frac{\boldsymbol{n}\boldsymbol{n}^\intercal}{\boldsymbol{n}^\intercal\boldsymbol{n}} + \boldsymbol{\mu}_2\boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger\frac{\boldsymbol{n}\boldsymbol{n}^\intercal}{\boldsymbol{n}^\intercal\boldsymbol{n}} \\
&= \boldsymbol{X}_2\boldsymbol{X}_1^\dagger - \boldsymbol{\mu}_2\boldsymbol{n}^\intercal - \boldsymbol{X}_2\boldsymbol{X}_1^\dagger\frac{\boldsymbol{n}\boldsymbol{n}^\intercal}{\boldsymbol{n}^\intercal\boldsymbol{n}} + \boldsymbol{\mu}_2\boldsymbol{n}^\intercal \\
&= \hat{\boldsymbol{A}}\left(\boldsymbol{I} - \frac{\boldsymbol{X}_1^{\dagger\intercal}\boldsymbol{1}\boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger}{\left\|\boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger\right\|^2}\right).
\end{aligned}$$

Again applying Lemma 2.5.1, $\boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger\hat{\boldsymbol{A}} = \boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger$, i.e. $\boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger$ is a left eigenvector of $\hat{\boldsymbol{A}}$ with eigenvalue 1. By Theorem 2.1 in [52], we conclude that $\bar{\boldsymbol{A}}$ shares all the same eigenvalues $\hat{\boldsymbol{A}}$, except the eigenvalue of 1 is replaced with 0.

For the eigenvectors, first note that by Lemma 2.4.3 we can express $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, and $\hat{\boldsymbol{A}}$, in terms of the eigenvectors of $\hat{\boldsymbol{A}}$,

$$\boldsymbol{X}_1 = \boldsymbol{V}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{T-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \lambda_r & \lambda_r^2 & \cdots & \lambda_r^{T-1} \end{bmatrix}, \; \boldsymbol{X}_2 = \boldsymbol{V}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_1^2 & \lambda_1^3 & \cdots & \lambda_1^{T} \\ \vdots & \vdots & \vdots & & \vdots \\ \lambda_r & \lambda_r^2 & \lambda_r^3 & \cdots & \lambda_r^{T} \end{bmatrix},$$

and

$$\hat{\boldsymbol{A}} = \boldsymbol{V}\begin{bmatrix} 1 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_r \end{bmatrix}\boldsymbol{V}^\dagger.$$

In this basis, the mean-subtracted data are

$$\bar{\boldsymbol{X}}_1 = \frac{\boldsymbol{V}}{T}\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 - \sum_{i=0}^{T-1}\lambda_1^i & \lambda_1 - \sum_{i=0}^{T-1}\lambda_1^i & \lambda_1^2 - \sum_{i=0}^{T-1}\lambda_1^i & \cdots & \lambda_1^{T-1} - \sum_{i=0}^{T-1}\lambda_1^i \\ \vdots & \vdots & \vdots & & \vdots \\ 1 - \sum_{i=0}^{T-1}\lambda_r^i & \lambda_r - \sum_{i=0}^{T-1}\lambda_1^i & \lambda_r^2 - \sum_{i=0}^{T-1}\lambda_1^i & \cdots & \lambda_r^{T-1} - \sum_{i=0}^{T-1}\lambda_1^i \end{bmatrix}$$

$$\bar{\boldsymbol{X}}_2 = \frac{\boldsymbol{V}}{T} \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \lambda_1 - \sum_{i=1}^{T} \lambda_1^i & \lambda_1^2 - \sum_{i=1}^{T} \lambda_1^i & \lambda_1^3 - \sum_{i=1}^{T} \lambda_1^i & \cdots & \lambda_1^T - \sum_{i=1}^{T} \lambda_1^i \\ \vdots & \vdots & \vdots & & \vdots \\ \lambda_r - \sum_{i=1}^{T} \lambda_r^i & \lambda_r^2 - \sum_{i=1}^{T} \lambda_1^i & \lambda_r^3 - \sum_{i=1}^{T} \lambda_1^i & \cdots & \lambda_r^T - \sum_{i=1}^{T} \lambda_1^i \end{bmatrix}.$$

We immediately see that

$$\boldsymbol{A}' = \boldsymbol{V} \begin{bmatrix} 0 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_r \end{bmatrix} \boldsymbol{V}^\dagger$$

satisfies $\bar{\boldsymbol{X}}_2 = \boldsymbol{A}'\bar{\boldsymbol{X}}_1$. By Theorem 2.4.6, the nonzero eigenvalues and corresponding eigenvectors of $\boldsymbol{A}'$ must be the same as those of $\bar{\boldsymbol{A}}$. We conclude that the eigenvectors corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_r$ of $\hat{\boldsymbol{A}}$ and $\bar{\boldsymbol{A}}$ must be equal up to scaling.

For part 2, suppose 1 is not an eigenvalue of $\hat{\boldsymbol{A}}$. Like before, we can explicitly compute $\bar{\boldsymbol{A}}$ as a rank one update to $\hat{\boldsymbol{A}}$ (Appendix A.1). Since 1 is not an eigenvalue of $\hat{\boldsymbol{A}}$, then by Lemma 2.5.1, $\left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger \boldsymbol{X}_1\right)^\intercal \boldsymbol{1} \neq \boldsymbol{0}$. We know that, since the data are linearly consistent, then the solution $\hat{\boldsymbol{A}}$ to DMD without centering satisfies $\boldsymbol{X}_2 = \hat{\boldsymbol{A}}\boldsymbol{X}_1$, and thus $\boldsymbol{X}_2\left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger \boldsymbol{X}_1\right) = 0$. Now,

$$\begin{aligned} \hat{\boldsymbol{A}} - \bar{\boldsymbol{A}} &= \left(\boldsymbol{X}_2\boldsymbol{X}_1^\dagger - \bar{\boldsymbol{X}}_2\bar{\boldsymbol{X}}_1^\dagger\right) \\ &= \left(\boldsymbol{X}_2\boldsymbol{X}_1^\dagger - \boldsymbol{X}_2\left(\boldsymbol{I} - \frac{\boldsymbol{1}\boldsymbol{1}^\intercal}{\boldsymbol{1}^\intercal\boldsymbol{1}}\right)\left(\boldsymbol{I} - \frac{\left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger\boldsymbol{X}_1\right)\boldsymbol{1}\boldsymbol{1}^\intercal}{\boldsymbol{1}^\intercal\left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger\boldsymbol{X}_1\right)\boldsymbol{1}}\right)\boldsymbol{X}_1^\dagger\right) \\ &= \left(\boldsymbol{X}_2\left(\frac{\left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger\boldsymbol{X}_1\right)\boldsymbol{1}\boldsymbol{1}^\intercal}{\boldsymbol{1}^\intercal\left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger\boldsymbol{X}_1\right)\boldsymbol{1}}\right)\boldsymbol{X}_1^\dagger\right) \\ &= \boldsymbol{0}. \end{aligned} \tag{2.25}$$

□

2.5.2 Linear Systems with Bias

We will now illustrate what can go wrong applying DMD without centering to data generated by a linear system with bias. Consider the system

$$\boldsymbol{x}_{j+1} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \boldsymbol{x}_j + \begin{bmatrix} 1 \\ 2 \end{bmatrix},$$

with $\boldsymbol{x}_1 = [1, 1]^\intercal$. This yields data matrices,

$$\boldsymbol{X}_1 = \begin{bmatrix} 1 & 3 & 7 \\ 1 & 5 & 17 \end{bmatrix} \text{ and } \boldsymbol{X}_2 = \begin{bmatrix} 3 & 7 & 15 \\ 5 & 17 & 53 \end{bmatrix}.$$

It can easily be shown that these data are linearly inconsistent, since $\boldsymbol{X}_2 \left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger \boldsymbol{X}_1 \right) \neq \boldsymbol{0}$, and so DMD without centering cannot accurately model the data. In particular, the computed eigenvalues and eigenvectors will not correspond to those from the affine system and the reconstruction of the data using the DMD model will be poor.

However, for some cases DMD without centering may be able to model data generated by a linear system with bias, i.e. affine dynamics. In Theorem 2.5.3, we present necessary and sufficient conditions for when this is possible. These boil down to (1) having "extra rank" available to capture the bias with an eigenvector of eigenvalue 1 and (2) a technical condition on the fixed point $\boldsymbol{c}$ and the eigenvectors of $\boldsymbol{A}$ as captured in $\boldsymbol{V}$.

Theorem 2.5.3 (DMD without Centering for Affine Systems)**.** *Consider data which satisfy the recursive affine equation* $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$ *for* $j = 1, \ldots, T$*, such that the affine problem is well-posed. Suppose* $\boldsymbol{A}$ *does not have an eigenvalue equal to* 1*, and define the fixed point* $\boldsymbol{c} = (\boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{b}$*. Like in* (2.19)*, we may factor the data* $\boldsymbol{X}$ *as*

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{V} & \boldsymbol{c} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}^\intercal \\ \boldsymbol{1}^\intercal \end{bmatrix}.$$

Then there exists an $\boldsymbol{A}'$ *such that* $\boldsymbol{x}_{j+1} = \boldsymbol{A}'\boldsymbol{x}_j$ *if and only if* $\boldsymbol{c}$ *is not in the span of the columns of* $\boldsymbol{V}$*.*

Corollary 2.5.3.1. *If $\boldsymbol{A}$ is full rank and has distinct eigenvalues, then DMD without centering will not be able to accurately represent the dynamics. In particular, the eigenvalues and eigenvectors of DMD without centering will not correspond to the eigenvalues and eigenvectors of $\boldsymbol{A}$ and the reconstruction of the data using this linear model will be poor.*

Proof. Suppose that $\boldsymbol{c}$ is not in the span of the columns of $\boldsymbol{V}$. We would like to show that

$$\boldsymbol{X} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Lambda}}^{\intercal}, \tag{2.26}$$

where $\tilde{\boldsymbol{V}}$ and $\tilde{\boldsymbol{\Lambda}}$ have full column rank and $\tilde{\boldsymbol{\Lambda}}$ is a rectangular Vandermonde matrix. Define $\tilde{\boldsymbol{V}} = [\boldsymbol{V}\ \boldsymbol{c}]$ and $\tilde{\boldsymbol{\Lambda}} = [\boldsymbol{\Lambda}\ \boldsymbol{1}]$. Then $\boldsymbol{X}$ satisfies (2.26). By assumption, the columns of $\tilde{\boldsymbol{V}}$ are linearly independent, and, since 1 is not an eigenvalue of $\boldsymbol{A}$, by Lemma 2.4.2, the columns of $\tilde{\boldsymbol{\Lambda}}$ are linearly independent. With this factorization, $\boldsymbol{X}$ satisfies a linear model $\boldsymbol{A}' = \tilde{\boldsymbol{V}}\mathrm{diag}(\lambda_1, \ldots, \lambda_r, 1)\tilde{\boldsymbol{V}}^{\dagger}$. By Theorem 2.4.6, reading off from $\tilde{\boldsymbol{\Lambda}}$, the modes of DMD without centering will be the same eigenvalues and corresponding eigenvectors of DMD with centering with an additional eigenvalue equal to 1.

Now suppose that $\boldsymbol{c}$ is in the span of the columns of $\boldsymbol{V}$. To show that a linear system cannot model the data, we will use proof by contradiction. Suppose that we can express

$$[\boldsymbol{V}\ \boldsymbol{c}][\boldsymbol{\Lambda}\ \boldsymbol{1}]^{\intercal} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Lambda}}^{\intercal},$$

where $\tilde{\boldsymbol{V}}$ and $\tilde{\boldsymbol{\Lambda}}$ have full column rank (Lemma 2.4.3). Since $\mathrm{rank}([\boldsymbol{V}\ \boldsymbol{c}]) < \mathrm{rank}([\boldsymbol{\Lambda}\ \boldsymbol{1}])$, this requires that $\mathrm{rank}(\tilde{\boldsymbol{\Lambda}}) < \mathrm{rank}([\boldsymbol{\Lambda}\ \boldsymbol{1}])$.

For a Vandermonde matrix $\boldsymbol{M}$, let $\sigma(\boldsymbol{M})$ denote its "spectrum," i.e. the eigenvalues that generate the columns, so that $\sigma(\boldsymbol{\Lambda}) = \{\lambda_1, \ldots, \lambda_r\}$. We have three cases:

Case 1: $\sigma(\tilde{\boldsymbol{\Lambda}})$ is a proper subset of $\sigma([\boldsymbol{\Lambda}\ \boldsymbol{1}])$.

In this case, we can partition the columns of $[\boldsymbol{\Lambda}\ \boldsymbol{1}]$ to form two Vandermonde matrices $\tilde{\boldsymbol{\Lambda}}$ and $\boldsymbol{\Lambda}'$, where the spectrum of $\sigma(\boldsymbol{\Lambda}') = \sigma([\boldsymbol{\Lambda}\ \boldsymbol{1}]) \setminus \sigma(\tilde{\boldsymbol{\Lambda}})$. Employing the same partition on the columns of $[\boldsymbol{V}\ \boldsymbol{c}]$, we can form matrices $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, such that

$$\begin{bmatrix} \boldsymbol{V}_1 & \boldsymbol{V}_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Lambda}}^{\intercal} \\ \boldsymbol{\Lambda}'^{\intercal} \end{bmatrix} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Lambda}}^{\intercal}. \tag{2.27}$$

Consider the solution to

$$\boldsymbol{B}\begin{bmatrix}\tilde{\boldsymbol{\Lambda}}^\intercal \\ \boldsymbol{\Lambda}'^\intercal\end{bmatrix} = \tilde{\boldsymbol{\Lambda}}^\intercal.$$

By assumption, the columns of $[\tilde{\boldsymbol{\Lambda}}\ \boldsymbol{\Lambda}']$ are linearly independent, and so there is a unique solution $\boldsymbol{B} = \tilde{\boldsymbol{\Lambda}}^\intercal \begin{bmatrix}\tilde{\boldsymbol{\Lambda}}^\intercal \\ \boldsymbol{\Lambda}'^\intercal\end{bmatrix}^\dagger = [\boldsymbol{I}\ \boldsymbol{0}]$. Multiplying both sides of (2.27) on the right by $\begin{bmatrix}\tilde{\boldsymbol{\Lambda}}^\intercal \\ \boldsymbol{\Lambda}'^\intercal\end{bmatrix}^\dagger$, we find

$$\begin{bmatrix}\boldsymbol{V}_1 & \boldsymbol{V}_2\end{bmatrix} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Lambda}}^\intercal \begin{bmatrix}\tilde{\boldsymbol{\Lambda}}^\intercal \\ \boldsymbol{\Lambda}'^\intercal\end{bmatrix}^\dagger = \begin{bmatrix}\tilde{\boldsymbol{V}} & \boldsymbol{0}\end{bmatrix}.$$

However, $\boldsymbol{V}_2$ cannot be $\boldsymbol{0}$, since it comes from a partition of the nonzero columns of $[\boldsymbol{V}\ \boldsymbol{c}]$.

Case 2: $\sigma(\tilde{\boldsymbol{\Lambda}}) = \sigma([\boldsymbol{\Lambda}\ \boldsymbol{1}])$.

The two spectra cannot be equal, since this would imply that $\operatorname{rank}(\tilde{\boldsymbol{\Lambda}}) = \operatorname{rank}([\boldsymbol{\Lambda}\ \boldsymbol{1}])$.

Case 3: There exists some $\lambda \in \sigma(\tilde{\boldsymbol{\Lambda}})$ with $\lambda \notin \sigma([\boldsymbol{\Lambda}\ \boldsymbol{1}])$.

Therefore, $\tilde{\boldsymbol{\Lambda}}$ contains a column which is not in the column space of $[\boldsymbol{\Lambda}\ \boldsymbol{1}]$. This means that there is a column in $\tilde{\boldsymbol{\Lambda}}$ which is linearly independent from the columns of $[\boldsymbol{\Lambda}\ \boldsymbol{1}]$. Thus, there does not exist a linear combination $\boldsymbol{C}$ such that

$$\boldsymbol{C}\begin{bmatrix}\boldsymbol{\Lambda}^\intercal \\ \boldsymbol{1}^\intercal\end{bmatrix} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Lambda}}^\intercal,$$

which is a contradiction.

□

2.5.3 Synthetic Examples

To review our results so far, we compare the eigenvalue spectra from DMD with centering and DMD without centering for four sets of measurements of affine systems $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$. The results are shown in Figure 2.2. The two top spectra correspond to data with $n < T$, while the bottom two correspond to $n > T$.

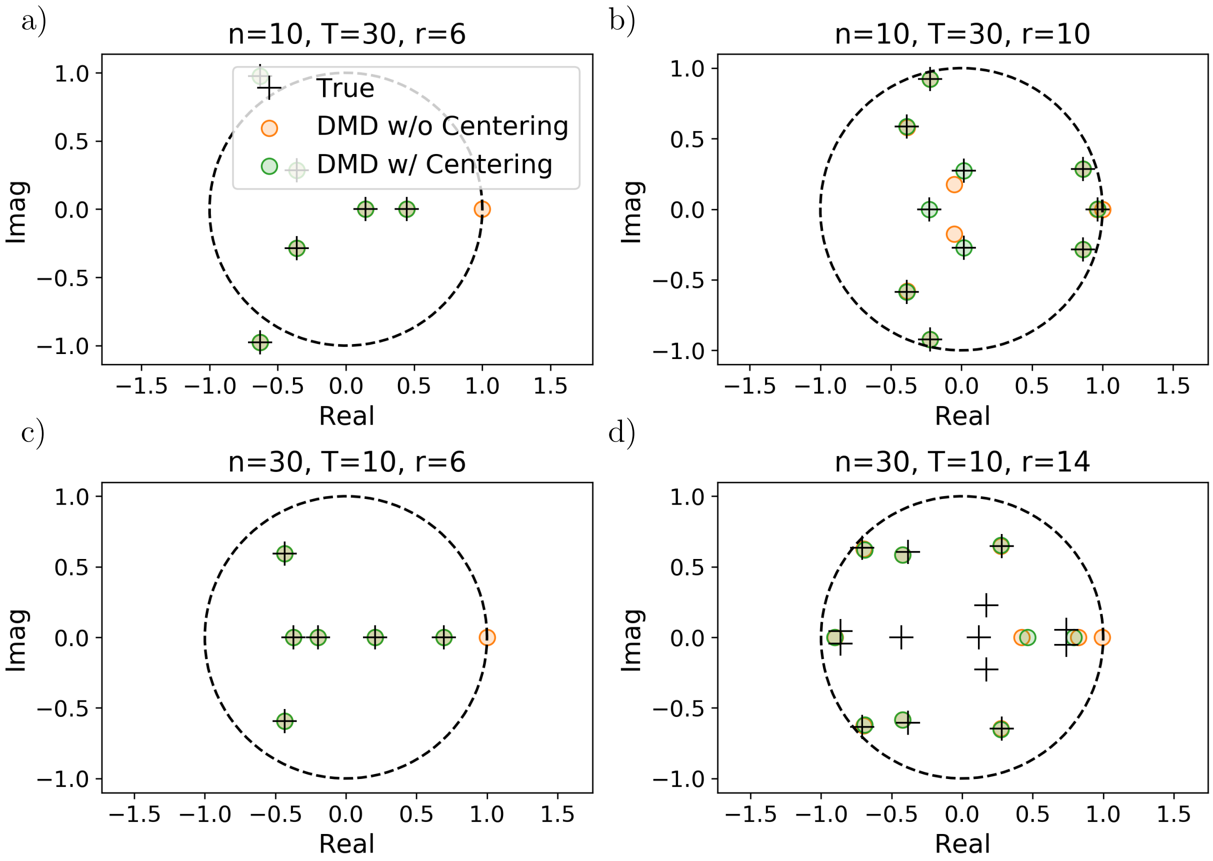


Figure 2.2: Comparison of the eigenvalues from DMD with centering (green) and DMD without centering (orange) to the true eigenvalues (black) of $\boldsymbol{A}$ for four different affine systems $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$. a) $n < T$ and $\boldsymbol{A}$ is low rank. DMD with and without centering both yield the eigenvalues of $\boldsymbol{A}$, except DMD with centering has an extra eigenvalue of 1, corresponding to the background mode. b) $n < T$ and $\boldsymbol{A}$ is full rank. DMD with centering yields the true eigenvalues of $\boldsymbol{A}$ while DMD without centering does not. c) $n > T$ and $\boldsymbol{A}$ is low rank. This yields the same result as a). d) Since $T < r$, the DMD problem is not well-posed and neither DMD with centering nor DMD without centering yields the eigenvalues of $\boldsymbol{A}$.

For $n < T$ if $\boldsymbol{A}$ is low rank ($r < n$), then DMD without centering has the same spectra as DMD with centering, but with an additional eigenvalue equal to 1. If $\boldsymbol{A}$ is full rank ($r = n$), DMD with centering computes the correct modes. However, by Corollary 2.5.3.1 DMD without centering cannot accommodate the affine term and yields incorrect eigenvalues and a poor one step reconstruction of $\left\|\boldsymbol{X}_2 - \hat{\boldsymbol{A}}\boldsymbol{X}_1\right\| = 0.019$.

For $n > T$ with $\boldsymbol{A}$ low rank, DMD without centering has the same spectrum as DMD with centering, but with an additional eigenvalue equal to 1. If $r > T$, by Theorem 2.4.6 the system is undersampled then the DMD problem is not well-posed and the modes of $\boldsymbol{A}$ are not unique. Consequently, the DMD modes for both with and without centering do not equal the true modes of $\boldsymbol{A}$. That being said, since the data are linearly consistent, all of the models are able to reconstruct the data.

2.5.4 The Effects of Noise

In Theorem 2.5.2 we showed that, for a linear system, DMD with centering and DMD without centering will yield the same modes, except that the constant mode without centering is replaced with a zero mode. However, one of the key assumptions in our proofs is that there exists $\boldsymbol{A}$, so that $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$. For real data with measurement noise, this assumption will not hold. We find empirically that these predictions do hold true, with uncertainty on the order of the noise level.

We simulated data $\boldsymbol{Y} = \boldsymbol{X} + \eta\boldsymbol{Z}$, where the elements of $\boldsymbol{Z}$ are from a standard normal distribution and perform DMD. We performed this for 500 instantiations of $\boldsymbol{Z}$ for each of 20 values of η ranging uniformly on a logarithmic scale from 10^{-9} to 1. Values of $n = 10$, $T = 30$, and $r = 7$ were used. For both DMD with centering and DMD without centering, we compute the sum of the distances from the computed eigenvalues of $\bar{\boldsymbol{A}}$ and $\hat{\boldsymbol{A}}$ to the nearest true eigenvalue of $\boldsymbol{A}$, then report the median.

Our results are shown in Figure 2.3. The eigenvalue distances, shown at top, for $\hat{\boldsymbol{A}}$ and $\bar{\boldsymbol{A}}$ scale linearly with η. These distances are very close for these two methods. Note that, when computing the sums for DMD without centering, we exclude the eigenvalue closest to one to establish a fairer comparison between these two methods.

For a specific example, in the bottom of Figure 2.3 we plot the eigenvalues computed using these two methods for $\eta = 0.005$ over 100 realizations of the noise. The black crosses show the true eigenvalues. As expected, the deviations of the eigenvalues from the true values are roughly the same. Note the presence of the additional eigenvalue equal to 1 for DMD without centering.

2.6 DMD with Centering is not a Temporal Discrete Fourier Transform

Similar to DMD, the temporal discrete Fourier transform (temporal DFT) can be used to decompose the times series data $\boldsymbol{x}_1, \ldots \boldsymbol{x}_{T+1}$ into linear combinations of modes with exponential time dependence. In particular, the temporal DFT is defined as [48, 81],

$$\hat{\boldsymbol{x}}_j := \left\{ \frac{1}{T+1} \sum_{k=1}^{T+1} \exp\left(-\frac{2\pi i (j-1)(k-1)}{T+1} \right) \boldsymbol{x}_k \right\},$$

with inverse transform

$$\boldsymbol{x}'_k := \left\{ \sum_{k=1}^{T+1} \exp\left(\frac{2\pi i (j-1)(k-1)}{T+1} \hat{\boldsymbol{x}}_k \right) \right\}. \tag{2.28}$$

In [43], Chen argues that when subtracting the mean $\boldsymbol{\mu} = \frac{1}{T+1} \sum_{j=1}^{T+1} \boldsymbol{x}_j$ the eigenvalues l_j of the companion matrix $\boldsymbol{C}$ are independent of the data:

$$l_j = \exp\left(\frac{2\pi i j}{T+1} \right) \; j = 1, \ldots, T+1.$$

and the eigenvectors $\boldsymbol{w}_j$ are given by

$$\boldsymbol{x}_k = \sum_{j=2}^{T+1} \exp\left(\frac{2\pi i (j-1)(k-1)}{T+1} \right) \boldsymbol{w}_j.$$

Comparing this to (2.28) we see that the eigenvectors of the companion matrix correspond to those given by the temporal DFT. Most significantly, this has the unintended consequence of restricting the eigenvalues to be roots of unity.

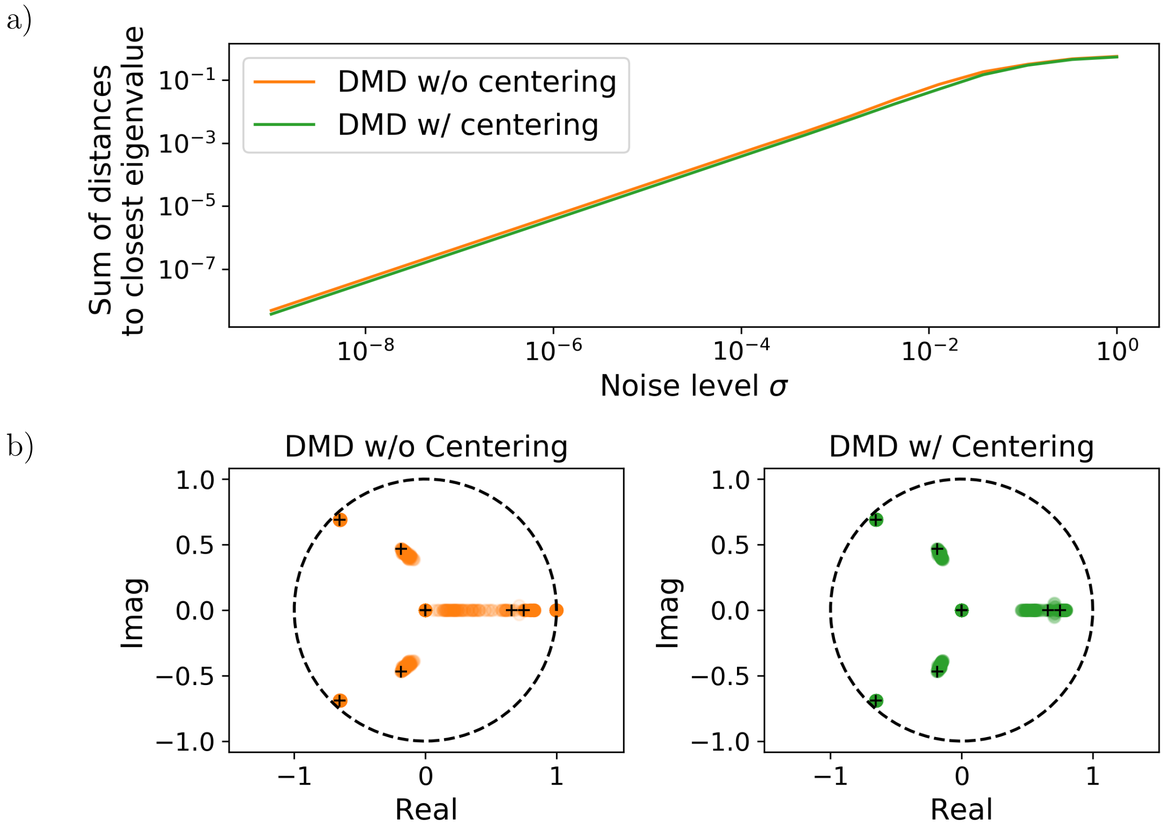


Figure 2.3: Comparison of DMD with centering and DMD without centering in the presence of measurement noise. a) For fixed $\boldsymbol{A}$, we compute the sum of the distances from the computed eigenvalues of $\boldsymbol{A}$ to the nearest true eigenvalue. For both methods, the sum scales linearly with noise level. b) For fixed noise level 0.005, we plot the eigenvalues of $\boldsymbol{A}$ computed using both method for 100 instantiations of noise. The fluctuations of the eigenvalues from the true values (black crosses) are roughly the same for both methods.

It is important to note that this argument is based on (1) the companion matrix approach and (2) the fact that the companion matrix is unique and hence the data matrix $\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^\intercal$ has full rank. We note that if the data is not full rank, then the companion matrix approach yields different modes than SVD-based DMD [43, 181].

Clearly, if we subtract the mean $\boldsymbol{\mu}_1$ from $\boldsymbol{X}_1$, then $\boldsymbol{X}_1 - \boldsymbol{\mu}_1\mathbf{1}^\intercal$ will be low-rank and therefore have linearly dependent columns. Hence, the argument above does not apply to this case. In particular, we do not expect the DMD with centering method (2.14) to be equivalent to the companion matrix approach $\boldsymbol{X}_1 - \boldsymbol{\mu}_1\mathbf{1}^\intercal$ or equivalent to the DFT.

If we subtract the overall mean $\boldsymbol{\mu}$ from the data as in [43], if $\boldsymbol{X}_1$ is low rank, then the argument above does not apply. Morevover, even if $\boldsymbol{X}_1$ is full rank, if there is a stationary mode, then $\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^\intercal$ will be low rank. This is proven in Proposition 2.6.1 below. In the case of low rank data or a nonzero stationary mode, if we subtract $\boldsymbol{\mu}$, the SVD-based approach will not equal the companion matrix. Furthermore, the assumptions in [43] are not satisfied and neither the SVD-based approach, nor the companion matrix approach is expected to equal the DFT. We have found it not to be equal to the DFT in all experiments we have considered for these cases.

Proposition 2.6.1. *Suppose we have sequential time series such that* $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j$ *which are used to define the matrices* $\boldsymbol{X}$ *and* $\boldsymbol{X}_1$ *as in* (2.3) *and* (2.17)*. If*

1. *the DMD problem is well-posed,*
2. $\hat{\boldsymbol{A}} = \boldsymbol{X}_2\boldsymbol{X}_1^\dagger$ *has eigenvalue* 1*, and*
3. $\boldsymbol{X}$ *has nonzero mean* $\boldsymbol{\mu} = \frac{1}{T+1}\boldsymbol{X}\mathbf{1} \neq 0$*,*

then $\operatorname{rank}(\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^\intercal) \leq \operatorname{rank}(\boldsymbol{X}_1) - 1$.

Proof. To prove that $\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^\intercal$ is rank deficient, we will show that there exists a nonzero vector $\boldsymbol{v}$ which is in the nullspace of $\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^\intercal$, but not in the nullspace of $\boldsymbol{X}_1$.

First, we need to show that range($\boldsymbol{X}_2$) $\subseteq$ range($\boldsymbol{X}_1$). To see this, we only need to consider the case where range($\boldsymbol{X}_2$) $\neq$ range($\boldsymbol{X}_1$). Then, since the DMD problem is well-posed, $T \geq r+1$. Thus, $\boldsymbol{X}_1$ contains at least r linearly independent vectors which are in the range of $\hat{\boldsymbol{A}}$. Thus, range($\hat{\boldsymbol{A}}$) $\subseteq$ range($\boldsymbol{X}_1$), and since $\boldsymbol{X}_2 = \hat{\boldsymbol{A}}\boldsymbol{X}_1$, then range($\boldsymbol{X}_2$) $\subseteq$ range($\hat{\boldsymbol{A}}$) $\subseteq$ range($\boldsymbol{X}_1$).

Since, range($\boldsymbol{X}_2$) $\subseteq$ range($\boldsymbol{X}_1$), then there exists $\boldsymbol{c} \in \mathbb{R}^{n-1}$ such that $\boldsymbol{X}_1\boldsymbol{c} = \boldsymbol{x}_{T+1}$. One possible solution to this is $\boldsymbol{c} = \boldsymbol{X}_1^{\dagger}\boldsymbol{x}_{T+1}$. Define $\boldsymbol{\alpha} = \frac{\mathbf{1}+\boldsymbol{X}_1^{\dagger}\boldsymbol{x}_{T+1}}{T+1}$. By definition,

$$\begin{aligned}
\mathbf{0} \neq \boldsymbol{\mu} &= \frac{1}{T+1}\boldsymbol{X}\mathbf{1} \\
&= \frac{1}{T+1}\left(\boldsymbol{X}_1\mathbf{1} + \boldsymbol{x}_{T+1}\right) \\
&= \frac{1}{T+1}\boldsymbol{X}_1\left(\mathbf{1} + \boldsymbol{c}\right) \\
&= \boldsymbol{X}_1\boldsymbol{\alpha}.
\end{aligned}$$

Thus, $\boldsymbol{\alpha}$ is not in the nullspace of $\boldsymbol{X}_1$ and therefore cannot be $\mathbf{0}$. By Lemma 2.5.1, since $\hat{\boldsymbol{A}}$ has eigenvalue 1, then $\mathbf{1}^{\intercal}\boldsymbol{X}_1^{\dagger}\boldsymbol{x}_{T+1} = \mathbf{1}^{\intercal}\boldsymbol{c} = 1$. Thus,

$$\begin{aligned}
\left(\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^{\intercal}\right)\boldsymbol{\alpha} &= \boldsymbol{X}_1\left(\boldsymbol{I} - \frac{1}{T+1}\left(\mathbf{1}+\boldsymbol{c}\right)\mathbf{1}^{\intercal}\right)\frac{\left(\mathbf{1}+\boldsymbol{c}\right)}{T+1} \\
&= \boldsymbol{X}_1\frac{\left(\mathbf{1}+\boldsymbol{c}\right)}{T+1} - \boldsymbol{X}_1\frac{1}{T+1}\left(\mathbf{1}+\boldsymbol{c}\right)\mathbf{1}^{\intercal}\frac{\left(\mathbf{1}+\boldsymbol{c}\right)}{T+1} \\
&= \boldsymbol{X}_1\frac{\left(\mathbf{1}+\boldsymbol{c}\right)}{T+1} - \boldsymbol{X}_1\frac{\left(\mathbf{1}+\boldsymbol{c}\right)}{T+1} \\
&= \mathbf{0}.
\end{aligned}$$

In conclusion, the dimension of the null space of the centered data ($\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^{\intercal}$) must be greater than the dimension of the null space of the uncentered data ($\boldsymbol{X}_1$) and the centered data must have a lower rank than the uncentered data. □

Remark 2.6.2. *Even in the case where the system has measurement noise, $\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^{\intercal}$ is effectively rank deficient. Thus, the companion matrix modes are not the DMD modes even if the data is mean subtracted using $\boldsymbol{\mu}$.*

To illustrate this point we generate data for an affine system $\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}$ (2.11) with $n = 10$, $T = 7$, and $r = 5$. In Figure 2.4, we plot the true modes (crosses), the spectrum computed with DMD with centering and the spectrum computed using the companion matrix on data with the total mean subtracted. We see that DMD with centering extracts out the correct modes. However, the companion matrix approach does not. Since the data $\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^\intercal$ is low rank, the eigenvalues of the companion matrix are not the roots of unity. In addition, note that the companion matrix has seven nonzero eigenvalues even though $\boldsymbol{A}$ has only 5. Next, we add some Gaussian distributed measurement noise with zero mean and standard deviation 0.001. Like the noiseless case, DMD with centering extracts the correct eigenvalues. Since the data are full rank the companion matrix approach yields a temporal DFT. However, since the data has low effective rank, the eigenvalues of the companion matrix do not equal the eigenvalues of $\boldsymbol{A}$.

2.7 Extracting Arbitrary Frequencies

By subtracting the means of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ individually, we have shown by Theorem 2.5.2 that we can successfully extract the dynamics about a background mode (corresponding to a DMD mode with eigenvalue equal to 1). We generalize this result to modes with fixed frequencies that correspond to known eigenvalues other than 1. As a concrete example, electrical recordings taken in the presence of an alternating current power source are often corrupted with a “background” signal at a fixed frequency (60 Hz in most countries). This line noise corresponds to a mode with a precisely known eigenvalue that we want to subtract from the measurements.

To subtract a mode of known frequency, note that in (2.11) the eigenvalue of 1 comes in through the decision to use $\mathbf{1}^\intercal$. By adding this term we enforce that

$$\mathbf{1}^\intercal = \begin{bmatrix} 1 & 1 \cdots 1 \end{bmatrix}, \tag{2.29}$$

appears in the rowspace of the data. We remove this mode by subtracting the mean from

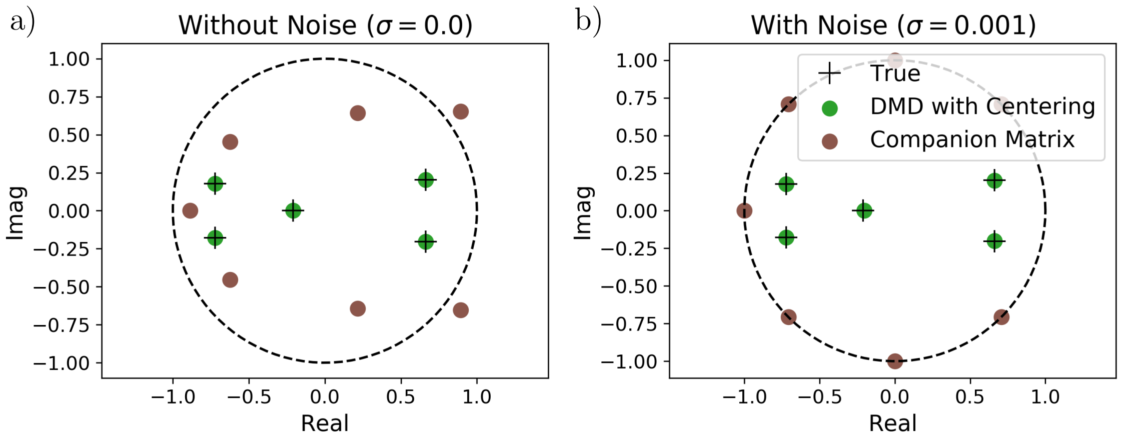


Figure 2.4: Comparison of performance of DMD with centering and the companion matrix approach on mean subtracted data. a) DMD modes (green) match with the true modes (black crosses). Since total mean subtracted data $\boldsymbol{X}_1 - \boldsymbol{\mu}\mathbf{1}^\intercal$ is low rank, the companion matrix eigenvalues do not equal the true eigenvalues. b) Same system as a) but with added measurement noise. DMD with centering yields the correct eigenvalues. Since the data is full-rank, the companion matrix eigenvalues equal the roots of unity. However, since the data has low effective rank, these modes do not equal the true modes of the system.

the data or equivalently applying the orthogonal projection,

$$\boldsymbol{I} - \frac{\mathbf{1}\mathbf{1}^{\mathsf{T}}}{\mathbf{1}^{\mathsf{T}}\mathbf{1}}, \tag{2.30}$$

to $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.

If we know that another eigenvalue λ exists in the data, then we simply replace (2.29) with

$$\boldsymbol{\lambda}^{\mathsf{T}} = \begin{bmatrix} 1 & \lambda & \lambda^2 & \lambda^3 \cdots \lambda^{T-1} \end{bmatrix}.$$

Thus, (2.11) becomes

$$\boldsymbol{X}_2 = \boldsymbol{A}\boldsymbol{X}_1 + \boldsymbol{b}\boldsymbol{\lambda}^{\mathsf{T}}. \tag{2.31}$$

Multiplying both sides by $\boldsymbol{\lambda}^{\mathsf{T}^{\dagger}} = \frac{\boldsymbol{\lambda}^*}{\boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{\lambda}^*}$, then

$$\begin{aligned} \boldsymbol{X}_2 \frac{\boldsymbol{\lambda}^*}{\boldsymbol{\lambda}^{T^*}\boldsymbol{\lambda}} &= \boldsymbol{b}\frac{\boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{\lambda}^*}{\boldsymbol{\lambda}^{T^*}\boldsymbol{\lambda}} + \boldsymbol{A}\boldsymbol{X}_1 \frac{\boldsymbol{\lambda}^*}{\boldsymbol{\lambda}^{T^*}\boldsymbol{\lambda}} \\ \boldsymbol{b} &= \boldsymbol{X}_2 \frac{\boldsymbol{\lambda}^*}{\boldsymbol{\lambda}^{T^*}\boldsymbol{\lambda}} - \boldsymbol{A}\boldsymbol{X}_1 \frac{\boldsymbol{\lambda}^*}{\boldsymbol{\lambda}^{T^*}\boldsymbol{\lambda}}. \end{aligned}$$

Plugging this into (2.31)

$$\boldsymbol{A}\boldsymbol{X}_1 \left(\boldsymbol{I} - \frac{\boldsymbol{\lambda}^*\boldsymbol{\lambda}^{\mathsf{T}}}{\boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{\lambda}^*} \right) = \boldsymbol{X}_2 \left(\boldsymbol{I} - \frac{\boldsymbol{\lambda}^*\boldsymbol{\lambda}^{\mathsf{T}}}{\boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{\lambda}^*} \right).$$

Thus, solving (2.31) is equivalent to applying the orthogonal projection, $\boldsymbol{I} - \frac{\boldsymbol{\lambda}^*\boldsymbol{\lambda}^{\mathsf{T}}}{\boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{\lambda}^*}$ to the data.

If there are multiple known distinct eigenvalues $\lambda_1, \ldots, \lambda_k$, then applying the same procedure we construct the matrix

$$\boldsymbol{\Lambda}^{\mathsf{T}} = \begin{bmatrix} \lambda_1 & \lambda_1^2 & \lambda_1^3 & \cdots & \lambda_1^{\mathsf{T}} \\ \lambda_1 & \lambda_1^2 & \lambda_1^3 & \cdots & \lambda_1^{T} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_k & \lambda_k^2 & \lambda_k^3 & \cdots & \lambda_k^{T} \end{bmatrix},$$

and assume that the data satisfies

$$\boldsymbol{X}_2 = \boldsymbol{A}\boldsymbol{X}_1 + \boldsymbol{B}\boldsymbol{\Lambda}^{\mathsf{T}}. \tag{2.32}$$

In the case that $k < T$, since $\boldsymbol{\Lambda}$ is a Vandermonde matrix it has full column rank (Lemma 2.4.2), and thus $\boldsymbol{\Lambda}^\dagger\boldsymbol{\Lambda} = \boldsymbol{I}$. Multiplying (2.32) by $\boldsymbol{\Lambda}^{\intercal^\dagger}$, and rearranging terms we get

$$\boldsymbol{B} = \boldsymbol{X}_2\boldsymbol{\Lambda}^{\intercal^\dagger} - \boldsymbol{A}\boldsymbol{X}_1\boldsymbol{\Lambda}^{\intercal^\dagger}.$$

Plugging this into (2.32) yields

$$\boldsymbol{A}\boldsymbol{X}_1\left(\boldsymbol{I} - \boldsymbol{\Lambda}^{\intercal^\dagger}\boldsymbol{\Lambda}^\intercal\right) = \boldsymbol{X}_2\left(\boldsymbol{I} - \boldsymbol{\Lambda}^{\intercal^\dagger}\boldsymbol{\Lambda}^\intercal\right).$$

So, solving (2.32) is equivalent to DMD after applying the orthogonal projection $\boldsymbol{I} - \boldsymbol{\Lambda}^{\intercal^\dagger}\boldsymbol{\Lambda}^\intercal$ to the data. As an example, we generate data with samples which satisfy,

$$\boldsymbol{x}_{j+1} = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{b}\lambda^{j-1} \text{ for } j = 1, \ldots, T+1. \tag{2.33}$$

We choose $n = 10$, $T = 9$, $r = 5$ and $\lambda = -i$. The eigenvalues of $\boldsymbol{A}$ (black crosses) are shown alongside the eigenvalues computed using DMD without fixing an eigenvalue (orange) and DMD with a fixed eigenvalue (green). As expected, DMD with a fixed eigenvalue extracts out the eigenvalues of $\boldsymbol{A}$ while DMD without a fixed eigenvalue includes an additional eigenvalue with value $-i$.

2.8 Examples

We demonstrate DMD with centering on three nonlinear examples, including one synthetic example and two real-world datasets. For the Lorenz system, Section 2.8.1 shows that DMD with centering improves the model of the dynamics, especially in the presence of measurement noise. Section 2.8.2 describes the surveillance video example, where the data is effectively low rank; DMD with and without centering extract the same foreground and background modes, as detailed in Section 2.5. From these two examples we see that DMD with centering performs at least as well as, if not better than, DMD without centering. Last, Section 2.8.3 shows extraction of modes at arbitrary frequencies (Section 2.7) using an example of brain activity recordings contaminated by 60 Hz line noise.

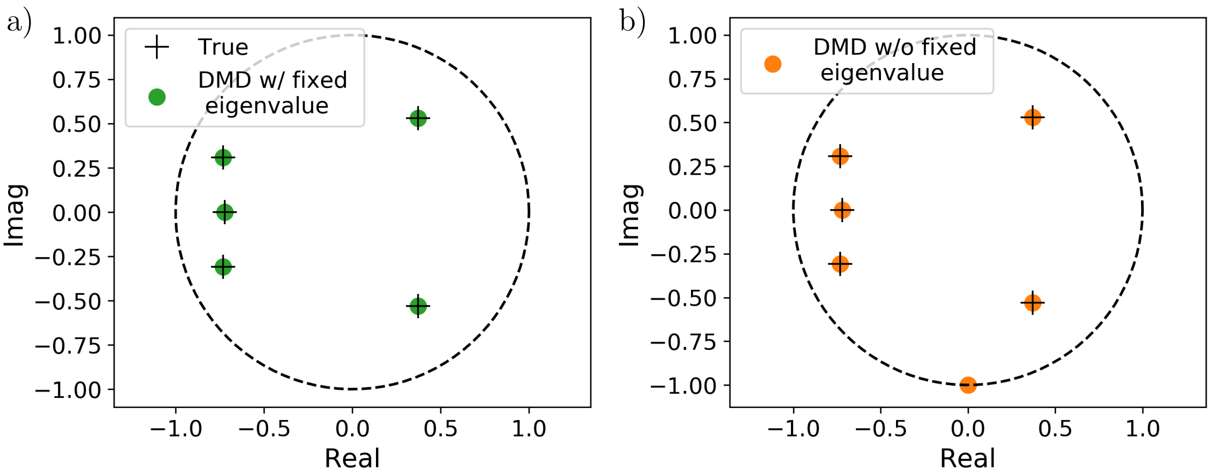


Figure 2.5: Comparison of DMD with fixed frequency subtraction versus ordinary DMD. Data was generated as in (2.33), where $\boldsymbol{A}$ has six nonzero eigenvalues with one fixed to be $-i$. a) Eigenvalues computed using DMD with fixed frequency subtraction (green) compared to true eigenvalues of $\boldsymbol{A}$. b) DMD modes computed without a fixed eigenvalue (orange) compared to true eigenvalues of $\boldsymbol{A}$. DMD without fixed frequency subtraction contains the additional eigenvalue equal to $-i$.

2.8.1 Lorenz System

As an example, we analyze the Lorenz (1963) system [133] which is defined by the set of differential equations

$$\begin{aligned}\dot{x}_1 &= \sigma(x_2 - x_1)\\ \dot{x}_2 &= x_1\left(\rho - x_3\right) - x_2\\ \dot{x}_3 &= x_1 x_2 - \beta x_3.\end{aligned}$$

These equations appear in a variety of systems including, fluid dynamics [130], lasers [224], and chemical reactions [54]. This system is nonlinear and thus the corresponding data matrix $\boldsymbol{X} \in \mathbb{R}^{3\times T+1}$ has linearly independent rows.

For this analysis, we will focus on applying DMD to a short trajectory that spirals outward from the unstable nonzero fixed point $(\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho-1)$. We choose to use the common values $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$. We simulate 4800 timepoints using the standard Runge-Kutta 4th-order method with fixed timestep 0.001 and initial condition $\boldsymbol{x}_1 = [6.7673, 6.1253, 25.8706]$.

In Figure 2.6 we plot the trajectory along with the reconstructed trajectories (forecasts from the initial time) using DMD with centering and DMD without centering. The corresponding eigenvalues from these two methods are shown on the right. DMD with centering and DMD without centering have different eigenvalues in this case. However, both methods give similar reconstructions. Note that DMD has an eigenvalue very close to 1, which indicates that there is a fixed point or nonzero mean in the data.

Next we add Gaussian measurement noise with variance 0.03^2, which is quite small relative to the variable scales. It is well-known that noise shrinks DMD eigenvalues towards the origin [14, 50, 84]. For DMD with centering, even though the eigenvalues shrink towards the origin, the reconstruction is still centered about the fixed point. In addition, since two of the DMD with centering eigenvalues remain outside of the unit circle, the reconstructions have the same growing trend as the simulation. However, for DMD without centering all of the eigenvalues fall within the unit circle, which causes the reconstruction to decay to the origin. So, we conclude that not centering the data can result in drastically different forecasts and estimates of stability.

2.8.2 Background Subtraction for Video Surveillance

Next we analyze an application to video surveillance. Here, we focus on one objective, namely foreground/background separation in video. In particular, we would like to split the data into two pieces: a slowly varying, highly correlated background and a foreground containing moving objects of interest. Several techniques have been developed to address this objective [206, 137, 83, 35]. In [77], Grosek and Kutz show that when DMD is applied to videos with static backgrounds, one of the DMD modes typically has an eigenvalue close to 1. This mode is a good approximation to the background and consequently the difference in the data and this mode is an estimate for the foreground. Here we illustrate the implications of Theorem 2.5.2 and show that DMD with centering yields the same eigenvalues and eigenvectors as DMD without centering, except the eigenvalue of 1 is replaced with an eigenvalue equal to 0. Using DMD with centering we show that the background mode can be computed using the means $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, and $\boldsymbol{A}$.

For this type of data, we expect the video to be approximately low rank. Hence, the foreground from Exact DMD should be approximately equal to the mean subtracted data. In addition, we assume the background corresponds to a nonzero fixed point $\boldsymbol{c}$, where

$$\boldsymbol{X}_2 - \boldsymbol{c}\mathbf{1}^\intercal = \bar{\boldsymbol{A}}(\boldsymbol{X}_1 - \boldsymbol{c}\mathbf{1}^\intercal).$$

Taking the difference between this equation and the equation for DMD with centering

$$\boldsymbol{X}_2 - \boldsymbol{\mu}_2\mathbf{1}^\intercal = \bar{\boldsymbol{A}}\left(\boldsymbol{X}_1 - \boldsymbol{\mu}_1\mathbf{1}^\intercal\right),$$

then

$$\boldsymbol{c} - \bar{\boldsymbol{A}}\boldsymbol{c} = \boldsymbol{\mu}_2 - \bar{\boldsymbol{A}}\boldsymbol{\mu}_1,$$

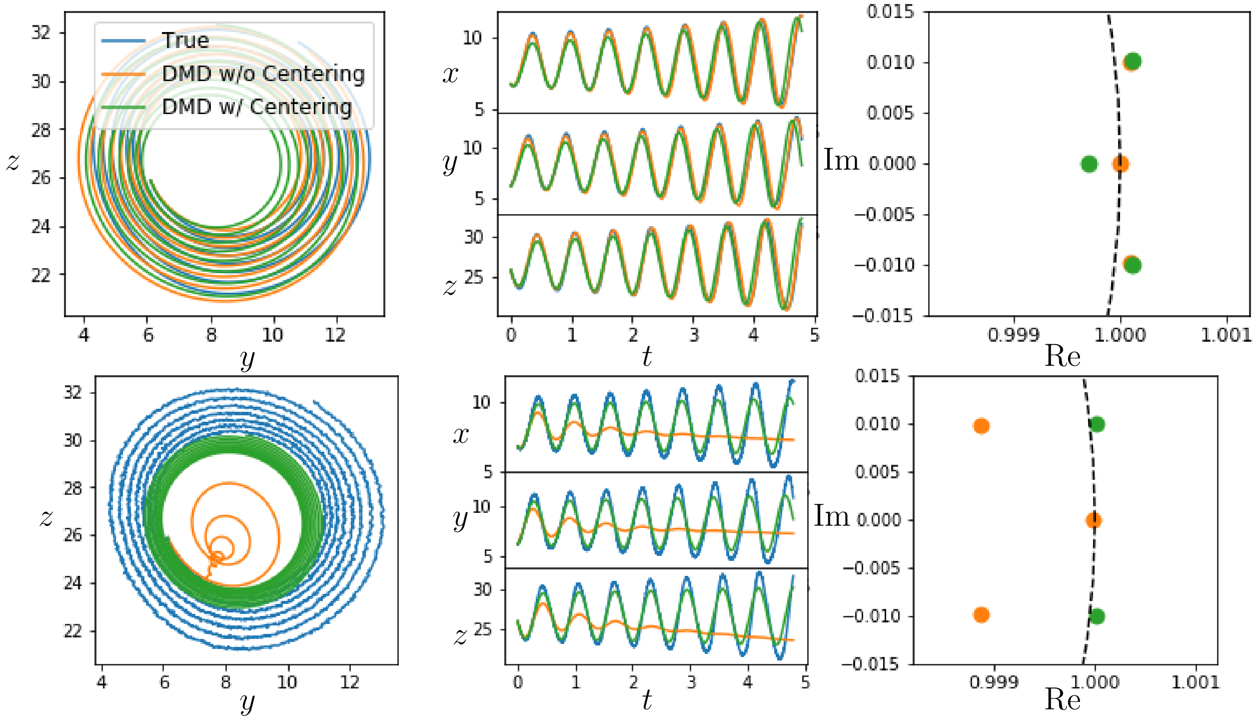


Figure 2.6: Comparison of performance of DMD with and without centering using Lorenz (1963) attractor data. Top left: Reconstruction of z plotted against reconstruction of y for the two methods. Top center: Reconstruction of x, y, and z as a function of t individually using different methods. Both methods produce similar reconstructions. Top right: Eigenvalue spectra for DMD with and without centering. Bottom row: Same as top row except simulation has added Gaussian measurement noise. Note that all of the eigenvalues for DMD without centering have magnitude less than one and decay to zero, causing the reconstructed trajectory to decay to zero. However, some of the DMD with centering modes have magnitude greater than one, yielding a better reconstruction. One eigenvalue equal to 0.8866 is not shown for DMD with centering.

and hence

$$\boldsymbol{c} = (\boldsymbol{I} - \bar{\boldsymbol{A}})^{-1}(\boldsymbol{\mu}_2 - \bar{\boldsymbol{A}}\boldsymbol{\mu}_1).$$

Note that $\boldsymbol{I} - \bar{\boldsymbol{A}}$ is invertible since $\bar{\boldsymbol{A}}$ does not have an eigenvalue equal to 1 according to Theorem 2.5.2. In general, $\boldsymbol{A}$ may be prohibitively large to compute $\boldsymbol{c}$ in (2.8.2). We note that this computation may be performed in the smaller r-dimensional space (see Appendix A.2).

We apply these methods to surveillance video of highway traffic from the CDNET dataset [222]. The video has a height and width of 120×160 pixels and consists of 41 frames. In this case, the foreground is the cars and the background is the grass, road, trees, etc. In Figure 2.7 we show a sample frame, the stationary mode from DMD without centering, the fixed point $\boldsymbol{c}$ from DMD with centering, and the overall mean of the data. The stationary mode and fixed point are visually identical but not equal to the overall mean of the data. As we increase the number of frames in this analysis, the stationary mode and fixed point converge to the overall mean. We choose a short number of frames for this study to illustrate the difference between these quantities. In practice, a larger number of frames will yield better foreground/background separation. Additionally, as predicted by Theorem 2.5.2, the spectra for DMD with and without centering are nearly identical except for the presence of the additional eigenvalue equal to 1 for DMD without centering.

2.8.3 Fixed Frequency Subtraction for Brain Activity Recordings

As a final example, we study an application of Section 2.7 to brain activity recordings and illustrate how fixed frequency subtraction naturally fits within the DMD framework. In particular, we study intracranial electrocorticography (ECoG) measurements from electrodes placed on a human brain surface [219]. The data we use contain 64 channels of measurements and are sampled for a duration of 5 seconds with a frequency of 1000 Hz.

One common source of signal pollution is 60 Hz power line hum, which results from the AC current in power lines [15, 128]. To illustrate the results of Section 2.7, we may apply our

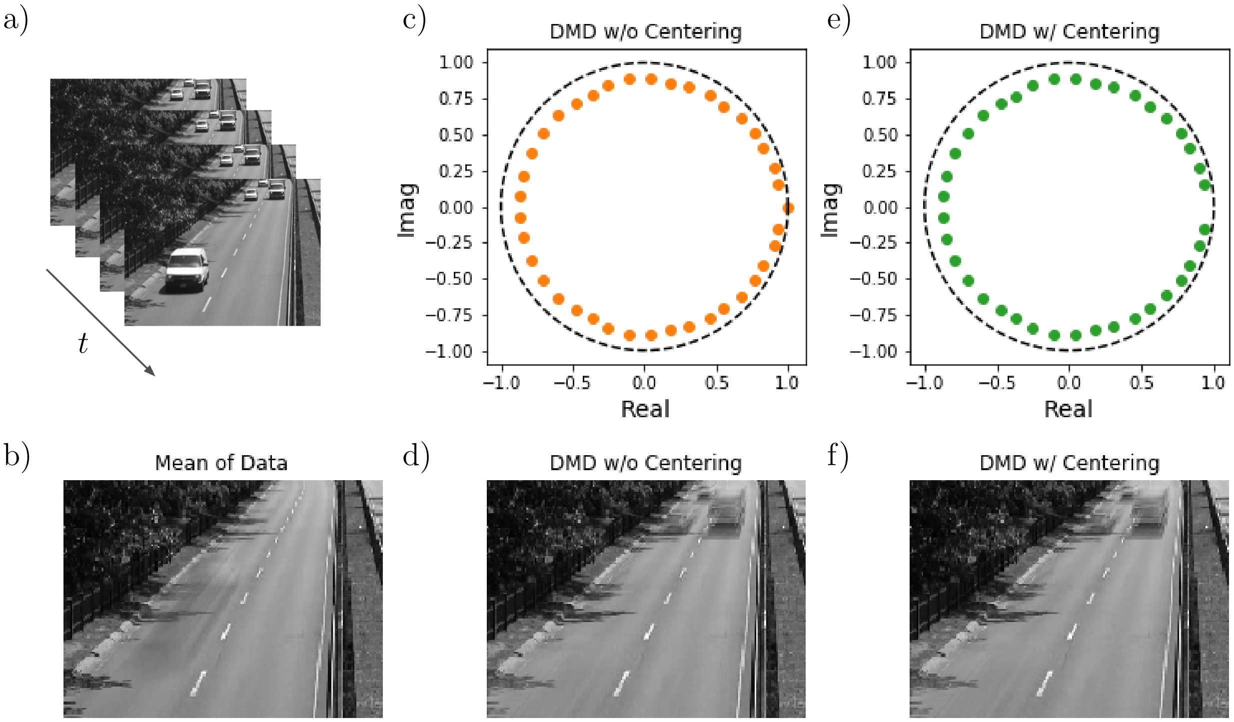


Figure 2.7: a) Sample frames from video of traffic. b) Overall mean μ of video. c) Eigenvalues of modes computed using DMD and d) static mode corresponding to eigenvalue closest to one. e) Eigenvalues of modes computed using DMD with centering and f) static mode corresponding to fixed point. Note that the spectra for these two methods is nearly identical with the exception of the eigenvalue at 1 corresponding to the static mode/fixed point.

method of fixed frequency subtraction to denoise the signal. In particular, we multiply $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ by the orthogonal projection $\boldsymbol{I} - \frac{\boldsymbol{\lambda}^*\boldsymbol{\lambda}^\intercal}{\boldsymbol{\lambda}^\intercal\boldsymbol{\lambda}^*}$ and then apply DMD to these data matrices. $\boldsymbol{\lambda}$ in this case is generated by an eigenvalue which corresponds to a frequency of 60 Hz. In the top left of Figure 2.8, we plot the subset of these channels. The corresponding power spectra computed using the temporal DFT and DMD are shown in the middle left and bottom left plots, respectively. As expected, there is a distinct peak near 60 Hz in both of these plots. On the right we show the corresponding plots after applying the fixed frequency subraction at 60 Hz. In the power spectrum we see that the peak near 60 Hz is suppressed by an order of magnitude. In addition, the mode near 60 Hz in the DMD spectrum is completely removed. Surprisingly, even though the original mode is not at exactly 60 Hz, frequency subtraction is able to remove it. As an alternative method, a notch filter can also be used to remove known frequencies and is the recommended method in practice, since a well-designed filter is less likely to introduce artifacts. In general, there may be additional frequencies corresponding to higher harmonics which we may choose to remove. The results of Section 2.7 may also be used in this case.

2.9 Discussions

In this chapter, we have proposed mean subtraction as a natural and computionally efficient preprocessing step when performing DMD. We have shown that DMD on mean subtracted data is equivalent to an additional affine term in the DMD framework, but is not equivalent to a temporal discrete Fourier transform (temporal DFT). In addition, we showed that, in a special subset of cases, DMD without centering extracts the same spectra as DMD with centering. However, in the case where the data are full rank, DMD with centering can extract the underlying dynamics even when DMD without centering cannot. By thinking of centering the data as subtracting a zero-frequency mode, we generalized this result to extracting non-zero, known frequencies in the data. Finally, we illustrated DMD with centering on three real examples with nonlinear dynamics, namely a trajectory of the Lorenz system, a surveillance video, and brain recordings.

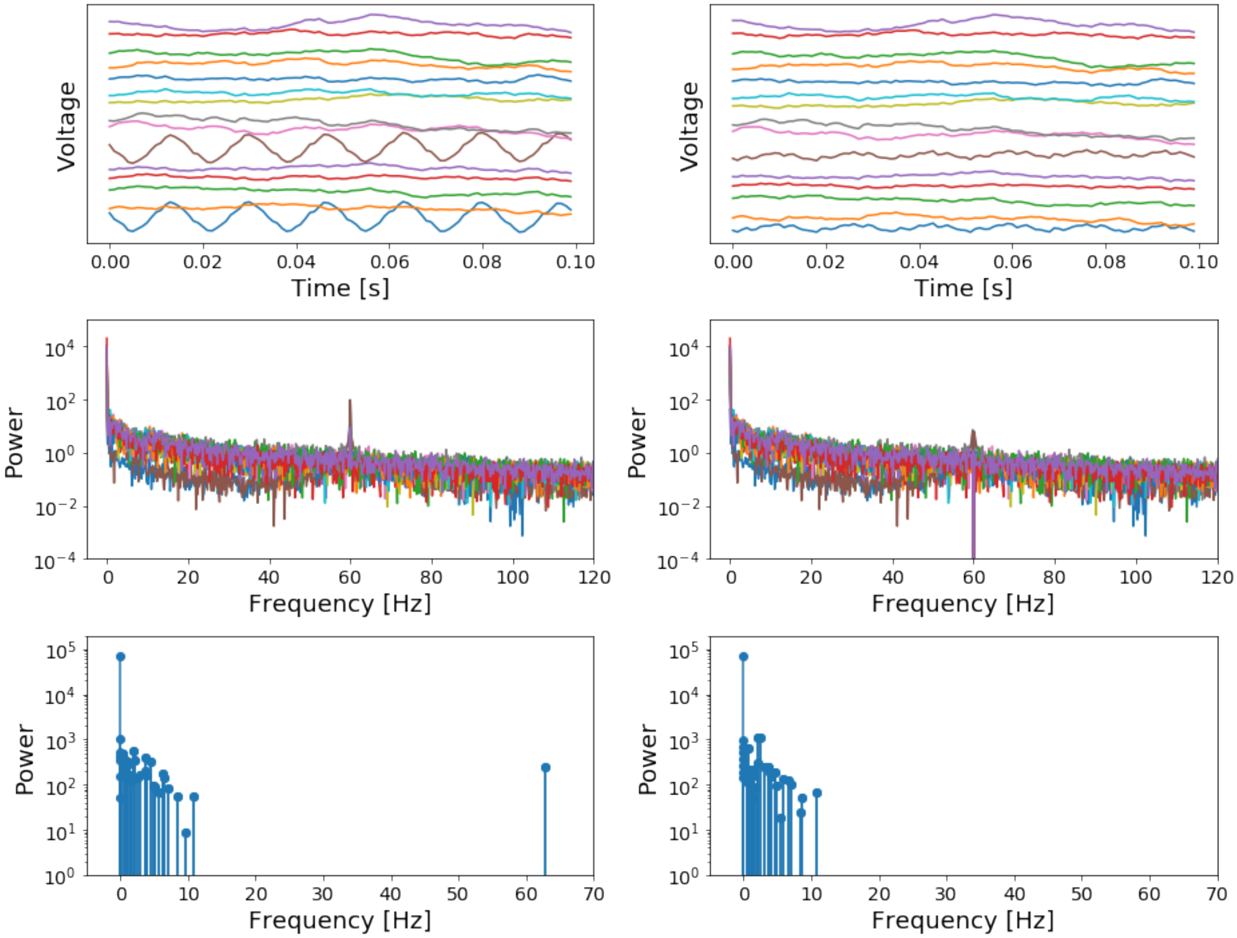


Figure 2.8: Application of fixed frequency subtraction to brain activity recordings . Top left: Raw voltage signals from subset of channels. Center left: Corresponding discrete Fourier transform power spectrum. Bottom left: Power spectrum computed using DMD. Right: Same as left column after fixed frequency of 60 Hz has been subtracted.

Many of the theorems in this work have depended on the assumption of sequential time series sampled at a fixed frequency. In particular, the uniqueness of the DMD modes (Theorem 2.4.6) is based on this assumption. However, exact DMD has been shown to successfully extract modes from data that is not sequential. One potential starting point is the theory of exponential Vandermonde matrices [169, 231]. It remains to be demonstrated that the modes extracted for non-sequential times data by exact DMD, or similar methods such as optimized DMD, are well-posed and unique. Furthermore, future work remains to more thoroughly explore the effects of noise on the DMD estimator and obtain a fully statistical theory.

Our analysis In this chapter has focused on computing DMD by what is known as the (SVD-based) exact DMD algorithm [213]. There exist many other algorithms for computing the DMD, including forward/backward DMD [50], total least squares DMD [84], and optimized DMD [12]. Although we suggest that data centering is generally advantageous, the consequences of centering remains to be explicitly characterized when using these other algorithms.

Chapter 3

SPATIOTEMPORAL MODAL DECOMPOSITION FOR NONSTATIONARY SIGNALS

3.1 Introduction

Here, we introduce the *spatiotemporal intrinsic mode decomposition* (STIMD) method, which factors spatiotemporal data into a product of spatial modes and temporal modes, with the constraint that the temporal modes are *intrinsic mode functions* (IMFs [98, 100]). Our method allows us to perform instantaneous time frequency analysis by computing a Hilbert transform of the data; in addition, it is possible to make future-state predictions of the spatiotemporal system.

The Fourier transform is a very widely used technique for analyzing the power spectral features in time-series signals. However, this technique assumes periodic systems and performs poorly when the signal is nonlinear and non-stationary. A large variety of very successful methods have been developed to work with such signals, including windowed versions of Fourier analysis and wavelet analysis [48]. One related method is the empirical mode decomposition (EMD [98, 167, 230, 101]), which had been developed with the motivation to compute instantaneous time frequency analysis of non-stationary signals. EMD decomposes real-valued signals into a set of intrinsic mode functions (IMFs), which have the feature that they are suitable for computing meaningful Hilbert transforms. EMD has been widely applied in a variety of application domains [99, 158, 220]. Thus, EMD is able to analyze non-stationary time-series data, where frequencies vary in time. However, EMD is an empirical algorithm that was developed without a rigorous mathematical foundation [167]. As alternatives to EMD, the synchrosqueezed wavelet transform [49] and the nonlinear matching pursuit method (NMP) [93, 96] have been developed. Like EMD these methods can be used to extract IMFs. However, these methods have greater mathematical justification in addition to greater statistical robustness.

For systems with spatiotemporal dynamics, it is often possible and desirable to use several sensors placed at different locations to simultaneously gather data about the system. These additional measurements and their correlations in space may be leveraged to produce more accurate models. A large family of methods have been developed for factoring spatiotemporal data into products of two sets of modes—one spatial and one temporal. This factorization problem is also known as the blind source separation problem; in other words, the goal is to extract and disambiguate the underlying signals that comprise the measurement data. The solution to the decomposition is generally underdetermined, and various results can be obtained by making different assumptions [66]. We give a brief overview of some of the most widely used decomposition methods in Section 3.2.

In this chapter, we propose a new solution to the blind source separation problem for spatiotemporal non-stationary signals. This *spatiotemporal intrinsic mode decomposition* (STIMD) is motivated by EMD and builds on the NMP method to factor spatiotemporal data into a set of spatial modes and IMF temporal modes. To our knowledge, no other decomposition has been described to satisfy these assumptions. With STIMD, we can compute an instantaneous time-frequency representation with a Hilbert transform and also perform future state prediction. In Section 3.3, we describe the STIMD method and characterize its behavior on several synthetic non-stationary time-series data examples; in particular, we focus on signals containing frequency modulation. We show that STIMD extracts the underlying source signals more accurately and reliably than several other commonly used factor analysis techniques. Further, we illustrate its dependence on noise magnitude and initial conditions. Next, in Section 3.4, we apply our method to two real-life datasets, namely measurements of gravitational waves from the laser interferometer gravitational observatory (LIGO) experiment and recordings from neural activity from the rodent hippocampus. Our results show that leveraging the architectures jointly greatly improves the performance of spatiotemporal decompositions. A summary and future improvements are found in Section 3.5.

3.2 Related Work

Our STIMD algorithm is based on a number of recent innovations in time frequency analysis of single signals, specifically the empirical mode decomposition (EMD) and nonlinear matching pursuit (NMP). We first describe these techniques and summarize their approach as algorithms in Sections 3.2.1, 3.2.2, 3.2.3.

Next, we give an overview of blind source separation and factor analysis methods to decompose spatiotemporal signals in Section 3.2.4. We highlight the common structure and differing assumptions of three widely used techniques: singular value decomposition (SVD, Section 3.2.5), independent component analysis (ICA, Section 3.2.6), and dynamic mode decomposition (DMD, Section 3.2.7).

3.2.1 EMD and the Hilbert Transform

Consider a signal $f(t) : [t_0, t_1] \to \mathbb{R}$ on which we would like to perform time-frequency analysis, extracting both the temporal and frequency features of the signal simultaneously. If the signal is stationary, we may choose to perform the Fourier transform, which decomposes the signal into a basis of sines and cosines; this basis may also be called a dictionary [92].

For non-stationary signals, a dictionary of sines and cosines do not well represent the signal, so we must choose to decompose our signals using a different basis set. One possible dictionary is the set of all intrinsic mode functions (IMFs); an IMF has the important property that it has a well-defined Hilbert spectrum.

IMFs are defined by the following criteria:

1. The number of extrema and the number of zero crossings of the function must be equal (or differ by at most one).

2. At any point of the function, the average of the upper envelope and the lower envelope

defined by the local extrema must be zero; in other words, the function is symmetric with respect to zero).

Mathematically, all real-valued time-series $s(t)$ obeying these criteria may be expressed in the form

$$s(t) = a(t)\cos(\theta(t)) \tag{3.1}$$

for some $a, \theta : [t_0, t_1] \to \mathbb{R}$ such that $\theta'(t) > 0$ [98].

One notable property of an IMF is that it has a well-defined Hilbert spectrum. Specifically, the function $s(t)$ has the analytic continuation $\tilde{s}(t) : \mathbb{R} \to \mathbb{C}$:

$$\tilde{s}(t) = a(t)e^{i\theta(t)}, \tag{3.2}$$

which has a well-defined instantaneous frequency

$$\omega(t) = \frac{d\theta}{dt}. \tag{3.3}$$

In 1998, Huang introduced a method for decomposing a signal $x(t)$ into a sum of IMFs $s_j(t)$ and a residual $\rho(t)$ through a recursive sifting process known as EMD [98]:

$$x(t) = \sum_j s_j(t) + \rho(t). \tag{3.4}$$

Briefly, at each recursive step of EMD, a cubic spline is fit to the local minima and maxima of the data, forming two envelopes. The mean of these envelopes $m(t)$ is then subtracted from $x(t)$ to form a residual. If the residual $\rho(t)$ is an IMF, it is extracted and the process is applied to the remainder of the data. Otherwise, this process is applied recursively to the residual until an IMF is obtained. Specifically, the jth IMF computed after k iterations (assuming that it satisfies the definition of an IMF) is

$$s_j(t) = x_j(t) - \sum_{i=1}^{k} m_{j,i}(t), \tag{3.5}$$

where $x_j(t) = \sum_{l=1}^{j-1} s_l(t)$ and $m_{j,i}(t)$ is the mean of the envelopes computed after j iterations [98, 19].

This method has been demonstrated to be successful in practice on a wide number of applications [220, 158, 99]. In addition, several multivariate and multidimensional extensions have been developed [168, 139, 59][1]. However, EMD is empirical in nature and its mathematical foundation is still poorly understood; therefore, the next section describes an alternative mathematical architecture with a more rigorous foundation.

3.2.2 Nonlinear Matching Pursuit Method (NMP)

One recent alternative to EMD is the nonlinear matching pursuit (NMP) method developed by Hou and Shi [93]. In particular, for NMP we assume that $x(t)$ can be represented by only a few IMFs. Thus, the goal of NMP is to solve the optimization problem

$$\min M \text{ such that } x(t) = \sum_{j=1}^{M} s_j(t). \tag{3.6}$$

We further assume in this algorithm that the IMFs s_j contain only interwave frequency modulation, as defined in the following definition. Since $\theta(t)$ is monotonic by the Invertible Function theorem, we can express an IMF $s(\cdot)$ as a function of θ

$$s(\theta) = a(\theta)\cos(\theta). \tag{3.7}$$

$s(\theta)$ is defined to have interwave frequency modulation if $a(\theta)$ and $\theta'(\theta) := \frac{d\theta}{dt}|_\theta$ are smoother than $\cos(\theta)$[2]. Saying that $a(\theta)$ and θ' are smoother than $\cos(\theta)$ means that $a(\theta)$ and θ' are in the set

$$V(\theta,\lambda) = \text{span}\left\{1, \cos\left(\frac{k\theta}{2L_\theta}\right), \sin\left(\frac{k\theta}{2L_\theta}\right) : 1 \le k \le 2\lambda L_\theta\right\}, \tag{3.8}$$

where $L_\theta = \left\lfloor \frac{\theta(t_1)-\theta(t_0)}{2\pi} \right\rfloor$ and $\lambda = 1/2$. Note that the parameter λ is important in the implementation of NMP, as described in Section 3.2.3.

---

[1] To the best of our knowledge none of these previous methods perform a matrix factorization comparable to the STIMD method described in Section 3.3.

[2] Since $\theta' > 0$, by the Inverse Function Theorem, there is a one-to-one mapping between t and θ, so that we can thus express the IMF in θ space without losing information.

Physically, the signals with interwave frequency modulation roughly correspond to solutions of second order differential equations of the form

$$\ddot{x} + b(t)\dot{x} + c(t)x = 0, \tag{3.9}$$

where $b(t)$ and $c(t)$ are sufficiently smooth. More details about this solution can be found in [97].

To summarize, the dictionary of interwave frequency modulated IMFs is

$$\mathcal{D} = \{a(\theta)\cos(\theta) : a \in V(\theta), \theta' \in V(\theta) \text{ and } \theta'(t) \geq 0\}, \tag{3.10}$$

and the minimization problem for the NMP method becomes

$$\min_{a_j, \theta_j} M \text{ such that } x = \sum_{j=1}^{M} a_j \cos\theta_j \text{ and } a_j \cos\theta_j \in \mathcal{D}\ \forall j \in 1, \ldots, M. \tag{3.11}$$

In other words, we would like to find the minimum number M of IMFs $a_j \cos\theta_j$ (for $j = 1, \ldots, M$) which sum up to the original signal x [93]. In the case of signals with noise, the equality in (3.11) is replaced with the inequality $\left\| x - \sum_{j=1}^{M} a_j \cos\theta_j \right\|_2 \leq \delta$.

3.2.3 NMP Implementation for Periodic Signals

The NMP minimization problem is solved using matching pursuit. As with the EMD algorithm, each IMF is discovered by a greedy optimization and subtracted from the residual $r_j(t)$ at each step. Specifically, given the signal $x(t)$ we extract the first IMF, by solving the minimization problem

$$\underset{a,\theta}{\operatorname{argmin}} \left\| x(t) - a(\theta(t)) \cos\left(\theta(t)\right) \right\|_2^2, \text{ where } a(\theta), \theta'(\theta) \in V(\theta). \tag{3.12}$$

The solution to this problem leads to the corresponding IMF, $s_1(t) = a(\theta(t))\cos(\theta(t))$. To find the second and subsequent IMFs, we replace the signal $x(t)$ with the residual $r_j(t) = x(t) - \sum_{l=1}^{j-1} s_l(t)$.

To solve the NMP minimization problem an alternating scheme is used by fixing θ and minimizing over a, then fixing a and updating θ. It is also important to note that we first minimize

$$\operatorname*{argmin}_{a,\theta} \|x(t) - a(\theta(t)) \cos(\theta(t))\|_2^2 \, , \text{ where } a(\theta), \theta'(\theta) \in V(\theta, \lambda), \tag{3.13}$$

with $\lambda = 0$ and slowly increase the value of λ up to $1/2$. The corresponding pseudocode is in Algorithm 1.

The minimization in (3.13) is nontrivial to compute. To solve it, Hou and Shi use the fact that projecting $a(\theta)$ into $V(\theta)$ is equivalent to applying a low pass filter in the θ-coordinate [93]. Pseudocode for this algorithm is shown in Algorithm 3. It is important to note that, in addition to taking in the measured signal $x(t)$, the minimization requires an initial guess for $\theta(t)$. Thus in the following sections, we will denote the NMP method as NMP(x, θ).

3.2.4 The Blind Source Separation Problem

For systems with spatiotemporal dynamics, mixed time-series data from multiple sensors may be factored into a sum of spatiotemporal modes. This problem is commonly known as blind source separation or factor analysis.

Generally, suppose we have spatiotemporal data $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, which contains n snapshots and m measurement features at each snapshot. The goal is to decompose $\boldsymbol{X}$ into the product of two matrices $\boldsymbol{B} \in \mathbb{R}^{m \times r}$ and $\boldsymbol{S} \in \mathbb{R}^{r \times n}$ such that

$$\boldsymbol{X} = \boldsymbol{B}\boldsymbol{S}. \tag{3.14}$$

Equivalently, $\boldsymbol{X}$ may be expressed as

$$\boldsymbol{X} = \sum_{j=1}^{r} b_j s_j, \tag{3.15}$$

where $b_j \in \mathbb{R}^m$ is the j^{th} column of $\boldsymbol{B}$ and $s_j \in \mathbb{R}^n$ is the jth row of $\boldsymbol{S}$. The column vectors b_j contain the spatial structure of the data, while the row vectors s_j contain the temporal structure. In other words, the b_j's are the spatial modes and the s_j's are the temporal modes of the data. m is typically referred to as the dimension of the system. For many systems, this quantity corresponds to the number of channels of measurements. The rank r is typically chosen to optimize some objective; if $r < m$, then the decomposition can be used to reduce the dimensionality of $\boldsymbol{X}$ by representing the data in the basis of b_j's.

This decomposition is highly under-determined; given different assumptions and constraints, a large variety of different results for $\boldsymbol{B}$ and $\boldsymbol{S}$ may be obtained. For example, methods such as SVD and ICA make assumptions about the orthogonality and the statistical independence of the data, but they do not assume explicit temporal dynamics. Other methods such as DMD enforce a strict temporal structure on data with a linear dynamic model. Stronger assumptions restrict the types of data that can be modeled and reconstructed accurately; however, these assumptions, if appropriate for the system, may denoise the data and improve the interpretability of the results. Sections 3.2.5, 3.2.6, and 4.2.1 give an overview of these methods, and their properties are summarized in Table 3.1.

3.2.5 Singular Value Decomposition (SVD)

One of the most widely used methods in matrix factorization is the SVD[3]. Given a matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, SVD decomposes $\boldsymbol{X}$ into a product of three matrices

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{3.16}$$

where the left singular vectors $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ and the right singular vectors $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ are unitary matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ is diagonal [72, 111]. It is customary that the diagonal

---

[3] Depending on the domain, this method (with small variations) is also known as Principal Component Analysis (PCA), Proper Mode Decomposition (POD), and the Karhunen-Loẽve Decomposition, among others.

elements of $\boldsymbol{\Sigma}$ be expressed as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_n \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix},$$

where the singular values are in decreasing order, $\sigma_1 \geq \sigma_2 \geq \dots \sigma_n$. The rank of $\boldsymbol{X}$ is R, which corresponds to the number of nonzero σ_j's [122]. Equivalently, we may incorporate the weightings $\boldsymbol{\Sigma}$ into $\boldsymbol{U}$. Letting $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Sigma}$ and $\boldsymbol{S} = \boldsymbol{V}^T$ we recover (3.14).

When $\boldsymbol{X}$ is spatiotemporal data, we may interpret $\boldsymbol{B}$ as the spatial modes of $\boldsymbol{X}$ and $\boldsymbol{S}$ as the temporal modes of $\boldsymbol{X}$. Consider the matrix $\boldsymbol{X}_r$ defined as

$$\boldsymbol{X}_r = \sum_{j=1}^{r} b_j s_j, \tag{3.17}$$

where $0 \leq r \leq R$. This matrix has rank r; importantly, $\boldsymbol{X}_r$ is the best rank r approximation to $\boldsymbol{X}$. More precisely, if $\boldsymbol{Y}$ is a rank $r \leq R$ matrix, then $\|\boldsymbol{X} - \boldsymbol{Y}\|$ is minimized for $\boldsymbol{Y} = \boldsymbol{X}_r$ with respect to both the ℓ_2 and Frobenius norms. The relative error in the rank r approximation with respect to the ℓ_2 norm is

$$\frac{\|\boldsymbol{X} - \boldsymbol{X}_r\|_2}{\|\boldsymbol{X}\|_2} = \frac{\sigma_{r+1}}{\sigma_1}. \tag{3.18}$$

From this, we see that if the singular values σ_i decay sufficiently rapidly such that $\sigma_{r+1} \ll \sigma_1$, then $\boldsymbol{X}_r$ will be a very good approximation to $\boldsymbol{X}$. This property makes SVD a popular tool for performing dimensionality reduction and mode extraction.

3.2.6 Independent Component Analysis (ICA)

Independent component analysis is another commonly used method for performing blind source separation. Common applications of ICA include brain imaging, finance, and image feature extraction [13, 129, 104, 103]. Like other blind source separation problems, we assume that there is a set of r signals $s_1, \ldots, s_r \in \mathbb{R}^n$ and we measure linear combinations $b_{i,j} \in \mathbb{R}$ to form the signals $x_i(t) \in \mathbb{R}^m$, as in (3.15)

$$x_i = \sum_{j=1}^{r} b_{i,j} s_j. \tag{3.19}$$

ICA makes the following assumptions:

1. The source signals s_j are mutually statistically independent.
2. The s_j's follow non-gaussian distributions.
3. The mixing matrix $\boldsymbol{B}$ is orthogonal.

By the central limit theorem, a linear combination of signals tends to be more Gaussian than the distribution of a single signal. Consequently, many ICA algorithms, compute the $\boldsymbol{B}$ and $\boldsymbol{S}$ matrices by maximizing the nongaussianity of the source signals, which can be computed using kurtosis or negentropy. Alternatively, some methods optimize the statistical independence between signals by minimizing their mutual information or by employing joint diagonalization [36, 18, 20].

One of the most popular algorithms for performing ICA is FastICA [105]. To solve this problem, FastICA uses the method of projection pursuit: find the direction w_j such that the projection $w_j^T \boldsymbol{X}$ maximizes the measure of nongaussianity. Constraining the w_j's to be orthogonal yields a solution to $\boldsymbol{W}\boldsymbol{X} = \boldsymbol{S}$ (where the jth row of $\boldsymbol{W}$ is w_j). By construction, $\boldsymbol{W}$ is an orthogonal matrix, letting $\boldsymbol{B}$ be the pseudoinverse $\boldsymbol{W}^\dagger = \boldsymbol{W}^T$ to yield $\boldsymbol{X} = \boldsymbol{B}\boldsymbol{S}$ as desired.

3.2.7 Dynamic Mode Decomposition (DMD)

As with SVD and ICA, we have some data $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, but here we define $x(t_k) \in \mathbb{R}^m$ to be the state of the system at time t_k. We will further assume that the state has been sampled evenly in time at some spacing Δt at a total of n snapshots. DMD has become a popular tool to model dynamical systems in the fields of fluid mechanics, neuroscience, and image analysis [173, 181, 125, 29].

The goal of DMD is to determine the best linear operator to $\boldsymbol{B} : \mathbb{R}^m \to \mathbb{R}^m$ such that

$$x(t_{k+1}) \approx \boldsymbol{B}x(t_k). \tag{3.20}$$

We let

$$\boldsymbol{X}_1^{n-1} = \begin{bmatrix} | & | & \cdots & | \\ x(t_1) & x_2(t_2) & \cdots & x(t_{n-1}) \\ | & | & \cdots & | \end{bmatrix} \text{ and } \boldsymbol{X}_2^n = \begin{bmatrix} | & | & \cdots & | \\ x(t_2) & x(t_3) & \cdots & x(t_n) \\ | & | & \cdots & | \end{bmatrix}.$$

Then, we can equivalently define $\boldsymbol{B} \in \mathbb{R}^{m \times m}$ to be the operator such that

$$\boldsymbol{X}_2^n \approx \boldsymbol{B}\boldsymbol{X}_1^{n-1}. \tag{3.21}$$

To find $\boldsymbol{B}$, we must solve the minimization problem

$$\min_{\boldsymbol{B}} \left\| \boldsymbol{X}_2^n - \boldsymbol{B}\boldsymbol{X}_1^{n-1} \right\|_F, \tag{3.22}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. A unique solution to this problem can be obtained using the exact DMD method [212]. For noisy data, we can use more robust methods such as optimized DMD [12].

One key benefit of DMD is that it builds an explicit temporal model, which allows for future state prediction. Specifically, if we let $\{\lambda_j\}$ and $\{b_j\}$ be the eigenvalues and eigenvectors of $\boldsymbol{B}$, respectively, then

$$x(t) = \sum_{j=1}^{r} b_j e^{\omega_j t} c_j, \tag{3.23}$$

where $\omega_j = \ln(\lambda_j)/\Delta t$ and $c \in \mathbb{R}^m$ corresponds to the initial conditions of the state. Thus, to compute the state at an arbitrary time t, simply evaluate (3.23) at that time.

Defining $s_j(t) = \exp(\omega_j t)c_j$, then we have

$$x(t_k) = \sum_{j=1}^{r} b_j s_j(t_k). \tag{3.24}$$

Thus, we can think of DMD as a matrix factorization as in (3.15), where the spatial modes are the b_j's, and the temporal modes s_j all have complex exponential temporal dependence.

3.3 Spatiotemporal Intrinsic Mode Decomposition (STIMD)

Our STIMD algorithm leverages the mathematical and algorithmic structures of EMD and NMP for improved spatiotemporal decompositions. The mathematical framework and algorithmic implementation are given in the following subsections.

3.3.1 Method Description

As introduced in Section 3.2, we assume we have a set of source signals s_j for $j = 1, \ldots, r$ that are linearly mixed to form the observed signals x_i,

$$x_i(t) = \sum_{j=1}^{r} b_{i,j} s_j(t), \tag{3.25}$$

or equivalently in matrix form

$$\boldsymbol{X} = \boldsymbol{BS}. \tag{3.26}$$

Here we assume that all modes s_j are IMFs with interwave frequency modulation, as defined in Section 3.2.2. Thus, each s_j takes the form

$$s_j(t) = a_j(\theta_j(t))\cos(\theta_j(t)) = a_j(\theta_j)\cos(\theta_j), \tag{3.27}$$

where $a_j(\theta_j), \theta_j' \in V(\theta_j)$ are as defined in (3.8).

As summarized in Table 3.1, this mathematical architecture provides a compromise between ICA, SVD and DMD. We obtain a model of the temporal dynamics that are constrained to obey a set of dynamics commonly found in physical systems, but they are not restricted to be stationary signals as in DMD. In addition, we have the ability to compute a Hilbert spectrum and perform future state prediction of non-stationary signals.

Table 3.1: Comparison of assumptions, features, and limitations of the spatiotemporal decomposition algorithms.

	SVD	ICA	DMD	STIMD
Derives orthogonal spatial modes	✓	✓	×	×
Models temporal dynamics	×	×	✓	✓
Adapts to non-stationary signals	✓	✓	×	✓
Predicts future states	×	×	✓	✓
Suitable for Hilbert spectrum computation	×	×	✓	✓

3.3.2 Method Implementation

Our goal is to find a set of directions b_j and IMFs s_j for $j = 1, \ldots, r$, which minimize the remainder

$$\boldsymbol{R} = \min_{b_j, s_j} \left\| \boldsymbol{X} - \sum_{j=1}^{r} b_j s_j \right\|_2 \text{ such that } s_j \in \mathcal{D}, \tag{3.28}$$

where $\mathcal{D}$ is the dictionary of IMFs with interwave frequency modulation described in (3.10). To solve this minimization problem we employ a variation on multichannel matching pursuit [74]. In particular, we first search for the IMF s_1 and vector b_1 which best satisfies

$$\boldsymbol{R}_1 = \min_{b_1, s_1} \|\boldsymbol{X} - b_1 s_1\|_2 \text{ such that } s_1 \in \mathcal{D} \tag{3.29}$$

and solve

$$\boldsymbol{R}_j = \min_{b_j, s_j} \left\| \boldsymbol{X} - \sum_{k=1}^{j} b_k s_k \right\|_2 \text{ such that } s_j \in \mathcal{D} \tag{3.30}$$

for subsequent IMFs $1 < j \leq r$. To solve (3.29) we first note that for given b_1, the solution for s_1 is approximately

$$s_1 \approx \mathrm{NMP}(b_1^{\dagger} \boldsymbol{X}) = \mathrm{NMP}(b_1^T \boldsymbol{X}). \tag{3.31}$$

Note that we apply NMP to the least squares solution to constrain s_1 to be in $\mathcal{D}$. In addition, without loss of generality we can constrain b_1 to unit norm, incorporating scaling factors into

s_1. Thus, the pseudoinverse of b_1, $b_1^\dagger$, is equal to b_1^T [157].

To find b_1 we substitute (3.31) into (3.29), which yields

$$b_1 = \underset{w}{\operatorname{argmin}} \left\| w\mathrm{NMP}(w^T \boldsymbol{X}, \theta_1) - \boldsymbol{X} \right\|_2^2 \quad \text{subject to } \|w\|_2 = 1. \tag{3.32}$$

To find the next IMF, we apply the same method to the remainder $\boldsymbol{R}_1$,

$$\boldsymbol{R}_1 = \boldsymbol{X} - b_1 s_1. \tag{3.33}$$

Thus, to find the jth IMF s_j we solve the minimization problem

$$b_j = \underset{w}{\operatorname{argmin}} \left\| w\mathrm{NMP}(w^T \boldsymbol{R}_{j-1}, \theta_1) - \boldsymbol{R}_{j-1} \right\|_2^2 \quad \text{subject to } \|w\|_2 = 1, \tag{3.34}$$

where $\boldsymbol{R}_j$ is given by

$$\boldsymbol{R}_j = \boldsymbol{X} - \sum_{k=1}^{j} b_k s_k. \tag{3.35}$$

The jth IMF s_j is then

$$s_j = \mathrm{NMP}(b_j^T \boldsymbol{R}_{j-1}). \tag{3.36}$$

This decomposition in matrix form produces

$$\underbrace{\begin{bmatrix} - & x_1 & - \\ - & x_2 & - \\ \vdots & \vdots & \vdots \\ - & x_m & - \end{bmatrix}}_{\boldsymbol{X}} \approx \underbrace{\begin{bmatrix} | & | & \cdots & | \\ b_1 & b_2 & \cdots & b_r \\ | & | & \cdots & | \end{bmatrix}}_{\boldsymbol{B}} \underbrace{\begin{bmatrix} - & s_1 & - \\ - & s_2 & - \\ \vdots & \vdots & \vdots \\ - & s_r & - \end{bmatrix}}_{\boldsymbol{S}}. \tag{3.37}$$

We note that this method is based on NMP which has dependence on initial guesses for $\theta(t)$. We discuss STIMD's robustness to initial guesses in Section 3.3.5. The corresponding pseudocode for STIMD is shown in Algorithm 2.

Future State Prediction

As a consequence of the connection to NMP, it is easy to extract the phase $\theta(t)$ for each IMF s_j evaluated in time from STIMD. From (3.8), we know that for each IMF $s_j(\theta) = a(\theta)\cos(\theta)$

we have assumed that $\theta'(\theta) \in V(\theta)$. In particular,

$$\theta'(\theta) = \alpha_0 + \sum_k \beta_k \cos\left(\frac{k\theta}{L_\theta}\right) + \gamma_k \sin\left(\frac{k\theta}{L_\theta}\right). \tag{3.38}$$

The coefficients α_0, β_k, and γ_k can all be computed using the Fourier transform in the θ coordinate. With these coefficients, we now have an implicit first order differential equation that can be integrated to compute $\theta(t)$, forming a model of $s_j(\theta(t))$ at any future time.

3.3.3 Experiments on Synthetic Examples

We demonstrate STIMD on a number of synthetic examples. In addition to showing its ability to accurately extract mixed non-stationary source signals, we characterize its sensitivity to noise and initial conditions.

2D Example

First, we consider a simple two-dimensional example. The true signals s_1 and s_2 are oscillatory signals with frequency modulation,

$$\begin{aligned} s_1(t) &= \sin(10\pi t) \\ s_2(t) &= \sin(20\pi(t+0.4)^2), \end{aligned}$$

for $t \in [0, 1]$, and the mixing matrix $\boldsymbol{B}$ is given by

$$\boldsymbol{B} = \begin{bmatrix} \cos(\phi_0) & -\sin(\phi_0) \\ \sin(\phi_0) & \cos(\phi_0) \end{bmatrix},$$

where $\phi_0 = 0.7$. By construction, $\boldsymbol{B}$ is orthogonal, since the matrix used is a standard rotation matrix. Adding a small amount of measurement noise to each of the sensors $N_1(t), N_2(t) \sim \mathcal{N}(0, 0.1)$, the observed (mixed signals) $\boldsymbol{X} = \boldsymbol{BS} + \boldsymbol{N}$ are given by

$$\begin{aligned} x_1(t) &= \cos(\phi)s_1(t) - \sin(\phi)s_2(t) + N_1(t) \\ x_2(t) &= \sin(\phi)s_1(t) + \cos(\phi)s_2(t) + N_2(t). \end{aligned}$$

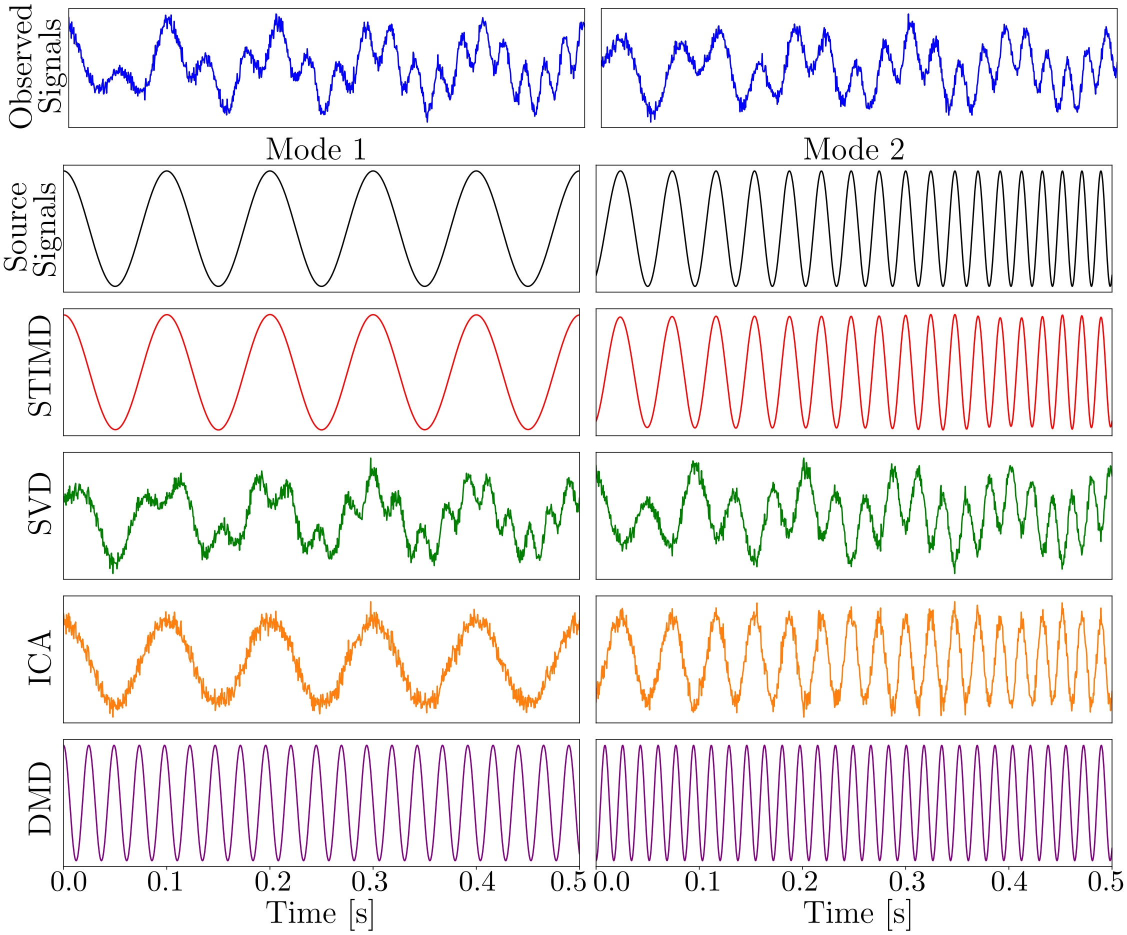


Figure 3.1: Comparison of temporal modes extracted by several blind signal separation algorithms for a two-dimensional system. Top row: The observed mixed measurement signals in blue. These correspond to linear combinations of source signals plus a small amount of Gaussian-distributed noise. Row 2: The two true source signals in black. Row 3: Modes extracted by STIMD in red. Row 4–6: Modes extracted by ICA (green), SVD (orange), and by optimized DMD (purple), respectively.

Figure 3.1 shows the source signals (s_1, s_2) (black), mixed noisy measured signals $x_1(t), x_2(t)$ (blue) and the signals extracted using STIMD (red). Initial guesses of $\theta_1(t) = 10\pi t$ and $\theta_2(t) = 30\pi t$ are used. For comparison, the modes extracted from SVD (green), ICA (orange), and optimized DMD (purple) are also shown. From this, we see that the STIMD modes closely capture the true source signals. The SVD modes are clearly still a mixture of the measured signals. The ICA modes contain some amplitude modulation in time not seen in the true signals. More importantly, the modes extracted by SVD and ICA are not IMFs and consequently are not guaranteed to have meaningful Hilbert spectrums. Lastly, using the optimized DMD algorithm, neither non-stationary mode is extracted correctly.

3D Example

Next, we consider an example containing 3 modes. The source signals are

$$\begin{aligned} s_1(t) &= \cos(20\pi t - 5\sin(\pi t)) \\ s_2(t) &= \cos(60\pi t + 2\sin(4\pi t)) \\ s_3(t) &= \cos(90\pi t + 3\sin(8\pi t)), \end{aligned}$$

and the mixing matrix is

$$\boldsymbol{B} = \begin{bmatrix} \cos(\phi_1)\sin(\phi_2) & -\sin(\phi_1) & \cos(\phi_1)\cos(\phi_2) \\ \sin(\phi_1)\sin(\phi_2) & \cos(\phi_1) & \sin(\phi_1)\cos(\phi_2) \\ \cos(\phi_2) & 0 & -\sin(\phi_2) \end{bmatrix},$$

with $\phi_1 = 0.6$ and $\phi_2 = 0.7$. As before, the observed signals x_i are the linear combinations

$$\underbrace{\begin{bmatrix} - & x_1(t) & - \\ - & x_2(t) & - \\ - & x_3(t) & - \end{bmatrix}}_{\boldsymbol{X}} = \boldsymbol{B} \underbrace{\begin{bmatrix} - & s_1(t) & - \\ - & s_2(t) & - \\ - & s_3(t) & - \end{bmatrix}}_{\boldsymbol{S}},$$

or writing it more succinctly, $\boldsymbol{X} = \boldsymbol{B}\boldsymbol{S}$.

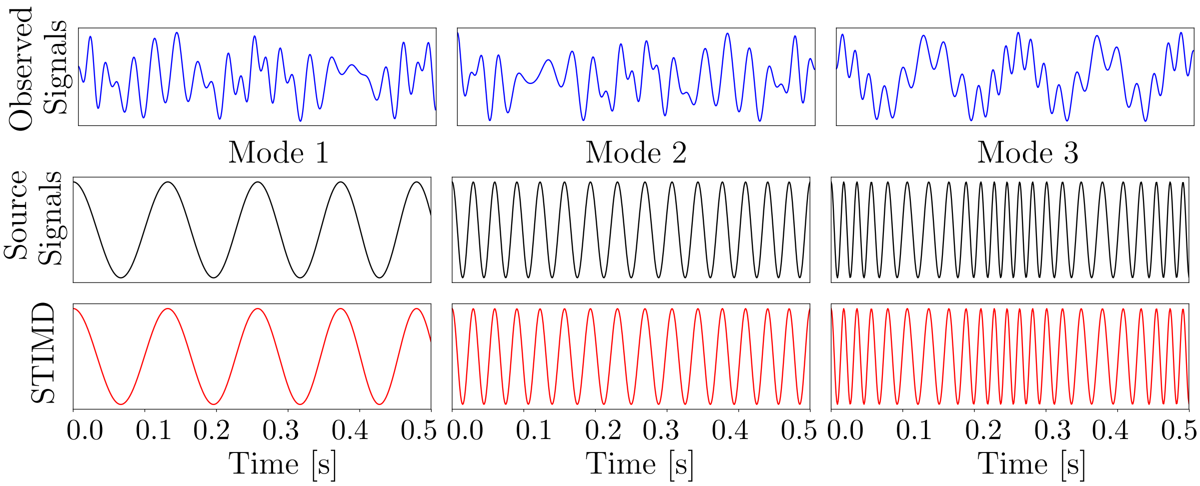


Figure 3.2: Example of STIMD applied to spatiotemporal data with three modes. Top: Observed signals which are linear combinations of source signals. Center: The source signals. Bottom: Signals reconstructed using STIMD.

The original modes (s_1, s_2, s_3) the observed signals (x_1, x_2, x_3) and the modes extracted by STIMD are shown in Figure 3.2. The modes extracted by STIMD are nearly identical to the original signals. Note that the initial guesses $\theta_1(t) = 20\pi t$, $\theta_2(t) = 60\pi t$ and $\theta_3(t) = 90\pi t$ are used.

We next consider how the results are affected by noise. As an example, we add Gaussian noise with standard deviation $\sigma = 0.3$ to the measurements like for the 2D example. As in the noiseless case, we plot the original signals s_j the measured signals x_i and the STIMD modes (Figure 3.3). There is a small amount of amplitude modulation not present in the original signals (which all have amplitude 1). Even so, the frequencies are nearly identical to the true signals.

Figure 3.4 characterizes how the results are affected for various noise levels. Here we add Gaussian noise to the measured signals, sweeping standard deviations σ from 10^{-4} to 1. For each value of σ we perform 100 trials and record the relative error of the extracted modes to the source signals with respect to the 2-norm. As expected, the error increases for increasing σ and only becomes order 1 (about the same size as the signals) when σ is of the same order

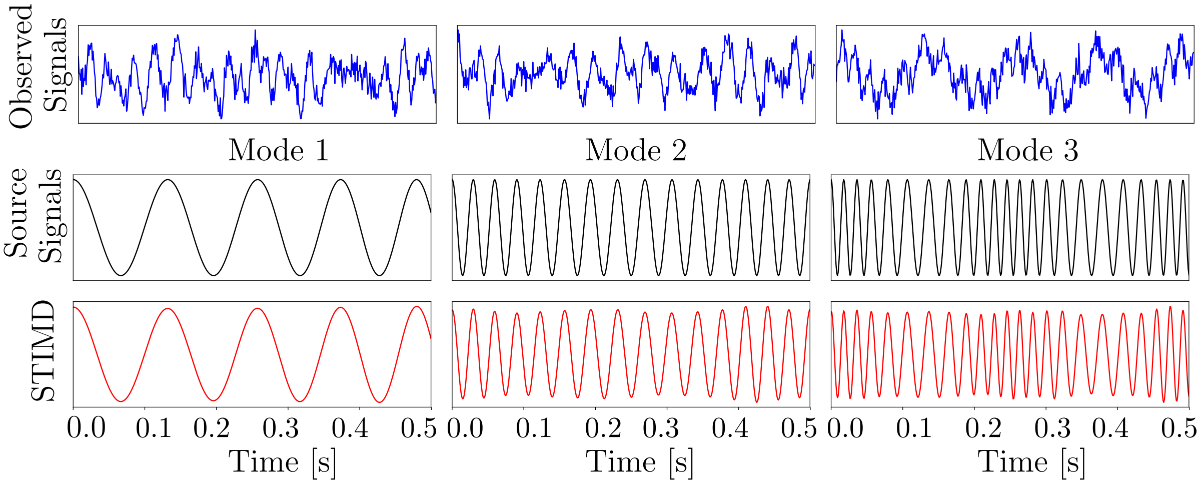


Figure 3.3: Example of STIMD applied to spatiotemporal data with three modes in the case of measurement noise. Top: The source signals. Center: Observed signals which are linear combinations of source signals plus Gaussian distributed noise. Bottom: Signals reconstructed using STIMD.

of magnitude.

Figure 3.5 illustrates how the STIMD results from this three-dimensional system can be used for future state prediction, as described in Section 3.3.2. As an example, consider the STIMD modes (red) over the range $t \in [0, 1]$ for the noiseless system(see Figure 3.2). We compute and predict the modes (black dotted lines) over the greater range of $t \in [0, 2]$ (Figure 3.5). The predicted modes follow the STIMD modes accurately over the interval $[0, 1]$ and accounts accurately for the frequency modulation and amplitude modulation over $[1, 2]$. Note that the state prediction will be only as good as the original reconstruction. In the case of noise the error in the prediction will go as the error in the STIMD modes.

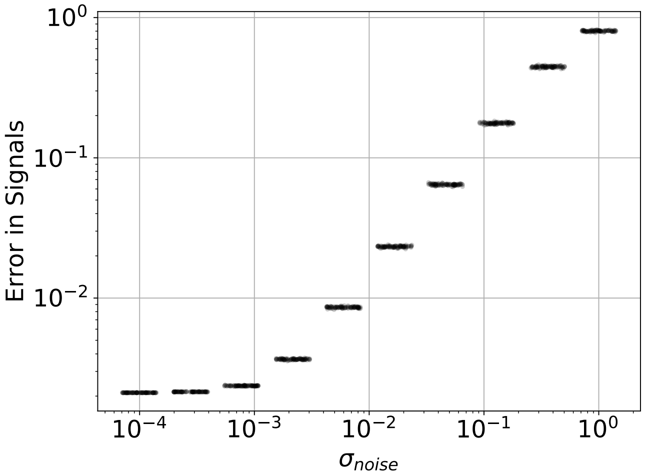


Figure 3.4: Characterization of accuracy of STIMD modes as a function of measurement noise. For each given value of noise 100 realizations of Gaussian distributed noise (with standard deviation σ) were added to the true signal. The relative error between the IMFs extracted by STIMD and the true IMFs are plotted. Random uniform jitter is added on a log-scale to each set of trials for visualization.

3.3.4 4D Example

As a final synthetic example, we show STIMD applied to the case of eight (8) observed signals and four (4) source signals

$$\begin{aligned} s_1(t) &= \cos(20\pi t - 5\sin(\pi t)) \\ s_2(t) &= \cos(30\pi t + \sin(4\pi t)) \\ s_3(t) &= \cos(60\pi t + 3\sin(5\pi t)) \\ s_4(t) &= \cos(80\pi t + 4\sin(5\pi t)). \end{aligned}$$

Thus, the mixing matrix $\boldsymbol{B} \in \mathbb{R}^{8\times 4}$ is a non-square matrix. Noise with standard deviation $\sigma = 0.1$ is added.

The source signals $\boldsymbol{S}$, the observed signals $\boldsymbol{X}$, and the STIMD modes are shown in

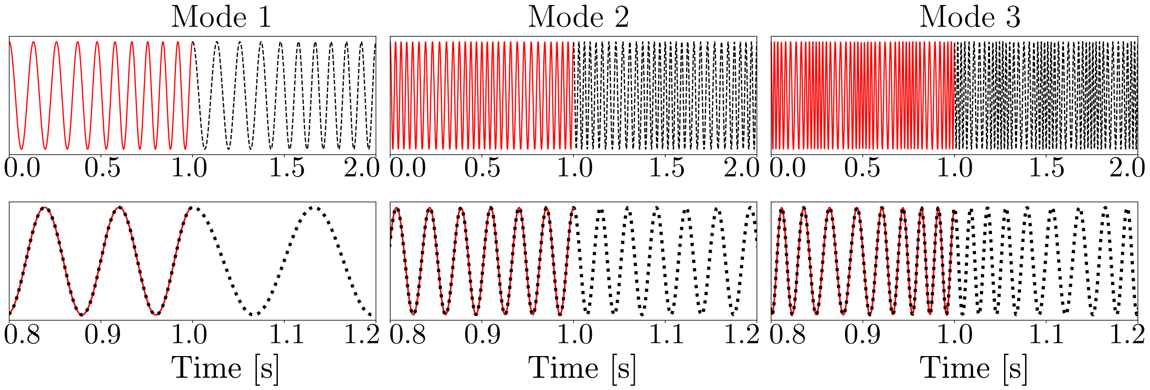


Figure 3.5: Example of future state prediction for spatiotemporal data with three modes. Observed signals, and corresponding STIMD modes are shown in Figure 3.2. Top row: The STIMD modes (red) were computed over the interval $t \in [0, 1]$, while the state was predicted over the window $t \in [0, 2]$ (black dotted). Note that the prediction ranges over the full interval. However, for illustration only the future state is shown. Bottow row: Same system as in the top row, zoomed in over the interval $t \in [0.8, 1.2]$.

Figure 3.6. Initial conditions $\theta_1(t) = 20\pi t$, $\theta_2(t) = 30\pi t$, $\theta_3(t) = 60\pi t$, and $\theta_4(t) = 80\pi t$ are used. Besides a small amount of amplitude modulation, once again the STIMD modes are nearly identical to the original signals.

3.3.5 Initial Conditions

It is important to emphasize that the NMP algorithm, and consequently the STIMD algorithm, takes as input initial guesses for the phases $\theta_j(t)$ of the IMFs. In many cases, only coarse guesses are needed. For example, for the 3D example, guesses corresponding to the central frequencies, $\theta_1(t) = 20\pi t$ and $\theta_2(t) = 60\pi t$ and $\theta_3(t) = 90\pi t$ are needed. For one dimension, Shi recommends taking the Fourier transform and picking the peaks in the spectrum as initial guesses [93]. For the STIMD algorithm, we recommend using the peaks in the Fourier spectra of the temporal modes computed using FastICA.

In addition, choosing the order in which the guesses are applied can affect the IMFs

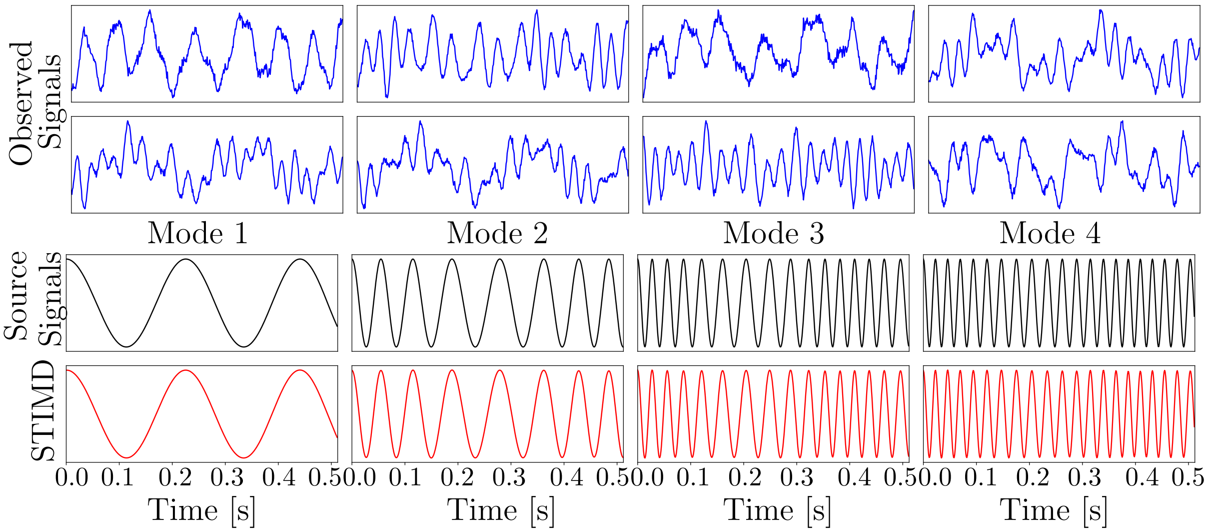


Figure 3.6: Example of STIMD applied to spatiotemporal data with four modes in the case of measurement noise, and eight measured signals. Top row: The source signals. 2nd row and 3rd row: Observed signals which are linear combinations of source signals plus Gaussian distributed noise. Bottom row: Signals reconstructed using algorithm.

extracted from STIMD. Here, two examples are presented in Figure 3.7. For this system, the source signals (shown in black) are

$$\begin{aligned} s_1(t) &= \sin(14\pi t - 5\sin(\pi t)) \\ s_2(t) &= \cos(30\pi t + 4\sin(2\pi t)), \end{aligned}$$

and a 2×2 mixing matrix is used. STIMD is then applied to the mixed signals with initial guesses for frequencies ranging from 1 s^{-1} to 25 s^{-1} (or equivalently θ' ranging from 2π rad/s to 50π rad/s).

The squared error between the STIMD modes and the true source signals is visualized on the right. The purple region corresponds to the region where the modes are visually correct. The blue regions with relative errors near 0.5 typically correspond to when one of the two modes was correct. When the relative error is near 1 (corresponding to the green and yellow regions) the modes extracted are completely incorrect.

For the second example, the source signals (like in Section 3.3.3) are

$$\begin{aligned} s_1(t) &= \sin(10\pi t) \\ s_2(t) &= \sin(20\pi(t + 0.4)^2). \end{aligned}$$

It's important to note that, in both examples, the guesses need to be within a few Hz of the central frequencies of the source signals to obtain accurate results. Also, note that the distributions are not symmetric. In other words, the order in which the guesses are made for each frequency matters. In general, we recommend guessing frequencies in ascending order. This makes sense since the NMP algorithm is based on successive applications of low-pass filters.

To give some indication of the computational expense of executing STIMD in practice, Figure 3.8 evaluates the run times for different numbers of time samples and mixing matrices. Simulations and timing scores were produced by a 32 core Intel Xeon E5-2620v4 computer with 128 GB RAM.

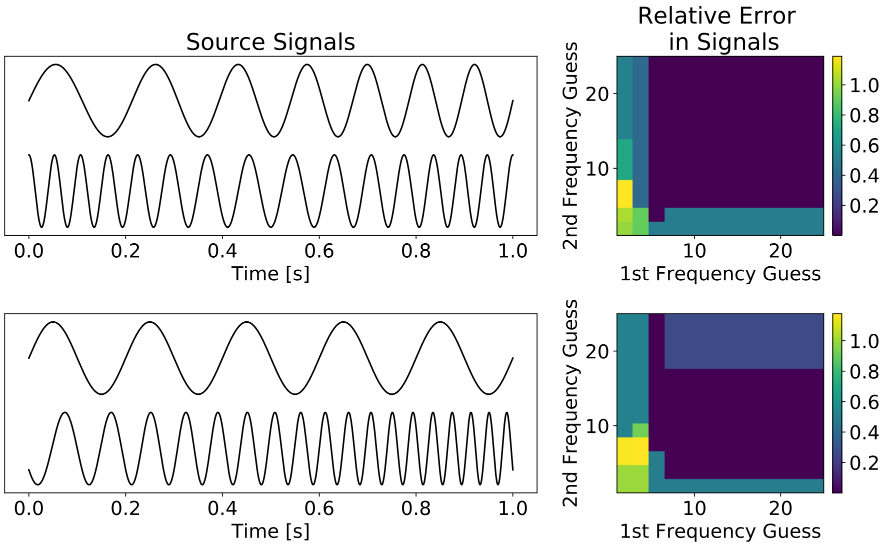


Figure 3.7: Two examples of the effect of initial conditions on the resulting STIMD modes. On the left are the source signals. STIMD is applied to the corresponding mixed signals. On the right is the relative error between the true source signals and the STIMD modes.

3.4 *Experiments on Real World Data*

In this section, we present results of STIMD on two real-world datasets in diverse domains.

3.4.1 Gravitational Waves from the LIGO Experiment

The Laser Interferometer Gravitational Wave Observatory (LIGO) is a recent Nobel prize-winning physics experiment with the goal of discovering and studying gravitational waves resulting from merging black holes [1]. The experiment consists of two detectors, one in Hanford, Washington and one in Livingston, Louisiana. These two detectors perform independent measurements, which can then be combined to increase confidence that a gravitational wave has been detected. These waves tend to be sinusoidal in nature, containing both frequency and amplitude modulation. The frequency modulation comes from the fact that as the black holes merge they rotate around each other with increasing frequency.

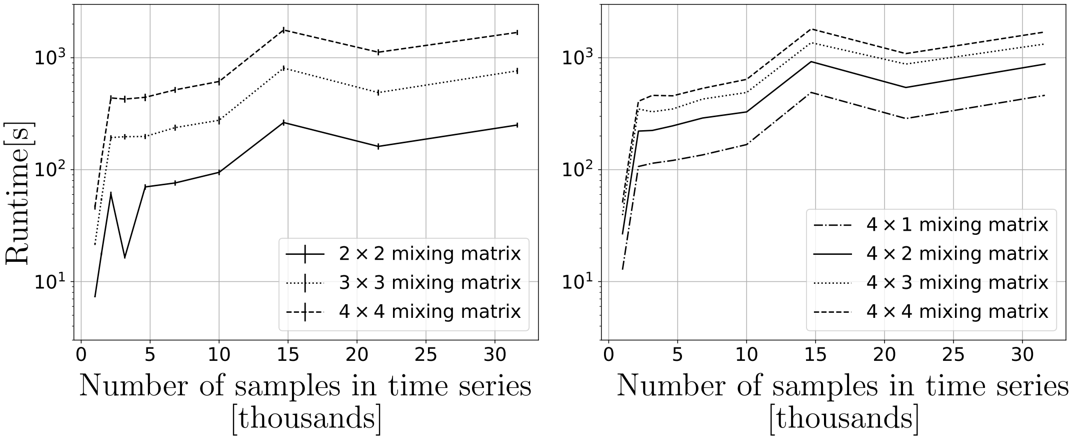


Figure 3.8: Runtimes for measured for different numbers of time samples and for different mixing matrix sizes.

The signals measured in the first gravitational wave detection are shown in Figure 3.7a. For their analysis, the LIGO collaboration computed a spectrogram (reproduced in Figure 3.7c). Here the chirp corresponding to the signal is readily apparent. In addition, there are clearly many other residual effects from using the Fourier transform. For example, there are clearly nonphysical high frequency components during earlier times.

Using initial guesses for the phases of $\theta_1(t)/2\pi = 50t$ and $\theta_2(t)/2\pi = 128t$, we obtained the IMF shown in Figure 3.7b, which clearly corresponds to the primary chirp seen in the data.

3.4.2 Neural Recordings from Rodent Hippocampus

As a second example, we analyze recordings of neural activity. This data is available at [79] and has been described previously [78]. It is well known that these local field potential signals in the rodent hippocampus contain rhythmic activity, with frequency and amplitude modulation [147].

Here we analyze a subset of the recordings from six neighboring electrodes, placed at

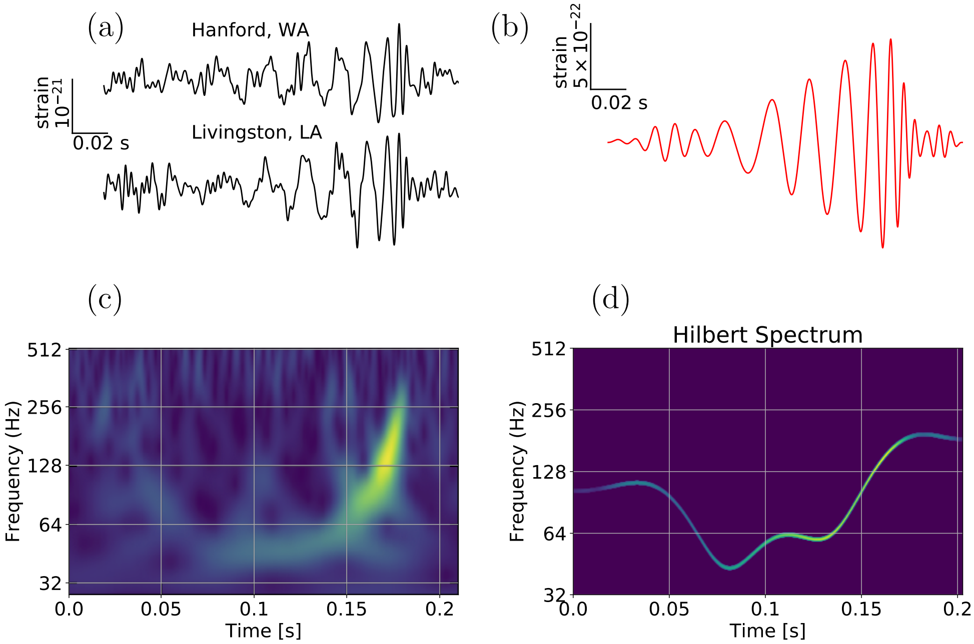


Figure 3.9: Application of STIMD to sample data from LIGO experiment. (a) Signals measured in LIGO experiment using two detectors. (b) STIMD modes. (c) Spectrogram computed in LIGO analysis. (d) Hilbert spectrum computed using STIMD modes. Yellow corresponds to frequencies of greater intensity while purple corresponds to frequencies of lower intensity.

equal spacing, over a 300ms period. Using peaks in the signals, we chose initial guesses for the phases $\theta_1(t) = 20\pi t - \pi/2$ and $\theta_2(t) = 200\pi t$, respectively. The mixing matrix $\boldsymbol{B}$ consequently has dimensions 6×2. In Figure 3.10, we show the observed signals (blue), the STIMD modes (red), the SVD modes (green), the ICA modes (orange), and the reconstructed signals (black) computed by multiplying the mixing matrix $\boldsymbol{B}$ and the STIMD modes. Note that in contrast to ICA and SVD, STIMD is able to separate the modes by their frequencies. It is unclear if the SVD or ICA decompositions are interpretable for this case. In addition, STIMD extracts spatial modes; in particular, the second STIMD mode, which corresponds to a high frequency wave, has higher amplitude in the central electrodes.

3.5 Conclusions

Principled mathematical methods for spatiotemporal decompositions are critically enabling for many emerging large-scale applications across the physical and biological sciences. Of particular interest is the ability to perform blind source separation on data generated from nonlinear and non-stationary dynamical processes. Our proposed STIMD mathematical architecture provides a compromise between the commonly used methods of ICA, SVD, and DMD. Specifically, STIMD is not constrained like DMD to model stationary signals and Fourier modes in time. However, we still make use of the constraint that our data must obey a certain set of dynamics commonly found in physical systems. In addition, we have the ability to compute a Hilbert spectrum and perform future state prediction, which is not guaranteed for ICA or SVD.

In this chapter, the STIMD method is applied on synthetic data to evaluate its feature extraction performance. The method leverages recent key innovations for signal processing from EMD and NMP. Indeed, by exploiting the IMF time constraint, the STIMD method frames an optimization problem that extracts meaningful features and low-rank modes from spatiotemporal data. We demonstrate the method on two real-world data sets, the LIGO experiment for the discovery of gravity waves and neural activity recordings from the rodent hippocampus. In both cases, the STIMD method produces a clean spatiotemporal decomposition with interpretable modes. We suggest STIMD as a method for data-driven discovery that may be widely applied to many domains with spatiotemporal data.

Algorithm 1 Nonlinear Matching Pursuit (NMP) Method

```
Input: measured signal $x(t)$ and initial guess for phase of IMF $\theta_0(t)$. For the 2nd and subsequent IMFs $x(t)$ is replaced with the remainder $r_k(t)$.
Output: IMF $s(t)$
$\theta^0 := \theta_0(t)$, $\eta := 0$
while $\eta < \lambda$ do
    $n = 0$
    while $n = 0$ or $\|\theta^{n+1} - \theta^n\|_2 > \epsilon_0$ do
        $a^{n+1}, b^{n+1} := \arg\min_{a,b} \|x - a\cos(\theta^n) - b\sin(\theta^n)\|_2^2$ s.t. $a(\theta), b(\theta) \in V(\theta, \eta)$ {Update $a$ and $b$}
        $\Delta\theta' := P_V(\theta; \eta)\left(\frac{d}{dt}\arctan\left(\frac{b(t)}{a(t)}\right)\right)$
        $\Delta\theta(t) := \int_0^t \Delta\theta'(s)ds$
        $\theta^{n+1} := \theta^n - \beta\Delta\theta$ where $\beta := \max\{\alpha \in [0,1] : \frac{d}{dt}(\theta^n - \alpha\Delta\theta) \geq 0\}$ {$\theta$ must be monotonic}
        $n := n + 1$
    end while
    $\eta = \eta + \Delta\eta$
end while
$a := \sqrt{(a^{n+1})^2 + (b^{n+1})^2}$
$\theta := \theta^{n+1}$
return $a\cos(\theta)$
```

Algorithm 2 STIMD

```
Input: measured signals $\boldsymbol{X} = [x_1, x_2, \ldots, x_n]^T$ and initial guess for phases of IMFs $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_r]^T$
Output: Matrix of IMFs $\boldsymbol{S} = [s_1^T, s_2^T, \ldots, s_r^T]^T$ and mixing matrix $\boldsymbol{B}$.
$\boldsymbol{R}_0 := \boldsymbol{X}$
for $j \in \{1, \ldots, r\}$ do
    $b_j := \arg\min_w \left\| w\mathrm{NMP}(w^T \boldsymbol{R}_{j-1}, \theta_i) - \boldsymbol{R}_{j-1} \right\|_2^2$ s.t. $\|w\|_2 = 1$
    $s_j := \mathrm{NMP}(b_j^T \boldsymbol{R}_{j-1})$
    $\boldsymbol{R}_j := \boldsymbol{R}_{j-1} - b_j s_j$
end for
$\boldsymbol{B} := [b_1, b_2, \ldots, b_r]$
$\boldsymbol{S} := [s_1^T, s_2^T, \ldots, s_r^T]^T$
return $\boldsymbol{B}, \boldsymbol{S}$
```

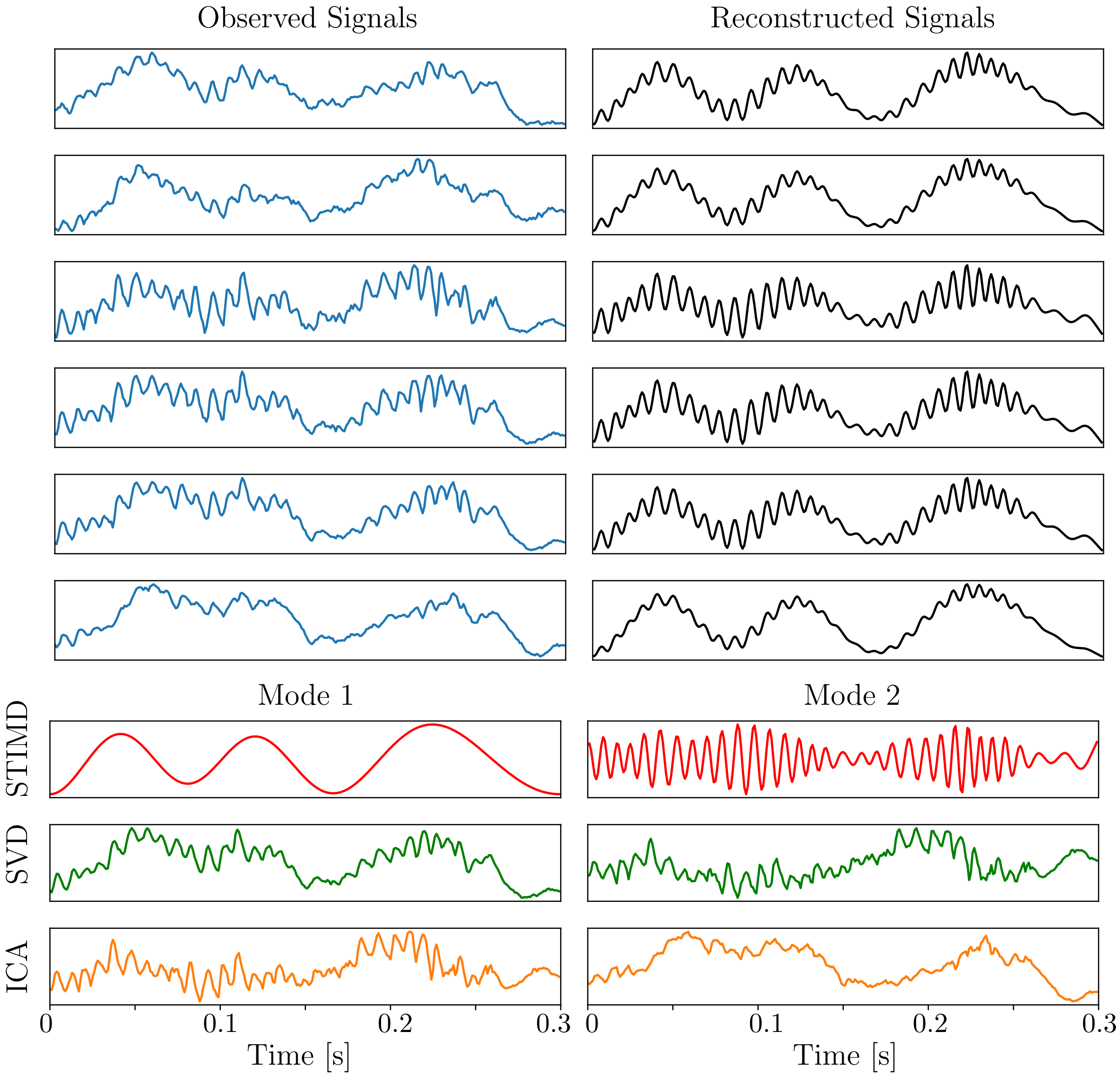


Figure 3.10: Application of STIMD to sample rodent neural recordings in hippocampus. Blue: Measured neural recordings. Red: STIMD modes. Green: SVD modes. Orange: ICA modes. Black: Signals reconstructed using STIMD modes and mixing matrix.

Chapter 4

TIME DELAY EMBEDDINGS AND THE FRENET-SERRET FRAME

4.1 Introduction

Dimensionality reduction techniques require that the dynamics are fully embedded in the data. However, in practice, the number of measurements necessary to embed the dynamics is unknown or difficult to measure. One method for addressing this is to use a delay embedding, in which the state at a given time t is constructed using the original state $x(t)$ and successive snapshots $x(t-\tau)$. Originally applied to the study of turbulence [203], delay embeddings have since been applied in a variety of domains including neuroscience [6], epidemiology [8, 86], and finance [223], due to several beneficial properties. Most notably, Takens showed that for a chaotic system given a sufficient number of time delays, the corresponding time delay embedding will be diffeomorphic to the original attractor [203]. In addition, recent work has shown that measurements of a time delay embedding is approximately low rank and hence well-approximated by a low dimensional representation [17]. Brunton *et al.* [30] combined time-delayed embeddings with the SVD and DMD to develop a method called the Hankel Alternative View of Koopman (HAVOK) to model the dynamics. Using this technique, Brunton found that the dynamics could be modeled using a linear model with an additional forcing term. Curiously, it was observed that for the Lorenz system the linear model took an antisymmetric structure with nonzero elements only on the off-diagonals.

To understand this tridiagonal antisymmetric structure, we can incorporate results from differential geometry. In the field of differential geometry a key goal is to find a set of intrinsic coordinates in which to measure the dynamics [53]. In general, the basis evolves in time and therefore cannot be accurately discovered using linear techniques, such as SVD. One

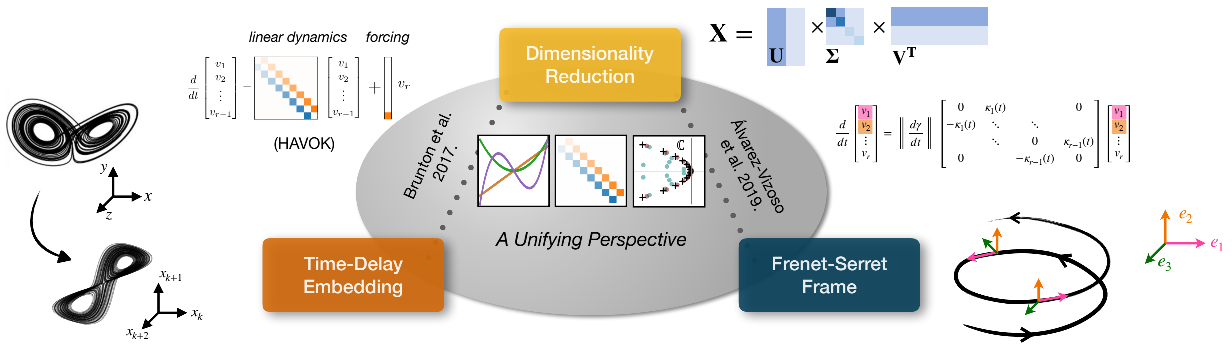


Figure 4.1: In Brunton *et al.*, the Hankel Alternative View of Koopman (HAVOK) method was proposed, which decomposes the dynamics of a system into a linear model plus an additional forcing term. In general this linear model is unconstrained. However, Brunton *et al.* observed that for the Lorenz system the dynamics take a specific form: namely the dynamics matrix is antisymmetric with nonzero elements only on the off-diagonal and the forcing term only in the last coordinate. In this work, we unify key results from dimensionality reduction, time-delay embedding, and the Frenet-Serret frame to explain this structure.

popular set of natural coordinates is the Frenet-Serret frame, which is formed by applying the Gram-Schmidt method to the derivatives of the trajectory $\dot{\boldsymbol{x}}(t), \ddot{\boldsymbol{x}}(t), \dddot{\boldsymbol{x}}(t), \ldots$ [150, 187, 196]. Using the properties of derivatives and orthogonality, it can be shown that the trajectory in this basis satisfies time varying-linear dynamics. Most notably, the dynamics matrix is antisymmetric with nonzero elements only on the off-diagonals. The nonzero terms have a clear geometric meaning, namely they correspond to the curvatures of the system.

A key goal for this work is to understand this tridiagonal antisymmetric structure. To do this, we utilize results in dimensionality reduction, time-delay embedding, and differential geometry. With these results, we develop a new method structured HAVOK (sHAVOK) which encourages the structure in the dynamics seen by Brunton *et al.* for the Lorenz system.

As illustrated in Figure 4.1, In this chapter we connect properties of Frenet-Serret frame with time delay embeddings and dimensionality reduction and show that together these three techniques provide a theoretical foundation for the structure of the HAVOK model observed for the Lorenz system. In Section 4.2, we review dimensionality reduction methods, time delay embeddings, and the Frenet-Serret frame. We additionally discuss current connections between these fields with particular focus on the HAVOK method and recent work by Vizoso *et al.*. In Section 4.3, we connect the SVD, time delay systems and the Frenet-Serret frame and show how together these three methods give rise to the tridiagonal antisymmetric structure seen in Figure 4.2. We then illustrate this theory on a synthetic example. In Section 4.4, we explore the limitations and requirements of the theory, giving recommendations for achieving this structure in practice. In Section 4.5, based on this theory we provide modifications to the HAVOK method. This set of modifications which we call structured HAVOK (sHAVOK) more closely aligns with the theory and consequently yields the tridiagonal antisymmetric form more in practice. We illustrate this method on three nonlinear synthetic examples and two real datasets, namely measurements of oscillations of a double pendulum and measles outbreak data, and show that sHAVOK yields a more stable and more accurate reconstruction.

4.2 Related Work

4.2.1 Dimensionality Reduction

Recent advancements in sensor and measurement technologies have led to a significant increase in the collection of time-series data from complex, spatio-temporal systems. Although such data is typically high dimensional, in many cases it can be well approximated with a low dimensional representation. One key goal is to learn the underlying structure of this data. Although there are many data-driven dimensionality reduction methods, here we focus on linear dimensionality reduction techniques. In particular, given a data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, the goal of these techniques is to decompose $\boldsymbol{X}$ into the product of matrices

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\intercal, \tag{4.1}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times k}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times k}$ are low rank ($k < \min(m,n)$). The problem of determining $\boldsymbol{U}$ and $\boldsymbol{V}$ is highly underdetermined. Thus given different assumptions, we obtain different results for these two matrices. Here we review two popular linear dimensionality reduction techniques: *singular value decomposition* (SVD) and *dynamic mode decomposition* (DMD). Both of these methods are key elements of the HAVOK algorithm, and will play a key role in determining the underlying tridiagonal antisymmetric structure in Figure 4.2.

Singular Value Decomposition (SVD)

The singular value decomposition is one of the most popular dimensionality reduction methods, which has been applied in a wide range of applications, including genomics [4], physics [178], and image processing [145]. Given the data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, the SVD decomposes $\boldsymbol{X}$ into the product of three matrices,

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\intercal},$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ are unitary matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ is a diagonal matrix with nonnegative entries [73, 109]. We will denote the ith columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ as $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, respectively. Typically the diagonal elements of $\boldsymbol{\Sigma}$, σ_i, are written in descending order and are defined as the singular values of $\boldsymbol{X}$. The rank of the data is defined to be r, which equals the number of nonzero singular values. Consider the low rank matrix approximation

$$\boldsymbol{X}_k = \sum_{j=1}^{k} \boldsymbol{u}_j \sigma_j \boldsymbol{v}_j^T,$$

with $k \leq r$. An important property of $\boldsymbol{X}_k$ is that it is the best rank k approximation to $\boldsymbol{X}$ in the least squares sense. More precisely, for a rank k matrix $\boldsymbol{Y}$ with $k \leq r$, then

$$\|\boldsymbol{X} - \boldsymbol{Y}\|$$

is minimized by $\boldsymbol{Y} = \boldsymbol{X}_k$ with respect to both the l_2 and Frobenius norms. In particular, the relative error in this rank k approximation (using the l_2 norm) is

$$\frac{\|\boldsymbol{X} - \boldsymbol{X}_k\|_{l_2}}{\|\boldsymbol{X}\|_{l_2}} = \frac{\sigma_{k+1}}{\sigma_1}.$$

From this, we immediately see that if the singular values decay rapidly, ($\sigma_{j+1} \ll \sigma_j$), then $\boldsymbol{X}_k$ will be a good low-rank approximation to our original data $\boldsymbol{X}$. This makes the SVD a popular tool for compressing data and is the underlying algorithm for *principal component analysis* (PCA).

Dynamic Mode Decomposition (DMD)

The dynamic mode decompostion is an alternative linear dimensionality reduction technique which has become a popular tool for modeling dynamical systems in such diverse fields as fluid mechanics [173, 181], neuroscience [29], and computer vision [77, 58]. Like the SVD, we begin with a data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$. Here we assume that our data is generated by an unknown dynamical system. In particular, the columns of $\boldsymbol{X}$, $\boldsymbol{x}_k$, are time snapshots related by some, possibly nonlinear, function $\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k)$. The goal of DMD is to determine the best-fit linear operator $\boldsymbol{A} : \mathbb{R}^m \to \mathbb{R}^m$ such that

$$\boldsymbol{x}(t_{k+1}) \approx \boldsymbol{A}\boldsymbol{x}(t_k).$$

If we define the two time-shifted matrices,

$$\boldsymbol{X}_1^{n-1} = \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{x}(t_1) & \boldsymbol{x}_2(t_2) & \cdots & x(t_{n-1}) \\ | & | & \cdots & | \end{bmatrix}, \text{ and } \boldsymbol{X}_2^n = \begin{bmatrix} | & | & \cdots & | \\ x(t_2) & x(t_3) & \cdots & x(t_n) \\ | & | & \cdots & | \end{bmatrix},$$

then we can equivalently define $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ to be the operator such that

$$\boldsymbol{X}_2^n \approx \boldsymbol{A}\boldsymbol{X}_1^{n-1}.$$

To find $\boldsymbol{A}$, we must then solve the minimization problem

$$\boldsymbol{A} = \min_{\boldsymbol{A'}} \left\| \boldsymbol{X}_2^n - \boldsymbol{A'}\boldsymbol{X}_1^{n-1} \right\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. A unique solution to this problem can be obtained using the exact DMD method and the Moore-Penrose pseudo-inverse $\boldsymbol{A} = \boldsymbol{X}_2^n \left(\boldsymbol{X}_1^{n-1}\right)^\dagger$ [213, 123]. For noisy data, we can use more robust methods such as optimized DMD which de-biases the DMD eigenvalues and eigenvectors [12].

One key benefit of DMD is that it builds an explicit temporal model, which allows for future state prediction. Defining $\{\lambda_j\}$ and $\{\boldsymbol{v}_j\}$ to be the eigenvalues and eigenvectors of $\boldsymbol{A}$, respectively, then

$$\boldsymbol{x}(t_k) = \sum_{j=1}^{r} \boldsymbol{v}_j e^{\omega_j t_k}, \tag{4.2}$$

where $\omega_j = \ln(\lambda_j)/\Delta t$ and the eigenvectors are normalized such that $\sum_{j=1}^{r} \boldsymbol{v}_j = \boldsymbol{x}(t_1)$. Thus, to compute the state at an arbitrary time t, simply evaluate (4.2) at that time. Additionally, letting $\boldsymbol{v}_j$ be the columns of $\boldsymbol{U}$ and $\{\exp \omega_j t_k \text{ for } k = 1, \dots r\}$ be the columns of $\boldsymbol{V}$, then we can express data in the form of (4.1).

4.2.2 Time Delay Embedding

Suppose we have a dynamical system

$$\frac{d\boldsymbol{\xi}}{dt} = \boldsymbol{F}(\boldsymbol{\xi}),$$

where $\boldsymbol{\xi}(t) \in \mathbb{R}^m$ for some unknown nonlinear differential equation. Typically, we measure some possibly nonlinear, function $\boldsymbol{x}(\boldsymbol{\xi}) \in \mathbb{R}^k$ at discrete time points $t = 0, \Delta t, \dots, q\Delta t$. In general, the dimensionality of the underlying dynamics is unknown, and the choice of measurements are limited due to constraints. Consequently, it is difficult to know whether the measurements $\boldsymbol{x}$ are sufficient for modeling the system. For example, k may be smaller than m. One of the primary goals of data-driven methods is to discover a good set of observables, or coordinates, in which to measure the system. Choosing a good set of observables is critical for analyzing the underlying dynamical system and performing accurate future state prediction.

In this work we are primarily interested in the case of $k = 1$; in other words, we have only a one dimensional time series for the system. We can construct an embedding of our system using successive time delays $x(t - \tau)$. Given a single measurement of our dynamical system

$x(t) \in \mathbb{R}$, for $t = 0, \Delta t, \dots q\Delta t$, we can form the Hankel matrix $\boldsymbol{H} \in \mathbb{R}^{m \times n}$ by stacking time shifted snapshots [155],

$$\boldsymbol{H} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & \cdots & x_n \\ x_2 & x_3 & x_4 & x_5 & \cdots & x_{n+1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m+1} & x_{m+2} & x_{m+3} & \cdots & x_q \end{bmatrix}. \tag{4.3}$$

With this representation we can think of our data as an m-dimensional trajectory measured over n snapshots in time.

There are several benefits of using time delay embeddings. Most notably, given a chaotic attractor, Taken's embedding theorem states that a time delay embedding of the system will be diffeomorphic to the original attractor [203]. In addition, recent results have shown that time delay matrices are guaranteed to have strongly decaying singular value spectra. Beckerman *et al.* [17] prove the following theorem.

Theorem 4.2.1. *Let $\boldsymbol{H}_n \in \mathbb{R}^{n \times n}$ be a positive definite Hankel matrix, with singular values $\sigma_1, \dots, \sigma_n$. Then $\sigma_k \leq C\rho^{-k/\log n}\sigma_1$ for constants C and ρ and for $k = 1, \dots, n$.*

Equivalently, $\boldsymbol{H}_n$ can be approximated up to an accuracy of $\epsilon \left\|\boldsymbol{H}_n\right\|_2$ by a rank $O(\log n \log 1/\epsilon)$ matrix. With this in mind we see that $\boldsymbol{H}_n$ can be well-approximated by a low-rank matrix.

Many methods have been developed to take advantage of this structure, including the *eigensystem realization algorithm* (ERA) [114], *singular spectrum analysis* (SSA) [28], and nonlinear Laplacian spectrum analysis [67]. In addition, this structure has also been incorporated into neural network architectures [217]. Here we will be primarily focused on HAVOK [30].

4.2.3 HAVOK (Dimensionality Reduction and Time Delay Embeddings)

The Hankel Alternative View of Koopman (HAVOK), originally developed by Brunton *et al.* [30], combines time delay embeddings and dimensionality reduction to construct a low

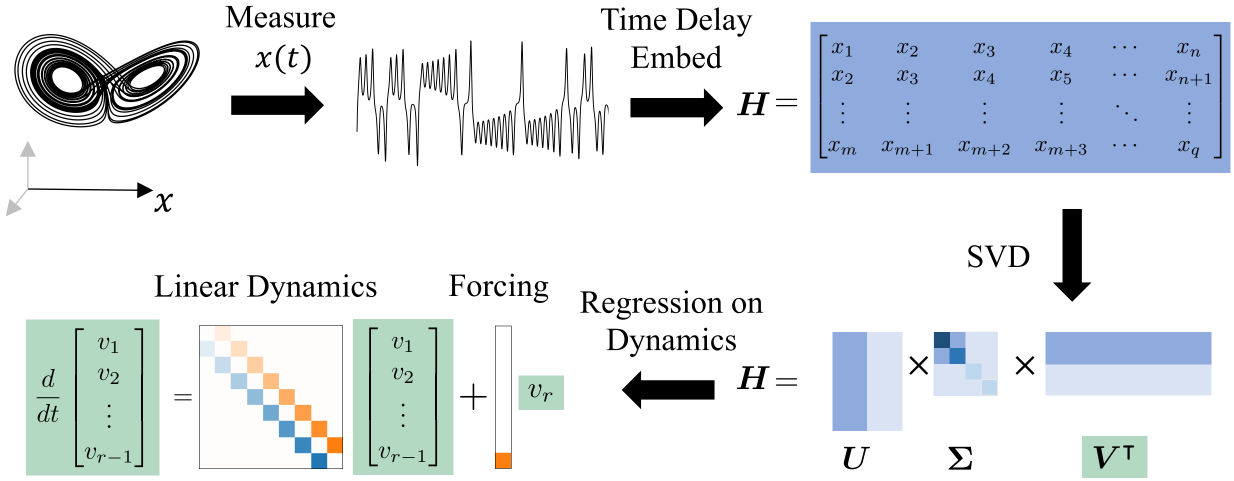


Figure 4.2: Outline of steps in HAVOK method. First, given a dynamical system a single variable $x(t)$ is measured. Time-shifted copies of $x(t)$ are stacked to form a Hankel matrix $\boldsymbol{H}$. The singular value decomposition (SVD) is applied to $\boldsymbol{H}$, producing a low dimensional representation $\boldsymbol{V}$. The dynamic mode decomposition (DMD) is then applied to $\boldsymbol{V}$ to form a linear dynamical model and a forcing term.

dimensional model of a dynamical system. Most notably, the method combines time delays with SVD to learn ”good” measurements of the system and estimate its intrinsic dimensionality. Additionally, the HAVOK model is simple, consisting of a linear model and a forcing term which can be used for short term forecasting.

In this method, we begin with a one dimensional time series $x(t)$ for $t = 0, \Delta t, \ldots, q\Delta t$. We construct a higher dimensional representation using time delay embeddings, producing a Hankel matrix $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ as in (4.3). We compute the singular value decomposition of $\boldsymbol{H}$,

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal.$$

If $\boldsymbol{H}$ is sufficiently low rank (with rank r), then we need only consider the reduced SVD,

$$\boldsymbol{H}_r = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\intercal,$$

where $\boldsymbol{U}_r \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V}_r \in \mathbb{R}^{n\times r}$ are orthogonal matrices and $\boldsymbol{\Sigma}_r \in \mathbb{R}^{r\times r}$ is diagonal. Now $\boldsymbol{V}_r^\intercal = \boldsymbol{\Sigma}_r^{-1}\boldsymbol{U}_r^\intercal \boldsymbol{H}_r$. We can think of

$$\boldsymbol{V}_r^\intercal = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix}$$

as a lower dimensional representation of our high dimensional trajectory. For quasi-periodic systems, the SVD decomposition of the Hankel matrix results in *principal component trajectories* (PCT) [56], which allows for the reconstruction of dynamical trajectories in terms of periodic orbits. To discover the linear dynamics, we apply DMD. In particular, we construct the time shifted matrices,

$$\boldsymbol{V}_1 = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_{n-1} \end{bmatrix} \text{ and } \boldsymbol{V}_2 = \begin{bmatrix} \boldsymbol{v}_2 & \boldsymbol{v}_3 & \cdots & \boldsymbol{v}_n \end{bmatrix}.$$

We can compute the best linear approximation $\hat{\boldsymbol{A}}$ such that $\boldsymbol{V}_2 = \hat{\boldsymbol{A}}\boldsymbol{V}_1$, where $\hat{\boldsymbol{A}} = \boldsymbol{V}_2\boldsymbol{V}_1^\dagger$. $\boldsymbol{V}_1^\dagger$ denotes the pseudoinverse of $\boldsymbol{V}_1$ [156]. In the continuous case,

$$\dot{\boldsymbol{V}} = \boldsymbol{A}\boldsymbol{V},$$

which is related to first order in Δt to the discrete case by

$$\boldsymbol{A} \approx \left(\hat{\boldsymbol{A}} - \boldsymbol{I}\right)/\Delta t.$$

For a general nonlinear dynamical system this linear model will yield a poor reconstruction. To remedy this Brunton *et al.* [30] proposed a nonlinear forcing term:

$$\dot{\boldsymbol{v}}(t) = \boldsymbol{A}\boldsymbol{v}(t) + \boldsymbol{B}v_r(t)$$

where $\boldsymbol{v}(t) \in \mathbb{R}^{r-1}$, $\boldsymbol{A} \in \mathbb{R}^{r-1\times r-1}$, and $\boldsymbol{B} \in \mathbb{R}^{r-1}$. In this case, $\boldsymbol{V}_2$ is defined as columns 2 to n of the SVD singular vectors with an $r-1$ rank truncation $\boldsymbol{V}_{r-1}^\intercal$. $\hat{\boldsymbol{A}} \in \mathbb{R}^{r-1\times r-1}$ and $\hat{\boldsymbol{B}} \in \mathbb{R}^{r-1\times 1}$ are computed as $\left[\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}\right] = \boldsymbol{V}_2\boldsymbol{V}_1^\dagger$. The continuous analog of $\hat{\boldsymbol{B}}$, $\boldsymbol{B}$, is computed by $\boldsymbol{B} \approx (\hat{\boldsymbol{B}} - \boldsymbol{I})/\Delta t$.

Brunton *et al.* [30] showed this to be a succesful model for a variety of systems, including a double pendulum and switchings of Earth's magnetic field. Most notably, it was discovered that for the Lorenz system [133], the HAVOK model yielded a particular structure. Namely, the dynamics matrix was antisymmetric with nonzero elements only on the superdiagonal and subdiagonal (Figure 4.2).

Much work has been done to study the properties of the HAVOK method. Namely, Arbabi *et al.* [7] showed that under certain properties (about ergodicity) in the limit of an infinite number of time delays ($m \to \infty$) $\boldsymbol{A}$ converges to the Koopman operator [7]. Bozzo *et al.* [27] showed that in a similar limit, for periodic data HAVOK converges to the temporal discrete Fourier transform. Kamb *et al.* [118] connects HAVOK to the use of convolutional coordinates. The primary goal of this work is to connect the HAVOK work to the concept of curvature in differential geometry, and with these new insights, improve the HAVOK algorithm.

4.2.4 The Frenet-Serret Coordinate Frame

Suppose we have a smooth curve $\boldsymbol{\gamma}(t) \in \mathbb{R}^m$ measured over some time interval $t \in [a, b]$. As before, we would like to determine a good set of coordinates in which to represent our data. In the case of DMD or SVD, the basis discovered, corresponding to the spatial modes of the data, is constant in time. However, for many systems it is sometimes natural to express both the coordinates and basis as functions of time [9, 142]. One popular method for developing this noninertial frame is the Frenet-Serret coordinate system, which has been applied in a wide range of fields, including robotics [46, 166], aerodynamics [160], and general relativity [25, 107].

Let us assume that $\boldsymbol{\gamma}(t)$ has r nonzero continuous derivatives, $\boldsymbol{\gamma}', (t), \boldsymbol{\gamma}''(t), \boldsymbol{\gamma}'''(t), \ldots$ and $\|\boldsymbol{\gamma}'(t)\| \neq \mathbf{0}$ for all t. Using the Gram-Schmidt algorithm, we can form the orthonormal basis, $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_r$,

$$\begin{aligned}
\boldsymbol{e}_1(t) &= \frac{\boldsymbol{\gamma}'(t)}{\|\boldsymbol{\gamma}'(t)\|} \\
\boldsymbol{e}_2(t) &= \frac{\boldsymbol{\gamma}''(t) - \langle \boldsymbol{\gamma}''(t), \boldsymbol{e}_1(t)\rangle \boldsymbol{e}_1(t)}{\|\boldsymbol{\gamma}''(t) - \langle \boldsymbol{\gamma}''(t), \boldsymbol{e}_1(t)\rangle \boldsymbol{e}_1(t)\|} \\
&\vdots \\
\boldsymbol{e}_r(t) &= \frac{\boldsymbol{\gamma}^{(r)}(t) - \sum_{k=1}^{r-1}\langle \boldsymbol{\gamma}^{(r)}(t), \boldsymbol{e}_k(t)\rangle \boldsymbol{e}_k(t)}{\left\|\boldsymbol{\gamma}^{(r)}(t) - \sum_{k=1}^{r-1}\langle \boldsymbol{\gamma}^{(r)}(t), \boldsymbol{e}_k(t)\rangle \boldsymbol{e}_k(t)\right\|}.
\end{aligned} \tag{4.4}$$

We choose $r \leq m$ so that these vectors are linearly independent and hence form an orthogonal basis. The expression $\langle \cdot, \cdot \rangle$ denotes an inner product.

We are interested in the evolution of this basis. To be more precise, let us define the matrix $\boldsymbol{Q}(t) = [\boldsymbol{e}_1(t), \boldsymbol{e}_2(t), \ldots, \boldsymbol{e}_r(t)]^\intercal \in \mathbb{R}^{r \times m}$. We are interested in the matrix $\boldsymbol{K}(t)$ such that

$$\frac{d\boldsymbol{Q}}{dt} = \|\boldsymbol{\gamma}'(t)\| \, \boldsymbol{K}(t)\boldsymbol{Q}, \tag{4.5}$$

where $\boldsymbol{K}(t) \in \mathbb{R}^{r \times r}$. To understand the structure of $\boldsymbol{K}(t)$ we can derive two key properties:

1. $\boldsymbol{K}_{i,k}(t) = -\boldsymbol{K}_{k,i}(t)$ (antisymmetry):

   Proof. Since $r \leq m$, then by construction $\boldsymbol{Q}(t)$ is a unitary matrix with $\boldsymbol{Q}\boldsymbol{Q}^\intercal = \boldsymbol{I}$. Taking the derivative with respect to t, we find that $\frac{d\boldsymbol{Q}}{dt}\boldsymbol{Q}^T + \boldsymbol{Q}\frac{d\boldsymbol{Q}^\intercal}{dt} = 0$, or equivalently

   $$\frac{d\boldsymbol{Q}}{dt}\boldsymbol{Q}^\intercal = -\left(\frac{d\boldsymbol{Q}}{dt}\boldsymbol{Q}^\intercal\right)^\intercal.$$

   Since $\boldsymbol{Q}$ is unitary, then $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^\intercal$, and hence

   $$\boldsymbol{K}(t) = \frac{1}{\|\boldsymbol{\gamma}(t)\|}\frac{d\boldsymbol{Q}}{dt}\boldsymbol{Q}^\intercal,$$

   from which we immediately see that $\boldsymbol{K}(t) = -\boldsymbol{K}(t)^\intercal$. □

2. $\boldsymbol{K}_{i,k}(t) = 0$ for $k \geq i + 2$:

   We first note that since $\boldsymbol{e}_i(t) \in \operatorname{span}\{\boldsymbol{\gamma}'(t), \ldots, \boldsymbol{\gamma}^i(t)\}$, then its derivative must satisfy $\boldsymbol{e}_i'(t) \in \operatorname{span}\{\boldsymbol{\gamma}'(t), \ldots, \boldsymbol{\gamma}^{(i+1)}(t)\}$. Now by construction using the Gram-Schmidt method, $\boldsymbol{e}_k$ is orthogonal to span$\{\boldsymbol{\gamma}'(t), \ldots, \boldsymbol{\gamma}^{(i+1)}(t)\}$ for $k \geq i+2$. Since $\boldsymbol{e}_i'(t)$ is in the span of this set, then $\boldsymbol{e}_k$ must be orthogonal to $\boldsymbol{e}_i'$ for $k \geq i+2$. Thus, $\boldsymbol{K}_{i,k}(t) = \langle \boldsymbol{e}_i'(t), \boldsymbol{e}_k \rangle = 0$ for $k \geq i+2$.

With these two constraints, $\boldsymbol{K}(t)$ takes the form,

$$\boldsymbol{K}(t) = \begin{bmatrix} 0 & \kappa_1(t) & & 0 \\ -\kappa_1(t) & \ddots & \ddots & \\ & \ddots & 0 & \kappa_{r-1}(t) \\ 0 & & -\kappa_{r-1}(t) & 0 \end{bmatrix}. \tag{4.6}$$

Thus $\boldsymbol{K}(t)$ is antisymmetric with nonzero elements only along the superdiagonal and subdiagonal. The values $\kappa_1(t), \ldots, \kappa_{r-1}(t)$ are defined to be the curvatures of the trajectory. From a geometric perspective, $\boldsymbol{e}_1(t), \ldots, \boldsymbol{e}_r(t)$ form an instantaneous (local) coordinate frame which moves with the trajectory. The curvatures define how quickly this frame changes with time. If the trajectory is a straight line the curvatures are all zero, if the trajectory lies completely in the plane $\kappa_2, \ldots, \kappa_r$ are all zero and so on. If κ_1 is constant and nonzero, while all other curvatures are zero, then the trajectory lies on a circle. If κ_1 and κ_2 are constant and nonzero with all other curvatures zero, then the trajectory lies on a helix. Comparing the structure of (4.6) to Figure 4.2 we immediately see similarity. Over the following sections we will shed light on this connection.

4.2.5 SVD and Curvature

Before introducing our innovations, we review recent work connecting the SVD to the Frenet-Serret frame, which will be critical for understanding the structure seen in Figure 4.2. As previously stated, given time series data, the SVD constructs an orthonormal basis which is fixed in time. The Frenet-Serret frame on the other hand constructs an orthonormal basis which moves with the trajectory. In recent work, Alvarez-Vizoso *et al.* [5] showed that these frames are related. In particular, the Frenet-Serret frame converges to the SVD frame in the limit as the time interval of the trajectory goes to zero.

To understand this further, consider a trajectory $\boldsymbol{\gamma}(t) \in \mathbb{R}^m$ as described in Section 4.2.4. If we assume that our measurements are from a small neighborhood $t \in (-\epsilon, \epsilon)$ (where $\epsilon \ll 1$), then $\boldsymbol{\gamma}(t)$ is well-approximated by its Taylor expansion,

$$\boldsymbol{\gamma}(t) - \boldsymbol{\gamma}(0) = \boldsymbol{\gamma}'(0)t + \frac{\boldsymbol{\gamma}''(0)}{2}t^2 + \frac{\boldsymbol{\gamma}'''(0)}{6}t^3 + \cdots$$

Writing this in matrix form we have that

$$\boldsymbol{\gamma}(t) - \boldsymbol{\gamma}(0) = \underbrace{\begin{bmatrix} | & | & | & | \\ \boldsymbol{\gamma}'(0) & \boldsymbol{\gamma}''(0) & \boldsymbol{\gamma}'''(0) & \cdots \\ | & | & | & | \end{bmatrix}}_{\boldsymbol{\Gamma}} \underbrace{\begin{bmatrix} 1 & & & \\ & \frac{1}{2} & & \\ & & \frac{1}{6} & \\ & & & \ddots \end{bmatrix}}_{\boldsymbol{\Sigma}} \underbrace{\begin{bmatrix} - & t & - \\ - & t^2 & - \\ - & t^3 & - \\ - & \vdots & - \end{bmatrix}}_{\boldsymbol{T}^{\intercal}}. \tag{4.7}$$

Recall, one key property of the singular value decomposition is that the rth rank truncation in the expansion is the best rank r approximation to the data in the least squares sense. Since $\epsilon \ll 1$, then each subsequent term in this expansion will be much smaller than the previous term

$$\|\boldsymbol{\gamma}'(0)t\|_2 \ll \left\|\frac{\boldsymbol{\gamma}''(0)}{2}t^2\right\|_2 \ll \left\|\frac{\boldsymbol{\gamma}'''(0)}{6}t^3\right\|_2 \ll \ldots, \tag{4.8}$$

From this, we see the that expansion in (4.7) is strongly related to the SVD. However, in the SVD we have the constraint that the $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices are orthogonal, while for the Taylor expansion $\boldsymbol{\Gamma}$ and $\boldsymbol{T}$ have no such constraint. However, Alvarez *et al.* [5] show that in the limit as $\epsilon \to 0$, then $\boldsymbol{U}$ is the result of applying the Gram-Schmidt process to the columns of $\boldsymbol{\Gamma}$, and $\boldsymbol{V}$ is the result of applying the Gram-Schmidt process to the columns of $\boldsymbol{T}$. Comparing this to above we see that

$$\boldsymbol{U} = \begin{bmatrix} | & | & | & | \\ \boldsymbol{e}_1(0) & \boldsymbol{e}_2(0) & \boldsymbol{e}_3(0) & \cdots \\ | & | & | & | \end{bmatrix} \text{ and } \boldsymbol{V} = \begin{bmatrix} | & | & | & | \\ p_1(t) & p_2(t) & p_3(t) & \cdots \\ | & | & | & | \end{bmatrix},$$

where $\boldsymbol{e}_1(t), \boldsymbol{e}_2(t), \ldots, \boldsymbol{e}_r(t)$ is the basis for the Frenet-Serret frame defined in (4.4) and

$$p_i(t) = \frac{t^i - \sum_{k=1}^{i-1} \langle t^i, p_k(t)\rangle\, p_k(t)}{\left\|t^i - \sum_{k=1}^{i-1} \langle t^i, p_k(t)\rangle\, p_k(t)\right\|} \text{ for } i = 1, 2, 3, \ldots \tag{4.9}$$

We note that $p_i(t)$ are a set of orthogonal polynomials independent of the dataset. In this limit, the curvatures depend solely on the singular values,

$$\kappa_j(t) = \sqrt{a_j}\frac{\sigma_{j+1}}{\sigma_1(t)\sigma_j(t)}, \text{ where } a_{j-1} = \left(\frac{j}{j+(-1)^j}\right)^2 \frac{4j^2-1}{3}.$$

4.3 Unifying Singular Value Decomposition, Time Delay Embeddings, and the Frenet-Serret Frame

The method HAVOK utilizes time-delay embeddings, SVD, and DMD to construct a linear model of a dynamical system with an additional forcing term. Most notably, in the case of the Lorenz system the linear model takes a particular tridiagonal antisymmetric form seen in Figure 4.2. In Section 4.3.1, we combine key results about the Frenet-Serret frame, time delays, and the singular value decomposition to explain this structure. In Section 4.3.2, we follow this theory with a simple synthetic example to illustrate this process. In this perspective, we must define a set of orthogonal polynomials. In Section 4.3.3, we explicitly describe these polynomials, comparing their properties to the Legendre polynomials.

4.3.1 Connecting SVD, Time Delay Embeddings, and Frenet-Serret Frame

In this section we provide a new perspective on the antisymmetric tridiagonal structure which appears for the Lorenz system. In particular, we will show that the properties of time-delay matrices, the Frenet-Serret frame, and the SVD combined with the results of [5] induce this structure. This perspective is summarized in Figure 4.3.

Like in Section 4.2.3, consider a one-dimensional time series $x(t) \in \mathbb{R}$ for $t = 0, \Delta t, \dots q\Delta t$ and the corresponding time delay embedding $\boldsymbol{H} \in \mathbb{R}^{m \times n}$. In particular, this embedding can be thought of as n snapshots of a trajectory $\boldsymbol{h}(t) \in \mathbb{R}^m$ for $t = 0, \dots, n\Delta t$.

Suppose that this trajectory occurs over a short interval in time. By Section 4.2.5, since the trajectory occurs over a short period of time, then the SVD of the data approximates the Frenet-Serret frame over that interval. Performing DMD (regression on the dynamics) on the Frenet-Serret frame at that instant will yield the antisymmetric tridiagonal dynamics.

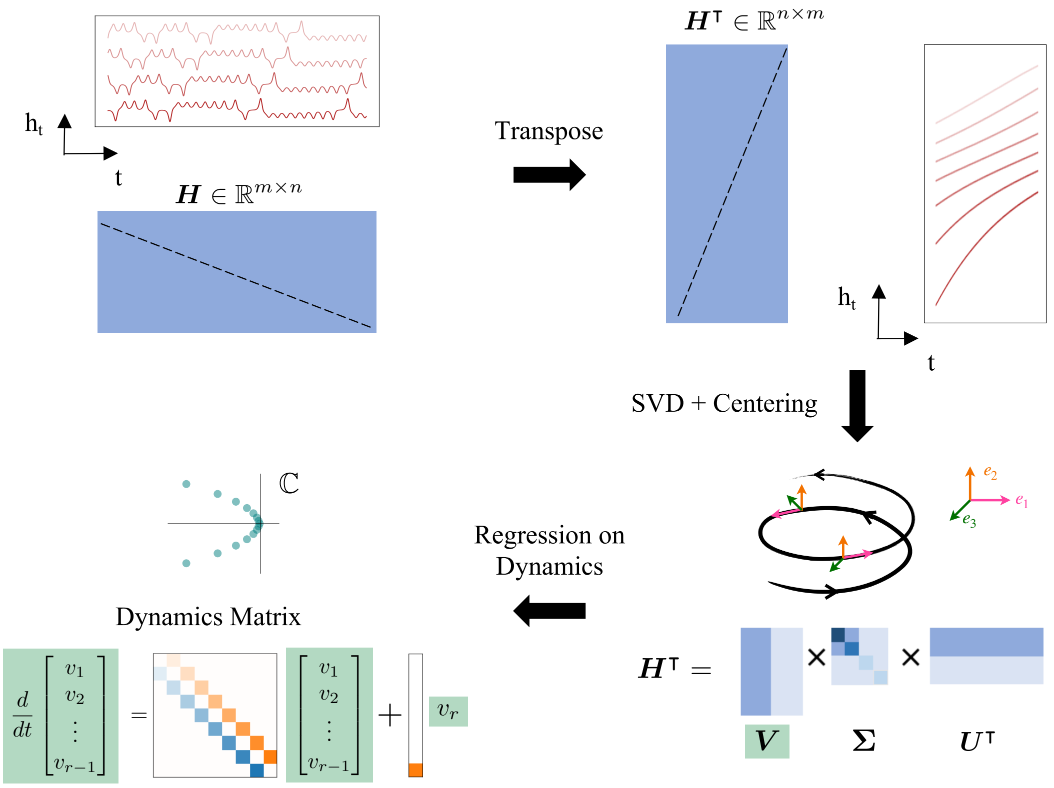


Figure 4.3: Illustration of how antisymmetric linear model arises from time delay data. Starting with a one dimensional time-series we construct a $m \times n$ Hankel matrix using time-shifted copies of the data. Assume that $n \gg m$, in which case $\boldsymbol{H}$ can be thought of as an m dimensional trajectory over a long period (n snapshots in time). Due to the properties of time-delay matrices, the transpose of $\boldsymbol{H}$, may be thought of as a high dimensional (n dimensional) trajectory over a short period (m snapshots) in time. With this interpretation, the results of [5] show that the singular vectors of $\boldsymbol{H}$ after applying centering yield the Frenet-Serret frame. Applying the dynamic mode decomposition (DMD) to the Frenet-Serret frame yields the tridiagonal antisymmetric linear model with an additional forcing term, which is nonzero only in the last component.

By the properties of the Frenet-Serret frame, this dynamics matrix will have exactly the structure seen in the Lorenz system.

One key assumption in this argument is that the trajectory $\boldsymbol{h}(t)$ is measured over a short time interval $n \ll m$. However, for data such as the Lorenz attractor this assumption does *not* hold. In fact, for the Lorenz system originally studied in Brunton *et al.* [30] $n \gg m$, and each component of $\boldsymbol{h}(t) \in \mathbb{R}^m$ is measured over a long period of time, covering several cycles around the fixed point. Instead of thinking of the trajectory defined by the columns of $\boldsymbol{H}$, we may choose to consider the trajectory $\boldsymbol{h}(t) \in \mathbb{R}^n$ defined by the columns of $\boldsymbol{H}^\intercal$. Like the trajectory defined by $\boldsymbol{H}$, this trajectory consists of time-shifted snapshots of $x(t)$. Since $m \ll n$, this trajectory does occur over a short period of time. From this perspective, applying the SVD to the singular vectors of $\boldsymbol{H}^\intercal$ will form the Frenet-Serret frame and regression on these vectors will yield a corresponding antisymmetric tridiagonal dynamics matrix.

More formally, let's begin with the time series $x(t)$ for $t = 0, \Delta t, \dots, q\Delta t$. We construct a time delay embedding $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ which we will assume is a short and fat matrix ($m \ll n$). Now let's consider the transpose $\boldsymbol{H}^\intercal \in \mathbb{R}^{n\times m}$. The columns of this matrix can be thought of as a trajectory $\boldsymbol{h}(t) \in \mathbb{R}^n$ for $t = 0, \Delta t, \dots, (m-1)\Delta t$. For simplicity, we redefine the origin of time so that $\boldsymbol{h}(t)$ spans $t = -(m-1)\Delta t/2, \dots, 0, \dots (m-1)\Delta t/2$, and we denote $\boldsymbol{h}(i\Delta t)$ as $\boldsymbol{h}_i$. In this form,

$$\boldsymbol{H}^\intercal = \begin{bmatrix} | & \cdots & | & \cdots & | \\ \boldsymbol{h}_{(-m+1)/2} & \cdots & \boldsymbol{h}_0 & \cdots & \boldsymbol{h}_{(m-1)/2} \\ | & \cdots & | & \cdots & | \end{bmatrix}.$$

Subtracting the central column $\boldsymbol{h}_0$ of $\boldsymbol{H}^\intercal$ from $\boldsymbol{H}^\intercal$ (or equivalently the central row of $\boldsymbol{H}$) yields the centered matrix

$$\bar{\boldsymbol{H}}^\intercal = \boldsymbol{H}^\intercal - \boldsymbol{h}_0\mathbf{1}^\intercal.$$

We can express $\boldsymbol{h}_i$ as a Taylor expansion about $\boldsymbol{h}_0$,

$$\boldsymbol{h}_i - \boldsymbol{h}_0 = \boldsymbol{h}_0' i\Delta t + \frac{1}{2}\boldsymbol{h}_0''(i\Delta t)^2 + \frac{1}{3!}\boldsymbol{h}_0'''(i\Delta t)^3 + \cdots .$$

With this in mind we can apply the results of [5] in Section 4.2.5. Namely, we compute the

SVD[1],

$$\bar{\boldsymbol{H}}^\intercal = \underbrace{\begin{bmatrix} | & | & | & \\ \boldsymbol{e}_0^1 & \boldsymbol{e}_0^2 & \boldsymbol{e}_0^3 & \cdots \\ | & | & | & \end{bmatrix}}_{\boldsymbol{V}} \underbrace{\begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \sigma_3 & \\ & & & \ddots \end{bmatrix}}_{\boldsymbol{\Sigma}} \underbrace{\begin{bmatrix} - & \boldsymbol{p}_1 & - \\ - & \boldsymbol{p}_3 & - \\ - & \boldsymbol{p}_3 & - \\ & \vdots & \end{bmatrix}}_{\boldsymbol{U}^\intercal}.$$

The singular vectors in $\boldsymbol{V}$ correspond to the Frenet-Serret frame (the Gram-Schmidt method applied to the vectors, $\boldsymbol{h}_0', \boldsymbol{h}_0'', \boldsymbol{h}_0'''$),

$$\boldsymbol{e}_0 = \frac{\boldsymbol{h}_0'}{\|\boldsymbol{h}_0'\|}$$
$$\boldsymbol{e}_0^i = \frac{\boldsymbol{h}_0^{(i)} - \sum_{k=1}^{i-1} \langle \boldsymbol{h}_0^{(i)}, \boldsymbol{e}_0^k \rangle \boldsymbol{e}_0^k}{\left\| \boldsymbol{h}_0^{(i)} - \sum_{k=1}^{i-1} \langle \boldsymbol{h}_0^{(i)}, \boldsymbol{e}_0^k \rangle \boldsymbol{e}_0^k \right\|}.$$

The matrix $\boldsymbol{U}$ is similarly defined by the discrete orthogonal polynomials

$$\boldsymbol{p}_1 = \frac{1}{c_1}\boldsymbol{p}$$
$$\boldsymbol{p}_i = \frac{1}{c_i}\left(\boldsymbol{p}^i - \sum_{k=1}^{i-1} \langle \boldsymbol{p}^i, \boldsymbol{p}_k \rangle \boldsymbol{p}_k \right),$$

where $\boldsymbol{p}$ is the vector

$$\boldsymbol{p} = \begin{bmatrix} (-m+1)/2 & (-m+2)/2 & \cdots & 0 & \cdots & (m-2)/2 & (m-1)/2 \end{bmatrix}, \tag{4.10}$$

and where c_i is the normalization constant so that $\langle \boldsymbol{p}_i, \boldsymbol{p}_i \rangle = 1$. Note that $\boldsymbol{p}^i$ means raise $\boldsymbol{p}$ to the power i element-wise. This is similar to the discrete orthogonal polynomials defined in [68], except $\boldsymbol{p}$ is the normalized ones vector $\frac{1}{c_1}[1 \cdots 1]$. These polynomials will be further discussed in Section 4.3.3.

Next we compute the derivatives of $\boldsymbol{e}_i$ for $i = 1, \ldots, r$ and perform DMD, to recover the dynamics with the tridiagonal antisymmetric form seen in (4.6). We note that we have

---

[1] We define the left singular matrix as $\boldsymbol{V}$ and the right singular matrix as $\boldsymbol{U}$. This can be thought of as taking the SVD of the transpose of this matrix. $\boldsymbol{H} - \boldsymbol{h}_0\boldsymbol{1}^\intercal$. This keeps the matrices the definitions of the matrices more inline with the notation used in HAVOK.

measured the Frenet-Serret frame $\boldsymbol{e}_i$ for $i = 1, \dots, r$ at a fixed point of time, and thus cannot immediately employ a finite difference scheme (or an alternative differentiation method). To compute the derivatives $\frac{d\boldsymbol{V}}{dt}$ we once again take advantage of the structure of time delay embeddings. With respect to $\boldsymbol{H}^\intercal$, $\boldsymbol{e}_1, \boldsymbol{e}_2, \dots$ form a instantaneous basis for a trajectory in $\mathbb{R}^n$. From the dual perspective, we can think about this set of vectors as being the coordinates of an r dimensional time series. Using the second interpretation, $\boldsymbol{e}_i$ can each be thought of as a one dimensional time series, in which case $\frac{d\boldsymbol{e}_i}{dt}$ is computed by simply applying a finite difference method to this vector. Once $\boldsymbol{e}_i$ and $\frac{d\boldsymbol{e}_i}{dt}$ have been computed we can compute the corresponding dynamics matrix, which will have precisely the tridiagonal antisymmetric structure in Figure 4.2. The superdiagonal and subdiagonal elements correspond to the curvatures scaled by $\|\boldsymbol{h}_0'\|$.

4.3.2 Synthetic Example

To illustrate the integration procedure, we start with a simple example. We apply the steps of HAVOK and show the resulting modes and dynamics model. We then compute the curvatures of the time delay system and show how they compare to the superdiagonal and subdiagonal of the HAVOK dynamics matrix. We consider the one dimensional time-series governed by

$$x(t) = \sin(t) + \sin(2t),$$

for $t \in [0, 10]$ and with $\Delta t = 0.001$. We follow the steps of HAVOK, forming the time delay matrix $\boldsymbol{H} \in \mathbb{R}^{41 \times 9961}$. We next center the data, subtracting the middle row $\boldsymbol{h}_0$ from all other rows, which forms $\bar{\boldsymbol{H}}$. We next apply the SVD to $\bar{\boldsymbol{H}}^\intercal = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\intercal$. We plot the columns of $\boldsymbol{U} \in \mathbb{R}^{41 \times 4}$ and the columns of $\boldsymbol{V} \in \mathbb{R}^{9961 \times 4}$ in Figure 4.4. The columns of $\boldsymbol{U}$ correspond to the orthogonal polynomials described in Section 4.3.3 and the columns of $\boldsymbol{V}$ are the instantaneous basis vectors $\boldsymbol{e}_i$ for the 9961 dimensional Frenet-Serret frame.

To compute the derivative of the state we now treat $\boldsymbol{V}$ as a 4 dimensional trajectory with

9961 snapshots. Applying DMD to $\boldsymbol{V}$ forms the $\boldsymbol{A}$ matrix,

$$\boldsymbol{A} = \begin{bmatrix} -1.245 \times 10^{-3} & 1.205 \times 10^{-2} & 4.033 \times 10^{-6} & 1.444 \times 10^{-7} \\ -1.224 \times 10^{-2} & 3.529 \times 10^{-4} & 4.458 \times 10^{-3} & 2.283 \times 10^{-6} \\ -9.390 \times 10^{-4} & -3.467 \times 10^{-3} & 5.758 \times 10^{-4} & 6.617 \times 10^{-3} \\ 3.970 \times 10^{-4} & -6.568 \times 10^{-4} & -7.451 \times 10^{-3} & 2.835 \times 10^{-4} \end{bmatrix}. \tag{4.11}$$

This matrix is approximately antisymmetric and tridiagonal as we expect.

Next, we compute the Frenet-Serret frame for the time delay embedding using analytic expressions and show that HAVOK indeed extracts the curvatures of the system multiplied by $\|\boldsymbol{h}_0'\|$. Forming the time delay matrix, we can easily compute $\boldsymbol{h}_0 = [x_{0.02}, x_{0.02+\Delta t} \dots, x_{9.98}]$.

$$\boldsymbol{h}_0 = \Big[\sin(t) + \sin(2t) \text{ for } t \in [0.02, 0.021, \dots, 9.98]\Big]$$

and the corresponding derivatives,

$$\begin{aligned} \dot{\boldsymbol{h}}_0 &= \Big[\cos(t) + 2\cos(2t) \text{ for } t \in [0.02, 0.021, \dots, 9.98]\Big] \\ \ddot{\boldsymbol{h}}_0 &= \Big[-\sin(t) - 4\sin(2t) \text{ for } t \in [0.02, 0.021, \dots, 9.98]\Big] \\ \dddot{\boldsymbol{h}}_0 &= \Big[-\cos(t) - 8\cos(2t) \text{ for } t \in [0.02, 0.021, \dots, 9.98]\Big] \\ \boldsymbol{h}_0^{(4)} &= \Big[\sin(t) + 16\sin(2t) \text{ for } t \in [0.02, 0.021, \dots, 9.98]\Big]. \end{aligned}$$

The 5th derivative $\boldsymbol{h}^{(5)}$ is given by $\cos(t) + 32\cos(2t)$ and can be expressed as a linear combination of the previous derivatives. Namely, $\boldsymbol{h}_0^{(5)} = -5\dddot{\boldsymbol{h}}_0 - 4\dot{\boldsymbol{h}}_0$. This can also be shown using the fact that $x(t)$ satisfies the 4th order ordinary differential equation $x^{(4)} + 5\ddot{x} + 4x = 0$.

Since only the first four derivatives are linearly independent we know that only the first three curvatures will be nonzero. We can compute the exact values of the first two curvatures using the formulas [80],

$$\kappa_1 = \frac{\sqrt{\det(\begin{bmatrix}\dot{\boldsymbol{h}}_0 & \ddot{\boldsymbol{h}}_0\end{bmatrix}^\intercal \begin{bmatrix}\dot{\boldsymbol{h}}_0 & \ddot{\boldsymbol{h}}_0\end{bmatrix})}}{\left\|\dot{\boldsymbol{h}}_0\right\|^{3/2}} \text{ and } \kappa_2 = \frac{\sqrt{\det(\begin{bmatrix}\dot{\boldsymbol{h}}_0 & \ddot{\boldsymbol{h}}_0 & \dddot{\boldsymbol{h}}_0\end{bmatrix}^\intercal \begin{bmatrix}\dot{\boldsymbol{h}}_0 & \ddot{\boldsymbol{h}}_0 & \dddot{\boldsymbol{h}}_0\end{bmatrix})}}{\det(\begin{bmatrix}\dot{\boldsymbol{h}}_0 & \ddot{\boldsymbol{h}}_0\end{bmatrix}^\intercal \begin{bmatrix}\dot{\boldsymbol{h}}_0 & \ddot{\boldsymbol{h}}_0\end{bmatrix})}.$$

These formulas yields the values $\kappa_1 = 0.01205$ and $\kappa_2 = 0.00446$. As we expect these values

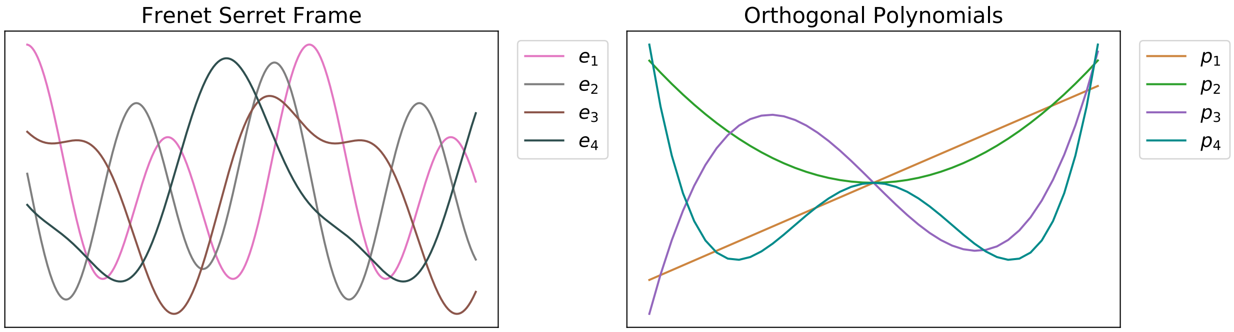


Figure 4.4: Frenet-Serret frame (left) and corresponding orthogonal polynomials (right) for HAVOK applied to time-series generated by $x(t) = \sin(t) + \sin(2t)$.

are very close to those measured with HAVOK.[2] In Section 4.5, we propose a modification to HAVOK which will provide an even better approximation to the Frenet-Serret frame.

4.3.3 Orthogonal Polynomials and Centering

In the HAVOK method, we apply the SVD to the centered matrix $\bar{\boldsymbol{H}}$. The columns of the matrix $\boldsymbol{U}$ in this decomposition yield a set of orthonormal polynomials, which are defined by (4.9). In the continuous case the inner product in (4.9) is $\langle a(t), b(t)\rangle = \int_{-p}^{p} a(t)b(t)dt$, while in the discrete case $\langle a, b\rangle = \sum_{j=-p}^{p} a_j b_j$. Here we list the first five of these polynomials $p_i(x)$ in the continuous case. The first five polynomials in the discrete case may be found in

---

[2] In particular, the superdiagonal of the matrix appears to be a very good approximation to the curvatures. Understanding why the superdiagonal is so close in value to the true curvatures is an active area of research.

Appendix C.1.

$$
\begin{aligned}
p_1(x) &= \frac{x}{c_1(p)} \text{ where } c_1(p) = \frac{\sqrt{6}\sqrt{p^3}}{3}\\
p_2(x) &= \frac{x^2}{c_2(p)} \text{ where } c_2(p) = \frac{\sqrt{10}\sqrt{p^5}}{5}\\
p_3(x) &= \frac{1}{c_3(p)}\left(x^3 - \frac{3}{5}p^2x\right) \text{ where } c_3(p) = \frac{2\sqrt{14}\sqrt{p^7}}{35}\\
p_4(x) &= \frac{1}{c_4(p)}\left(x^4 - \frac{5}{7}p^2x^2\right) \text{ where } c_4(p) = \frac{2\sqrt{2}\sqrt{p^9}}{21}\\
p_5(x) &= \frac{1}{c_5(p)}\left(x^5 + \frac{5}{21}p^4x - \frac{10}{9}p^2x^3\right) \text{ where } c_5(p) = \frac{8\sqrt{22}\sqrt{p^{11}}}{693}.
\end{aligned}
$$

By construction $p_i(t)$ for $i = 1, 2, \ldots$ form a set of orthonormal polynomials, where $p_i(t)$ has degree i. The Legendre polynomials $\boldsymbol{l}_i$ are a similar set of orthogonal polynomials [2, 226]. They are defined by the recursive relation

$$
\begin{aligned}
\boldsymbol{l}_1 &= \frac{1}{c_1}\begin{bmatrix}1 & 1 & \cdots & 1 & \cdots & 1 & 1\end{bmatrix}\\
\boldsymbol{l}_i &= \frac{1}{p_i}\left(\boldsymbol{p}^i - \sum_{k=1}^{i-1}\langle \boldsymbol{p}^i, \boldsymbol{l}_k\rangle\right),
\end{aligned}
$$

where $\boldsymbol{p}$ is defined in (4.10). For the corresponding Legendre polynomials normalized over $[-p, p]$ we refer the reader to [68]. The primary difference, between these two sets of polynomials is that the first polynomial $\boldsymbol{p}_1$ is linear, while the first Legendre polynomial is constant (corresponding in the discrete case to the normalized ones vector). In particular, if $\boldsymbol{H}$ is not centered before the SVD applied, the resulting columns of $\boldsymbol{U}$ will be the Legendre polynomials. However, the resulting $\boldsymbol{V}$ will no longer be the Frenet-Serret frame. The resulting frame in this case would correspond to applying the Gram-Schmidt method to the set $\{\boldsymbol{\gamma}(t), , \boldsymbol{\gamma}'(t), \boldsymbol{\gamma}''(t), ...\}$ instead of $\{\boldsymbol{\gamma}'(t), \boldsymbol{\gamma}''(t), \boldsymbol{\gamma}'''(t)...\}$. Recently it has been shown that using centering as a preprocessing step is beneficial for the dynamic mode decomposition [88]. That being said, since the derivation of the tridagonal and antisymmetric structure seen in the Frenet-Serret frame is based on the properties of the derivatives and orthogonality this same structure can be computed without the centering step.

4.4 Limits and Requirements

From Section 4.3.1, we showed that HAVOK will yield a good approximation to the Frenet-Serret in the limit that the time interval spanned by each row $\boldsymbol{H}$ goes to zero. To be more precise, HAVOK will yield the Frenet-Serret frame if (4.8) is satisfied. However, this can be difficult to check in practice. Here we establish approximate rules which can be followed:

1. *Choose* Δt *to be small.* The specific constraint we have from (4.8) is

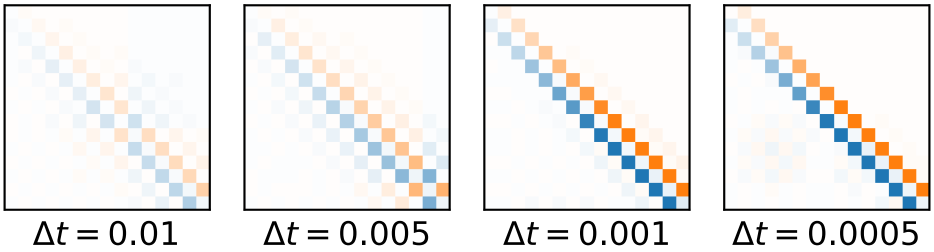
   $$\|\boldsymbol{h}_0' t_i\| \gg \left\|\frac{\boldsymbol{h}_0''}{2} t_i^2\right\| \gg \left\|\frac{\boldsymbol{h}_0'''}{6} t_i^3\right\| \gg \cdots \gg \left\|\frac{\boldsymbol{h}_0^{(k)}}{k!} t_i^k\right\|,$$

   for $-m\Delta t/2 \leq t_i \leq m\Delta t/2$ or more simply $|t_i| \leq m\Delta t$, where Δt is the sampling period of the data and m is the number of delays in the Hankel matrix $\boldsymbol{H}$. If we assume that $m\Delta t < 1$, then rearranging,

   $$m\Delta t \ll \frac{2\,\|\boldsymbol{h}_0'\|}{\|\boldsymbol{h}''(0)\|}, \frac{3\,\|\boldsymbol{h}_0''\|}{\|\boldsymbol{h}_0'''\|}, \ldots, \frac{k\left\|\boldsymbol{h}_0^{(k-1)}\right\|}{\left\|\boldsymbol{h}_0^{(k)}\right\|}. \tag{4.12}$$

   In practice, the series of ratios of derivatives defined in (4.12) grows, in which case it is only necessary to check the first inequality. By choosing the sampling period of the data to be small we can constrain the data to satisfy this inequality. To illustrate this we show in Figure 4.5 the resulting dynamics matrix $\boldsymbol{A}$ from HAVOK for several different values of Δt. As we expect, as we decrease the sampling period the dynamics becomes antisymmetric and tridiagonal.

2. *Choose the number of columns* n *to be large.* The number of columns comes into the Taylor expansion through the derivatives $\left\|\boldsymbol{h}_0^{(k)}\right\|$, since $\boldsymbol{h}_0^{(k)} \in \mathbb{R}^n$. In general, the number of necessary columns is system dependent. For the synthetic example $x(t) = \sin(t) + 2\sin(t)$ we can show that the ratio $2\,\|\boldsymbol{h}_0'\| \,/\, \|\boldsymbol{h}_0''\|$ saturate to a fixed value in the limit as n goes to infinity (see Appendix C.2). However, for short time series (small values of n) this ratio can be arbitrarily small, and hence (4.12) will be difficult to satisfy.


Increasing Sampling Frequency

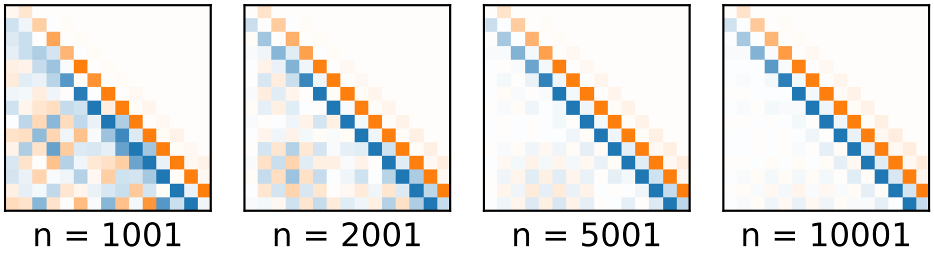

Increasing Number of Data Columns

Figure 4.5: Effects of sampling frequency and number of columns on HAVOK model for the Lorenz system. Given the Hankel matrix $\boldsymbol{H}$, the linear dynamical model is plotted for values of sampling period Δt (top) equal to $0.01, 0.005, 0.001, 0.0005$ for a fixed number of rows and fixed time span of measurement (top). Similarly, the model is plotted for values of number of columns n equal to $1001, 2001, 5001$, and 10001 for fixed sampling frequency and time span of measurement $q\Delta t$(bottom). As we increase the sampling frequency and the number of columns the data becomes more antisymmetric with nonzero elements only on the super- and sub-diagonals. This is in line with the results in Section 4.4.

In general, due to practical constraints and restrictions, it may be difficult to guarantee that given data satisfies these two requirements. In Section 4.4.1 and Section 4.5 we propose methods to remedy this.

4.4.1 Interpolation

From the first requirement, we see that the sampling frequency Δt needs to be sufficiently small to recover the antisymmetric structure. However, this assumption may not always be satisfied in practice due to, for example, measurement constraints. One solution to remedy this is to use interpolation. To be more precise, we can fit the data to splines. We then construct a new dataset by evaluating these models at a finer Δt that satisfies (4.12). We note that the ratio of the derivatives $\|\boldsymbol{h}_0'\| \,/\, \|\boldsymbol{h}_0''\|\,, \|\boldsymbol{h}_0''\| \,/\, \|\boldsymbol{h}_0'''\|\,, \ldots$ may also contain some dependence on Δt, but is not significantly affected in practice.

As an example, we consider a set of time series measurements time generated from the Lorenz system (see 4.5 for more details about this system). We choose a sampling period of $\Delta t = 0.1$. This time series is plotted in Figure 4.6. Applying HAVOK with centering with $m = 201$, we plot the $\boldsymbol{A}$ matrix and the first three columns of $\boldsymbol{U}$. As expected, $\boldsymbol{A}$ is not antisymmetric and $\boldsymbol{U}$ are not the orthogonal polynomials like from the synthetic example in Figure 4.4.

Next, we applied cubic spline interpolation, evaluating at a sampling rate of $\Delta t = 0.001$. This interpolated data is in gray. Applying HAVOK to this interpolated data yields a new $\boldsymbol{A}$ matrix. As predicted, this matrix has a antisymmetric structure and the $\boldsymbol{U}$ corresponds to the orthogonal polynomials in Section 4.3.3.

4.5 Structured HAVOK (sHAVOK)

We see that in general, under these assumptions about the number of delays and sampling size and with the incorporation of centering, the HAVOK model is a method for approximately computing the Frenet-Serret frame. Here we propose a modification of HAVOK to induce this antisymmetric structure for a smaller number of delays n. We focus on the step in HAVOK

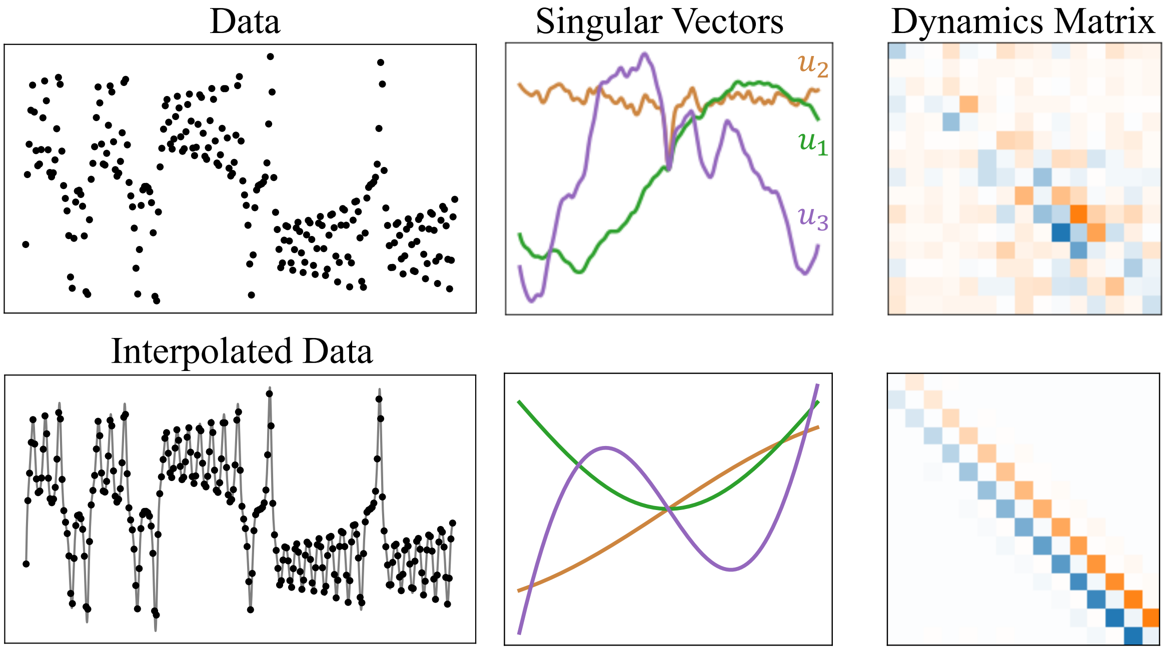


Figure 4.6: In the case where a dynamical system is sparsely sampled, interpolation can be used to recover a more tridiagonal and antisymmetric matrix for the linear model in HAVOK. First, we simulate the Lorenz system, measuring $x(t)$ with sampling period of $\Delta t = 0.1$. The resulting dynamics model $\boldsymbol{A}$ and corresponding singular vectors of $\boldsymbol{U}$ are plotted. Due to the low sampling frequency these values do not satisfy the requirements in (4.12). Consequently the dynamics matrix is not antisymmetric and the singular vectors do not corresponding to the orthogonal polynomials in Section 4.3.3. Next, the data is interpolated using cubic splines, and subsequently sampled using a sampling period of $\Delta t = 0.001$. In this case the data satisfies the assumptions in (4.12), which yields the tridiagonal antisymmetric structure for $\boldsymbol{A}$ and orthogonal polynomials for $\boldsymbol{U}$ as predicted.

of splitting $\boldsymbol{V}$ into $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$. In the Frenet-Serret framework we measure the evolution of the orthonormal frame $\boldsymbol{e}_1(t), \boldsymbol{e}_2(t), \ldots, \boldsymbol{e}_r(t)$. Comparing this to HAVOK, $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ should correspond to instances of this frame. Although $\boldsymbol{V}$ is a unitary matrix in general, $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ which consist of removing a column from $\boldsymbol{V}$ are not. To enforce this orthogonality, we propose to split $\bar{\boldsymbol{H}}$ into two time-shifted matrices $\bar{\boldsymbol{H}}_1$ and $\bar{\boldsymbol{H}}_2$ and compute two SVDs with rank truncation r,

$$\bar{\boldsymbol{H}}_1 = \boldsymbol{U}_1 \boldsymbol{\Sigma}_1 \boldsymbol{V}_1^\intercal \text{ and } \bar{\boldsymbol{H}}_2 = \boldsymbol{U}_2 \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^\intercal.$$

By construction, $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ are now orthogonal matrices and thus

$$\hat{\boldsymbol{A}} = \boldsymbol{V}_2^\intercal \boldsymbol{V}_1.$$

or in the continuous case $\boldsymbol{A} = \left(\boldsymbol{V}_2^\intercal \boldsymbol{V}_1 - \boldsymbol{I}\right)/\Delta t$. If this system is not closed (nonzero forcing term), then $\boldsymbol{V}_2$ is defined as columns 2 to $n-1$ of the SVD singular vectors with an $r-1$ rank truncation $\boldsymbol{V}_{r-1}^\intercal$. $\hat{\boldsymbol{A}} \in \mathbb{R}^{r-1 \times r-1}$ and $\hat{\boldsymbol{B}} \in \mathbb{R}^{r-1 \times 1}$ are computed as $\left[\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}\right] = \boldsymbol{V}_2^\intercal \boldsymbol{V}_1$. Corresponding pseudocode may be found in Appendix C.3.

This simple adjustment to HAVOK yields a new method shown in Figure 4.7, which we denote as *structured HAVOK* (sHAVOK). As a simple example, we first apply sHAVOK with centering to the synthetic example introduce in Section 4.3.2 generated by $x(t) = \sin(t) + \sin(2t)$. The resulting dynamics matrix is

$$\boldsymbol{A} = \begin{bmatrix} -1.116 \times 10^{-5} & 1.204 \times 10^{-2} & -1.227 \times 10^{-5} & 8.728 \times 10^{-8} \\ -1.204 \times 10^{-2} & -1.269 \times 10^{-5} & 4.458 \times 10^{-3} & 4.650 \times 10^{-6} \\ 2.053 \times 10^{-5} & -4.458 \times 10^{-3} & -4.897 \times 10^{-6} & 6.617 \times 10^{-3} \\ -9.956 \times 10^{-8} & -1.118 \times 10^{-7} & -6.617 \times 10^{-3} & -3.368 \times 10^{-6} \end{bmatrix}$$

We see immediately that with this small modification, in comparison to (4.11), the estimates of the curvatures both below and above the diagonal are now equal and the rest of the elements in the matrix, which should be zero, are almost all smaller by an order of magnitude. In addition, the curvatures are equal to the true values to three decimal places.

Although these two methods converge in the limit of infinite data, the difference is most prominent in the case of shorter time series, in which case we may not have measurements over a large period of time. We demonstrate these results on three nonlinear systems, the Lorenz attractor, the Rössler attractor, and a double pendulum. For each of these systems we apply both the HAVOK and sHAVOK methods to this data and compute the corresponding

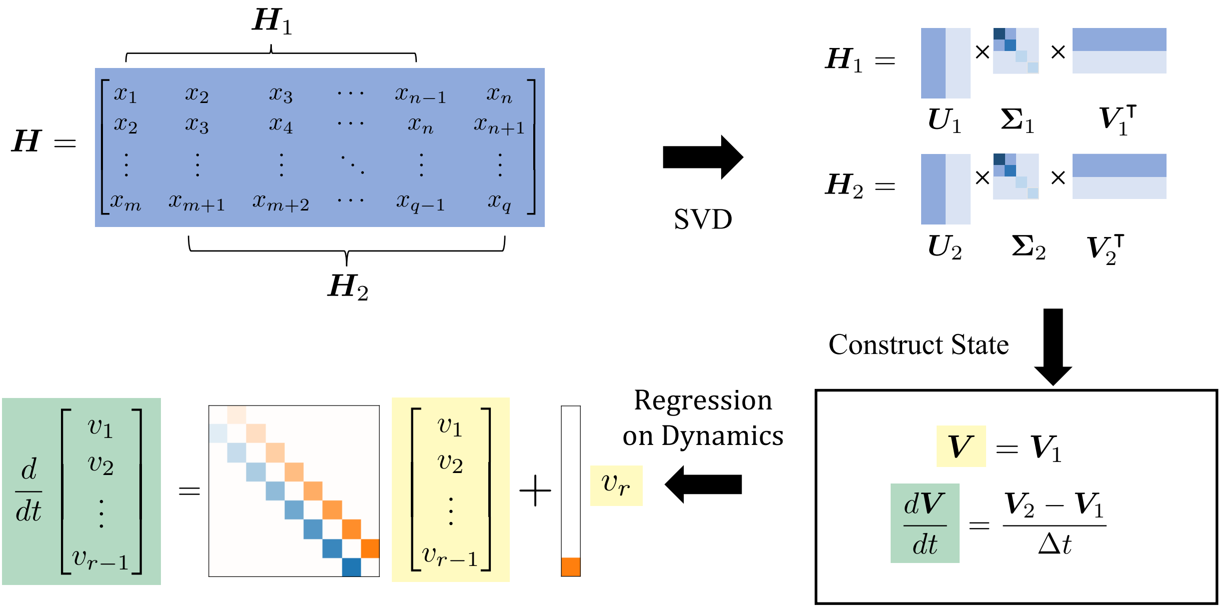


Figure 4.7: Outline of steps in structured HAVOK (sHAVOK) method. First, given a dynamical system a single variable $x(t)$ is measured. Time-shifted copies of $x(t)$ are stacked to form a Hankel matrix $\boldsymbol{H}$. $\boldsymbol{H}$ is split into two time-shifted matrices, $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$. The singular value decomposition (SVD) is applied to these two matrices individually. This results in reduced order representations, $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$, respectively. The matrices, $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ are then used to construct an approximation to this low dimensional state and its derivative. Finally, linear regression is performed on these two matrices to form a linear dynamical model with an additional forcing term in the last component.

dynamics matrices $\boldsymbol{A}$. In all of these cases the sHAVOK dynamics matrix is much more antisymmetric and tridiagonal than the corresponding HAVOK matrix. This can be attributed to the fact that sHAVOK better approximates the Frenet-Serret frame than HAVOK.

In addition to the dynamics matrices, we plot the eigenvalues of $\boldsymbol{A}$, $\omega_k \in \mathbb{C}$ for $k = 1, \ldots r$ for HAVOK (teal) and sHAVOK (maroon). We additionally plot the eigenvalues (black crosses) corresponding to the data measured in the large data limit, but at the same sampling frequency. In this large data limit both sHAVOK and HAVOK yield the same antisymmetric tridiagonal dynamics matrix and corresponding eigenvalues. Comparing the eigenvalues we immediately see that eigenvalues from sHAVOK are much closer in value to those computed in the large data limit than those of HAVOK. Thus, we see that even with a short trajectory we can still recover key information about the corresponding dynamics. Below, we describe each of the systems and their configurations.

Lorenz Attractor: We first illustrate these two methods on the Lorenz system. Originally developed in the fluids community, the Lorenz (1963) system is governed by three first order differential equations [133]:

$$\begin{aligned}\dot{x} &= \sigma(y - x)\\ \dot{y} &= x(\rho - z) - y\\ \dot{z} &= xy - \beta z.\end{aligned}$$

The Lorenz system has since been used to model systems in a wide variety of fields, including chemistry [161], optics [224], and circuits [85]. We simulate $3,000$ samples with initial condition $[-8, 8, 27]$ and a stepsize of $\Delta t = 0.001$, and measure the variable $x(t)$. We use the common parameters $\sigma = 10, \rho = 28$, and $\beta = 8/3$. This trajectory is shown in Figure 4.8, and corresponds to a few oscillations about a fixed point. We apply both the HAVOK and sHAVOK methods to this data and compute the corresponding dynamics matrices $\boldsymbol{A}$. Clearly, the sHAVOK dynamics matrix is much more antisymmetric than the corresponding HAVOK matrix. We compare the spectra to that of a longer trajectory containing $300,000$ samples.

Rössler Attractor: The Rössler attractor is given by the following nonlinear differential equations [172, 171]:

$$\begin{aligned}\dot{x} &= -y - z\\ \dot{y} &= x + ay\\ \dot{z} &= b + z(x - c)\end{aligned}$$

We choose to measure the variable $x(t)$. This attractor is a canonical example of chaos, like the Lorenz attractor. Here we perform a simulation with $70,000$ samples and a stepsize of $\Delta t = 0.001$. We choose the following common values of $a = 0.1$, $b = 0.1$ and $c = 14$ and the initial condition $x_0 = y_0 = z_0 = 1$. We similarly plot the trajectory and dynamics matrices. We compare the spectra in this case to a longer trajectory using a simulation for $300,000$ samples.

Double Pendulum: The double pendulum is a similar nonlinear differential equation, which models the motion of a pendulum which is connected at the end to another pendulum [193]. This system is typically represented by its Lagrangian, $\mathcal{L}(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$

$$\mathcal{L} = \frac{1}{6}ml^2\left(\dot{\theta}_2^2 + 4\dot{\theta}_1^2 + 3\dot{\theta}_1\dot{\theta}_2\cos\left(\theta_1 - \theta_2\right)\right) + \frac{1}{2}mgl\left(3\cos\theta_1 + \cos\theta_2\right), \tag{4.13}$$

where θ_1 and θ_2 are the angles between the top and bottom pendula and the vertical axis, respectively. m is the mass at the end of each pendulum, l is the length of each pendulum and g is the acceleration constant due to gravity. Using the Euler-Lagrange equations,

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} - \frac{\partial \mathcal{L}}{\partial \theta_i} = 0 \text{ for } i = 1, 2,$$

we can construct two second order differential equations of motion. The trajectory is computed using a variational integrator to integrate the equations to approximate

$$\delta \int_a^b \mathcal{L}(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)dt = 0.$$

We simulate this system with stepsize of $\Delta t = 0.001$ and for 1200 samples. We choose $m_1 = m_2 = l_1 = l_2 = 1$ and $g = 10$, and use initial conditions $\theta_1 = \theta_2 = \pi/2$, $\dot{\theta}_1 = -0.01$ and $\dot{\theta}_2 = -0.005$. As our measurement for HAVOK and sHAVOK we use $x(t) = \sin(\theta_1(t))$, and compare our data to a trajectory containing $100,000$ samples.

4.5.1 Real Data

Here we apply sHAVOK to two real datasets, the trajectory of a double pendulum and measles outbreak data. Similar to the synthetic examples, we find that the the dynamics matrix from sHAVOK is much more antisymmetric and tridiagonal compared to the dynamics matrix for HAVOK. In both cases, some of the HAVOK eigenvalues contain positive real components. However, for sHAVOK they do not, resulting in much more accurate and stable reconstructions.

Double Pendulum: We first look at measurements of a double pendulum. [116]. A picture of the setup can be found in Figure 4.9. The Lagrangian in this case is very similar to that in (4.13). One key difference in the synthetic case is that all of the mass is contained at the joints, while for in this experiment, the mass is spread over each arm. To accommodate this, the Lagrangian can be slightly modified

$$\mathcal{L} = \frac{1}{2}\left(m_1(\dot{x}_1^2 + \dot{y}_1^2) + m_2(\dot{x}_2^2 + \dot{y}_2^2)\right) + \frac{1}{2}\left(I_1\dot{\theta}_1^2 + I_2\dot{\theta}_2^2\right) - (m_1 y_1 + m_2 y_2)\, g$$

where $x_1 = a_1\sin(\theta_1)$, $x_2 = l_1\sin(\theta_1) + a_2\sin(\theta_2)$, $y_1 = a_1\cos(\theta_1)$, and $y_2 = l_1\cos(\theta_1) + a_2\cos(\theta_2)$. m_1, and m_2 are the masses of the pendula, l_1 and l_2 are the lengths of the pendula, a_1 and a_2 are the distances from the joints to the center of masses of each arm, and I_1 and I_2 are the moments of inertia for each arm. When $m_1 = m_2 = m$, $a_1 = a_2 = l_1 = l_2$, and $I_1 = I_2 = ml^2$ we recover (4.13). We sample the data at $\Delta t = 0.001$s. We plot $\sin(\theta_2(t))$ over a 15s time interval. The data over this interval appears approximately periodic.

Measles Outbreaks: Last, as a final example we apply measles outbreak data. Measles data has been shown to exhibit chaotic behavior [179, 198]. Brunton et. al applied HAVOK to measles data and successfully showed that the method could extract transient behavior [30]. Here, we apply sHAVOK to data corresponding to the number of measles cases in New York City between 1928 to 1964 [132].

For both systems, we first apply sHAVOK and interpolation to a subset of the data corresponding to shown in Figure 4.9. We then compare that to HAVOK applied over the same interval. We use, $m = 101$ delays with a $r = 5$ rank truncation for the double pendulum, and $m = 51$ delays and a $r = 6$ rank truncation for the measles data. Like in previous examples, the sHAVOK dynamics is tridiagonal and antisymmetric while the HAVOK dynamics matrix is not. Next, we plot the corresponding spectra for these two methods, in addition to the eigenvalues applied to HAVOK over the entire time series. Most noticeably, the eigenvalues from sHAVOK are closer to the long data limit values. In addition, two of the HAVOK eigenvalues lie to the right of the real axis, and thus have positive real components. All of the sHAVOK eigenvalues, on the other hand, have negative real components. This difference is most prominent in the reconstructions of the first singular vector. In particular, since two of the eigenvalues from HAVOK are positive, the reconstructed time series grows exponentially. In contrast, for sHAVOK the corresponding time-series remains bounded providing a much better reconstruction to the true data.

4.6 Discussion

In this chapter, we connect results from time delay embeddings, dimensionality reduction techniques, and differential geometry to decompose sparse linear dynamical model plus a forcing term. Furthermore this linear model has a very specific structure. Namely, it is antisymmetric with nonzero elements only along the sub- and super-diagonals. This structure was previously observed in the Lorenz system for the HAVOK model. We show that if certain requirements about our measurements are satisfied, this structure can be observed in other systems. In the case, where these requirements are not satisfied, we provide methods to overcome this barrier. In particular, interpolation may be used as a preprocessing step if the sampling frequency of the data is low. In the case where the length of data is limited we provide a modification to the HAVOK method, sHAVOK, which is closer connected to theory. We demonstrate sHAVOK on three chaotic examples, and two real world examples from measurements of a double pendulum and measles outbreaks. Using this modification, we observed improved stability in the model and better reconstructions of the dynamics.

Connecting these three fields opens up a wide variety of applications and future work. Namely, by understanding this new perspective, we now better understand the requirements and limitations of HAVOK and have proposed simple modifications to the method which improve its performance on data. However, the full implications of this theory remain unknown. Going forward, we encourage the exploration of these connections. Differential geometry, dimensionality reduction and time delay embeddings are all well-established fields, and by understanding these connections we can develop more robust methods for modeling time series.

More concretely, in this thesis by connecting HAVOK to the Frenet-Serret frame, we recognized the importance of enforcing orthogonality for $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, and consequently developed sHAVOK. However, with this theory we can incorporate further improvements on the method. For example, sHAVOK can be thought of as a first order forward difference method, approximating the derivative and state by $\left(\boldsymbol{V}_2 - \boldsymbol{V}_1\right)/\Delta t$ and $\boldsymbol{V}_1$, respectively. By employing a central difference scheme, such as approximating the state by $\boldsymbol{V}$, we have observed this to further enforce the antisymmetry in the dynamics matrix and move the corresponding eigenvalues towards the imaginary axis.

Throughout this analysis, we have focused purely on linear methods. In recent years nonlinear methods for dimensionality reduction, such as autoencoders and diffusion maps, have gained popularity [44, 146]. Like these methods, we would like nonlinear methods to achieve similar goals, of sparsity and interpretibility. By understanding these structures in the linear domain, we hope to generalize these methods to create more accurate and robust methods which can accurately model a greater class of functions.

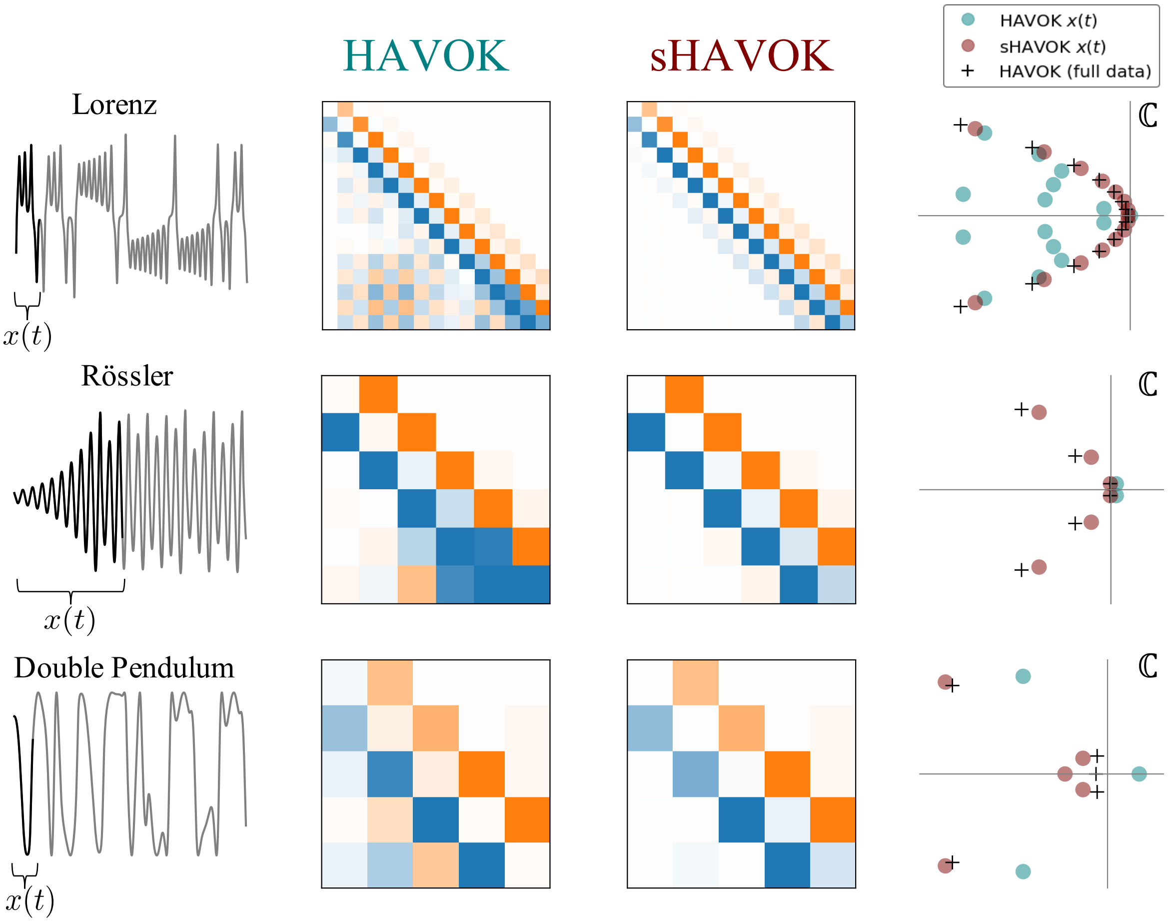


Figure 4.8: Comparison of HAVOK and structured HAVOK (sHAVOK) for three example systems. For each system, we simulated a trajectory extracting a single coordinate (gray). We then apply HAVOK and sHAVOK to a subset of this trajectory shown in black. The resulting linear dynamical models are plotted. The resulting model for sHAVOK yields an antisymmetric structure with nonzero elements only along the subdiagonal and superdiagonal. The corresponding eigenvalue spectra for HAVOK and sHAVOK are additionally plotted in teal and maroon, respectively. Additionally, we plot the eigenvalues from HAVOK for the full trajectory. In all cases, the eigenvalues of sHAVOK are much closer in value to those in the long trajectory limit than HAVOK. Thus, sHAVOK can inform us about long term dynamics, even with a much shorter trajectory.

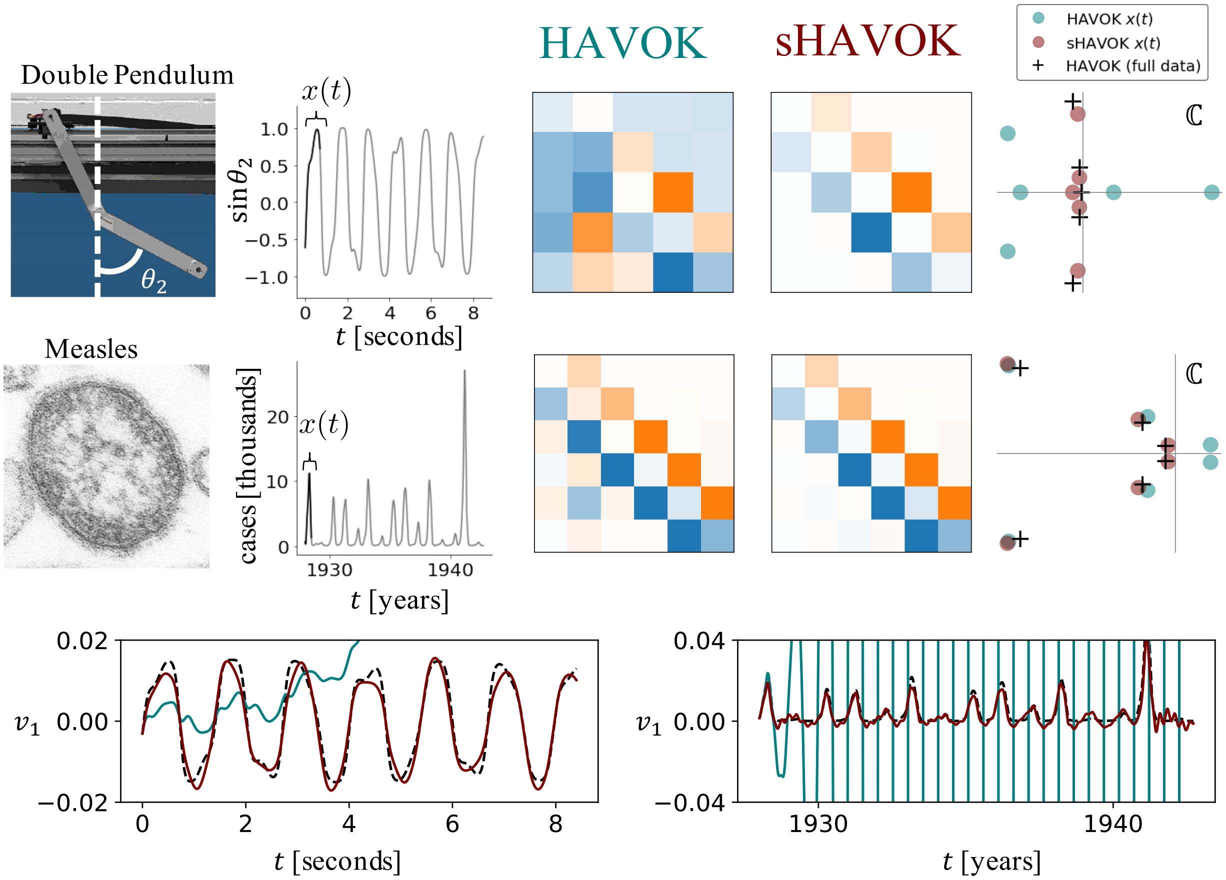


Figure 4.9: Comparison of HAVOK and structured HAVOK (sHAVOK) for two real world systems: a double pendulum and measles outbreak data. For each system, we measure a trajectory extracting a single coordinate (gray). We then apply HAVOK and sHAVOK to a subset of this trajectory shown in black. The resulting linear dynamical models are plotted. The resulting model for sHAVOK yields an antisymmetric structure with nonzero elements only along the subdiagonal and superdiagonal. The corresponding eigenvalue spectra for HAVOK and sHAVOK are additionally plotted in teal and maroon, respectively. Additionally, we plot the eigenvalues from HAVOK for a long trajectory. In both cases, the eigenvalues of sHAVOK are much closer in value to those in the long trajectory limit than HAVOK. In addition, some of the eigenvalues of HAVOK contain positive real components. The corresponding reconstructions of the first singular vector of the corresponding Hankel matrices, in addition to the true values are plotted. Note that that the HAVOK reconstruction exponentially grows due to the unstable eigenvalues, while the sHAVOK reconstruction does not.

Chapter 5

SPARSE SYSTEM IDENTIFICATION WITH UNCERTAINTY QUANTIFICATION

5.1 Introduction

In recent years there has been a rapid increase in measurements gathered from complex nonlinear dynamics for which their governing equations are unknown. A key challenge is to discover explicitly representations of these equations, which can then be used for system identification, forecasting and control. Measurements are often contaminated by random noise or may exhibit chaotic behavior, in which case it is critical to quantify how uncertainty affects the model discovery process. In order to address this challenge, we introduce the uncertainty quantification sparse identification of nonlinear dynamics (UQ-SINDy) framework, which leverages sparsity promotion in a Bayesian probabilistic setting to extract a parsimonious set of governing equations. Our method provides uncertainty estimates of both the parameter values and the inclusion probabilities for different terms in the models.

Discovery of governing equations plays a fundamental role in the development of physical theories. In more recent years, with increasing computing power and data availability, there have been substantial efforts to identify the governing equations directly from data [26, 184, 232]. There has been particular emphasis on parsimonious representations as they have the benefit of promoting interpretibility and generalize well to unknown data [16, 32, 31, 136, 152, 162, 210, 221]. In [31], Brunton et. al proposed the SINDy method, which leverages dictionary learning and sparse regression to model systems. This approach has been successful in modeling a wide array of fields, systems, including chemistry [90], optics [195], epidemiology [91], and plasma physics [47]. Furthermore, there have been a variety of modifications, including improved robustness to noise [42, 115], generalizations to partial differential equations [165, 175, 174], boundary conditions [190], and libraries of rational functions [140, 117].

Although these methods identify the equations, measurements often contain observation error and are governed by complex or chaotic behaviors, making it difficult to determine how these models will generalize to unknown datasets. A common approach to remedy this is to use the Bayesian probability framework, in which uncertainty is quantified in terms of probability, and in which priors are employed to encode assumptions and prior knowledge about model parameters [63, 225]. Bayesian methods have been powerful tools for uncertainty quantification in time series models, with applications to weather forecasting [3, 57, 233], disease modeling [21, 127, 234], traffic flow [40, 199, 237], and finance [65, 208, 229], among many others. More recently, these methods have been incorporated into model discovery frameworks, showing state of the art performance for system identification in the presence of noise [61, 148, 232]. Although these methods provide a range of possible values, realizations of these models are in general not sparse and consequently lack the capability to identify relevant terms in the model.

Sparse regression is a popular tool to identify a small subset of variables which explain the data. However, finding the true minimum is computationally intractable in practice. In the frequentist setting, a popular solution is to use the Lasso, which corresponds to an l_1 penalty term [207]. In the Bayesian setting, sparsity is generated by fundamentally different mechanisms. Most notably, although the corresponding prior (the Laplace prior) shares the same maximum likelihood estimator as the Lasso [154], the distribution has fat tails and thus does not produce sparse realizations [41]. The spike and slab model remedies this by explicitly using Bernoulli variables to determine whether a term is present in the model, and has become the "gold standard" for sparsity in the Bayesian framework [144, 106, 138]. One disadvantage to this prior however is its dependence on discrete variables. This makes the inference prohibitively expensive for high dimensional systems. Smooth approximations, such as the horseshoe [38, 39], Horseshoe+ prior [22], Dirichlet-Laplace prior [24], and R2-D2 prior [236], have been shown to yield comparable to performance. For this work we will primarily focus on the regularized horseshoe prior, also known as the Finnish horseshoe [159].

In this work, we propose a robust uncertainty quantification SINDy (UQ-SINDy) framework which provides uncertainty estimates of both the parameter value and inclusion probabilities and promotes sparsity in realizations of the model. This model leverages advances in Bayesian approaches to solving ODEs and sparsity to achieve this goal. In sections 5.2.1 and 5.2.2 we review the SINDy method and Bayesian inference for ordinary differential equations. In section 5.2.3 we review sparsity promoting priors, namely the spike and slab and regularized horseshoe priors, comparing their performance to the Laplace prior. In section 5.3.1, we introduce two sparsity promoting Bayesian methods, spike and slab SINDy and regularized horseshoe. In sections 5.3.2 and 5.3.3, we illustrate these methods on two synthetic nonlinear models, a Lotka Volterra model and nonlinear oscillator, and real world lynx hare population data. We find that these methods are able to extract accurate and meaningful Bayesian models even in the presence of significant noise and sparse samples. These results are summarized future improvements are discussed in section 3.5.

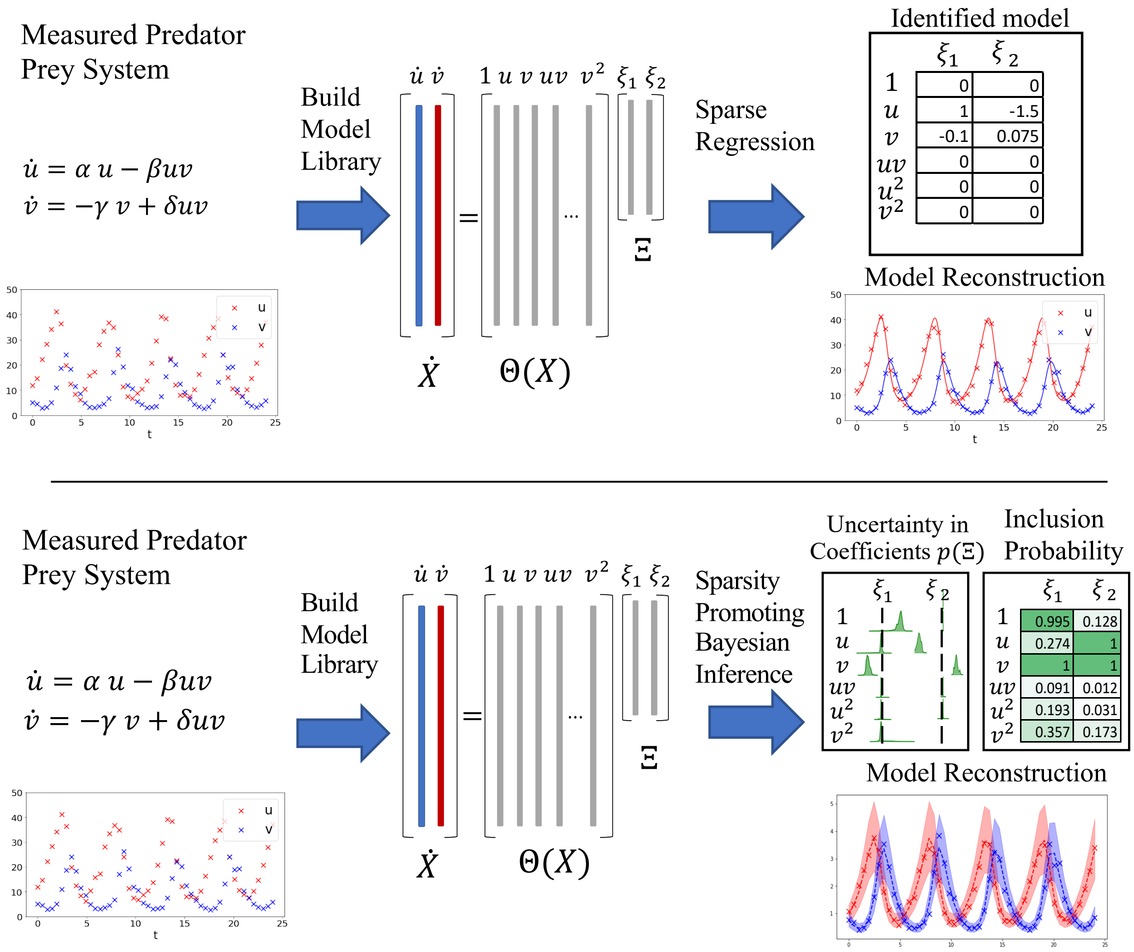
5.2 Background

5.2.1 SINDy

The sparse identification of nonlinear dynamics (SINDy) method is a recently developed technique which leverages compressed sensing and machine learning techniques to identify the governing equations from a given time series (Figure 5.1). In particular, we start with a time series $\boldsymbol{x}(t) = [x_1(t), x_2(t), \ldots x_d(t)]^\intercal \in \mathbb{R}^d$ for $t = t_1, \ldots, t_n$, which is governed by the differential equation,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}),$$

for some unknown function $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$. The goal of SINDy is to discover $\boldsymbol{f}$.



Identified model

	ξ_1	ξ_2
1	0	0
u	1	-1.5
v	-0.1	0.075
uv	0	0
u^2	0	0
v^2	0	0

Inclusion Probability

	ξ_1	ξ_2
1	0.995	0.128
u	0.274	1
v	1	1
uv	0.091	0.012
u^2	0.193	0.031
v^2	0.357	0.173

Figure 5.1: Comparison of SINDy algorithm and UQ-SINDy. Top: Schematic of SINDy algorithm. A dynamical system governed by unknown governing equations is measured. Next, we computed the derivative of the time series $\dot{\boldsymbol{X}}$ and construct a library $\boldsymbol{\Theta}(\boldsymbol{X})$ of candidate terms. Last, we perform sparse regression to identify the terms in the model which can to reconstruct the time series. Bottom: Schematic of UQ-SINDy algorithm. A dynamical system governed by unknown governing equations is measured. Next, we compute a library $\boldsymbol{\Theta}(\boldsymbol{X})$ of candidate terms. Last, we perform sparsity promoting Bayesian inference to compute inclusion probability of each term in the model and probability distribution for each term in the model. An ensemble of reconstructions can then be compute, which provides uncertainty quantification in the reconstruction.

To do so, we first define $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]^\intercal \in \mathbb{R}^{n \times d}$ as the collection of snapshots and construct a library of possible terms in our model

$$\boldsymbol{\Theta}(\boldsymbol{x}^\intercal) = \begin{bmatrix} | & | & | & & | \\ \boldsymbol{\theta}_1(\boldsymbol{x}^\intercal) & \boldsymbol{\theta}_2(\boldsymbol{x}^\intercal) & \boldsymbol{\theta}_3(\boldsymbol{x}^\intercal) & \cdots & \boldsymbol{\theta}_l(\boldsymbol{x}^\intercal) \\ | & | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times l}.$$

For example, a commonly used library is a polynomial library

$$\boldsymbol{\Theta}(\boldsymbol{x}^\intercal) = \begin{bmatrix} | & | & | & | & \\ x_1(t) & \theta(x_2(t) & x_1^2(t) & x_1 x_2(t) & \cdots \\ | & | & | & | & \end{bmatrix}$$

We then measure or compute the time derivative of the data $\dot{\boldsymbol{X}}$ and solve the following equation for $\Xi \in \mathbb{R}^{l \times d}$,

$$\dot{\boldsymbol{X}} = \boldsymbol{\Theta}(\boldsymbol{X})\Xi. \tag{5.1}$$

A key assumption of SINDy is that $\boldsymbol{f}(\boldsymbol{x})$ may be represented by a small number of terms from $\boldsymbol{\Theta}$. Thus, (5.1) is typically solved through sparse regression, using minimization techniques such as sequential least squares thresholding (STLSQ) [31] or LASSO [207]. This then yields a set of nonlinear differential equations

$$\dot{\boldsymbol{x}} = \Xi^\intercal \left(\boldsymbol{\Theta}(\boldsymbol{x}^\intercal)\right)^\intercal. \tag{5.2}$$

This is a symbolic differential equation which may be used for system identification, prediction, and control.

5.2.2 Bayesian Inference for Data-Driven Discovery

Suppose we have data pairs $(\boldsymbol{x}_i, y_i)$ which we would like to fit to a linear regression model. In the frequentist setting, our goal is to determine a vector $\boldsymbol{\beta}$ such that

$$y_i = \boldsymbol{\beta}^\intercal \boldsymbol{x}_i + \epsilon_i, \tag{5.3}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is Gaussian distributed measurement noise with variance σ^2. For Bayesian inference, our goal is to determine a distribution of possible values for $\boldsymbol{\beta}$ given the data, i.e. $p(\beta_j|y)$.

To compute this distribution, we leverage Bayes' rule,

$$p(\beta_j|y) \propto p(y|\beta_j)p(\beta_j). \tag{5.4}$$

The prior term $p(\beta_j)$ incorporates any domain knowledge about the distribution of the β_j's. The term $p(y|\beta_j)$ defines the marginal likelihood, also known as the model evidence of the observed data given a set of parameters.

Here we are primarily interested in the case of determining the governing equations. In the case of identifying ordinary differential equations, the underlying dynamics are governed by an equation of the form $\dot{\boldsymbol{y}}^\intercal = \boldsymbol{\Theta}(\boldsymbol{y}^\intercal)\Xi$, in which case we would like to find the optimal initial condition $\boldsymbol{x}_0$, and parameters Ξ such that

$$\boldsymbol{x}(t_i)^\intercal = \int_0^{t_i} \boldsymbol{\Theta}(\boldsymbol{x}(t)^\intercal)\Xi dt + \epsilon_i. \tag{5.5}$$

Bayes' rule then takes the form

$$p(\Xi, \boldsymbol{x}_0, \boldsymbol{\sigma}, \boldsymbol{\mathcal{D}}|\boldsymbol{X}) \propto p(\boldsymbol{X}|\Xi, \boldsymbol{x}_0, \boldsymbol{\sigma}, \boldsymbol{\mathcal{D}})p(\boldsymbol{\mathcal{D}})p(\boldsymbol{\sigma})p(\Xi)p(\boldsymbol{x}_0), \tag{5.6}$$

where $\boldsymbol{\sigma}$ is the noise level of the data and $\boldsymbol{\mathcal{D}}$ is any additional parameters inferred in the model. In general, computing the posterior distribution is in general not analytically tractable, in which case Monte Carlo sampling methods may be used. For this analysis, we use the No-U-Turn Sampler (NUTS) [89]. This method is a specific Hamiltonian Monte Carlo method which leverages gradient information to efficiently sample high dimensional distributions. NUTS is particularly popular since it requires minimal parameter tuning.

Computing the gradient of (5.5) with respect to Ξ and $\boldsymbol{x}_0$ is particularly challenging. To do this we use the adjoint of the equation. For more information about this see [186]. We note that computing the gradients are incorporated into the PyMC3 library and do not need to be explicitly computed by the user.

With these posterior distributions of the parameters, we may then compute reconstructions and forecasts of the data [64, 209]. In particular, to compute the range of values of $\boldsymbol{x}$ at an arbitrary time t, we can compute the following integral

$$p(\boldsymbol{x}(t)|\boldsymbol{X}) = \iiint p(\boldsymbol{x}(t)|\Xi, \boldsymbol{x}_0, \boldsymbol{\sigma}, \boldsymbol{\mathcal{D}})p(\Xi, \boldsymbol{x}_0, \boldsymbol{\sigma}, \boldsymbol{\mathcal{D}}|\boldsymbol{X})d\Xi d\boldsymbol{x}_0 d\boldsymbol{\sigma} d\boldsymbol{\mathcal{D}}, \tag{5.7}$$

where $p(\boldsymbol{x}(t)|\Xi, \boldsymbol{x}_0, \boldsymbol{\sigma}, \boldsymbol{\mathcal{D}})$ is related to the likelihood used in (5.6). $p(\boldsymbol{x}(t)|\boldsymbol{X})$ is defined to be the posterior predictive distribution (PPD) and the integral in (5.7) is approximated using sampling.

5.2.3 Sparsity Promoting Priors

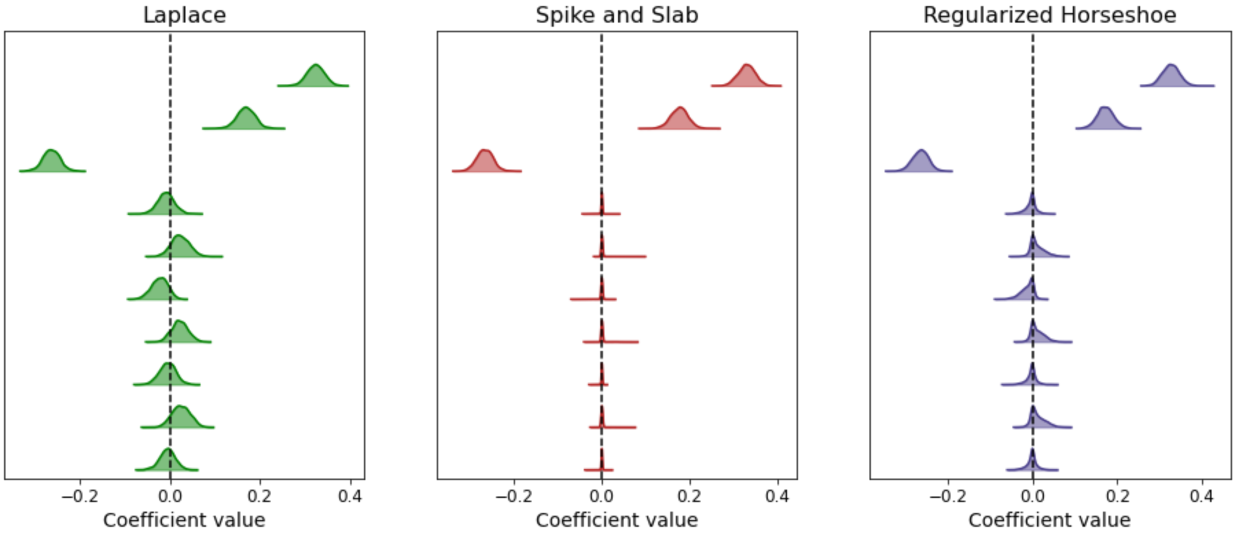


Figure 5.2: Comparison of posterior distributions Laplace, spike and slab, and regularized horseshoe prior for linear regression problem. Both the spike and slab and regularized horseshoe priors promote sparsity in the posterior distributions, while the Laplace prior does not.

Consider the regression problem in (5.3). In many cases we assume only a few components of $\boldsymbol{x}_i$ are correlated with y_i, in which case we expect $\boldsymbol{\beta}$ to be sparse. In the Bayesian setting, there have been many approaches to developing a prior for $\boldsymbol{\beta}$ which induce sparsity. We describe a few of these approaches below, namely the Laplace, spike and slab and the regularized horseshoe priors.

Laplace Prior

Originally proposed by Laplace [51], the Laplace distribution, also known as the double exponential distribution [63], corresponds to the probability distribution function (PDF) $f(x|\mu, b)$ given by

$$f(x|\mu, b) = \frac{1}{2b} e^{-|\frac{x-\mu}{b}|}.$$

We typically denote this distribution as

$$\beta_j|\lambda_j, b \sim \text{Laplace}(\mu, \text{b}).$$

Most notably, the maximum a posterior (MAP) for this prior corresponds to performing regression with L1 regularization of the coefficients β_j [154],

$$\hat{\boldsymbol{\beta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\beta}} P(\boldsymbol{y}|\boldsymbol{\beta}) = \arg\min_{\boldsymbol{\beta}} ||\boldsymbol{y} - \boldsymbol{\beta}^{\intercal}\boldsymbol{X}|_2^2 + \|\boldsymbol{\beta}\|_1 .$$

In the frequentist setting solving this regression problem, also known as the LASSO method, has been shown to yield sparse solutions for $\boldsymbol{\beta}$ [207]. This sparsifying behavior of the Laplace distribution is attributed to the fact that for values of x smaller than b, the distribution is sharply peaked, thus pushing many terms towards 0. For values of x greater than b, the distribution is super-Gaussian allowing elements to escape significant shrinkage.

Although this induces sparsifying behavior in the frequentist case, in the Bayesian setting the resulting posterior distributions are not sparse [41]. In particular, in the Bayesian case we must consider the whole distribution simultaneously. With the Laplace prior every β_j has probability mass simultaneously pushed towards the origin and away from the origin, forcing relevant β_j's to be shrunk towards the origin and irrelevant terms to have significant probability mass far away from the origin.

To illustrate this we simulate 400 samples of data $(\boldsymbol{x}_i, y_i)$ satisfying (5.3), where $\boldsymbol{x}_i \sim \mathcal{N}(0,1) \in \mathbb{R}^{10}$, $\epsilon_i \sim \mathcal{N}(0, 0.5^2)$ and $\boldsymbol{\beta} \in \mathbb{R}^{10}$ is chosen to be the sparse vector

$$\boldsymbol{\beta} = [0.3, 0.2, -0.3, 0, 0, 0, 0, 0, 0, 0]^{\intercal}.$$

We perform Bayesian inference to estimate $\boldsymbol{\beta}$ using a Laplace prior, and in Figure 5.2 we plot the resulting posterior distribution. We note that using the Laplace prior, the posterior distributions are centered about the true value $\boldsymbol{\beta}$. However, many distributions are peaked at nonzero values, making it difficult to differentiate between relevant and irrelevant variables. Further, due to the wide widths of all the distributions, samples from this posterior distribution will not be sparse. To better enforce sparsity in a Bayesian setting and induce sparse realizations, the distribution of each β_j must either be fully shrunk towards the origin or pushed away from the origin. In sections 5.2.3 and 5.2.3 we discuss two priors which satisfy these properties.

Spike and Slab Prior

The Spike and Slab prior is one of the popular sparsifying priors and is typically referred to as the "gold standard" [144, 106, 138] for sparsity in the Bayesian setting. For this prior we assume that $\boldsymbol{\beta}$ is the product of a Bernoulli and normal distribution. In particular,

$$\beta_j | \lambda_j, c \sim \mathcal{N}(0, c^2)\lambda_j \tag{5.8}$$

$$\lambda_j = \text{Ber}(\pi), \tag{5.9}$$

where π is the probability that λ_j is 1. Otherwise λ_j is 0. From this we immediately see that if λ_j is 1, then the jth term belongs to the model, and β_j follows a normal distribution. If λ_j is 0 then the jth term is not in the model β_j a delta spike at zero.

The distribution may be relaxed to

$$\beta_j | \lambda_j, c \sim \lambda_j \mathcal{N}(0, c^2) + (1 - \lambda_j)\mathcal{N}(0, \epsilon^2)$$

$$\lambda_j = \text{Ber}(\pi),$$

where $\epsilon \ll c$. This is similar to before, except when $\lambda_j = 0$, β_j follows a narrow normal distribution with variance ϵ, in contrast to a delta spike.

This prior for $\boldsymbol{\beta}$ is very intuitive and has shown a robust performance on data. In Figure 5.2, we plot the resulting posterior distribution for the example in section 5.2.3. Most notably we see that, similar to the Laplace prior the spike and slab prior extracts out wide distributions for the three nonzero coefficients. The seven zero coefficients, on the other hand, are spikes at the origin. Consequently, any sampling of this posterior distribution will be sparse. Compared to the Laplace distribution this makes it much more clear which terms are nonzero. Furthermore, the means of the λ_j's provides us with an explicit likelihood that a particular β_j is relevant to the model.

Although this prior has many beneficial properties, one downside is that due to the incorporation of Bernoulli distribution, this prior is not differentiable. Consequently, computation of the posterior distribution scales exponentially with the number of dimensions. Many approaches have been made to develop smooth approximations to this distribution. We discuss one recent approach in section 5.2.3.

Regularized Horseshoe Prior

The horseshoe prior and the recently developed regularized horseshoe prior are smooth priors, that have shown comparable performance to the spike and slab model. The horseshoe is defined as

$$\begin{aligned} \beta_i|\lambda_i, \tau &\sim \mathcal{N}(0, \lambda_i^2\tau^2) \\ \lambda_i &\sim \mathrm{C}^+(0,1) \\ \tau\ &\sim \mathrm{C}^+(0,\tau_0), \end{aligned}$$

where C^+ denotes the half-Cauchy distribution [23, 38, 39]. The key intuition for this prior is that τ promotes global sparsity, shrinking the posterior distributions of all β_i. λ_i, also known as the local shrinkage parameters have a half-Cauchy prior. This prior has a long tail allowing some of the β_i's to escape significant shrinkage. Many analyses have focused on choosing an optimal value of τ_0. [159] recommend to choose a value of $\tau_0 = \frac{m_0}{M-m_0}\frac{\sigma}{N}$, where is m_0 is an estimate of the number of relevant β_j. We note that decreasing the value of τ, increases the sparsity in $\boldsymbol{\beta}$.

One downside of the horseshoe is that relevant terms which "escape" are unregularized, and thus elements of the posterior distribution may become arbitrarily large. One notable advancement to remedy this by [159] proposed to include a small amount of regularization on $\boldsymbol{\lambda}$,

$$\begin{aligned} \beta_i|\tilde{\lambda}_i,\tau &\sim \mathcal{N}(0,\tilde{\lambda}_i^2\tau^2) \\ \tilde{\lambda}_i &\sim \frac{c\lambda_i}{\sqrt{c^2+\tau^2\lambda_i^2}} \\ \lambda_i &\sim \mathrm{C}^+(0,1) \\ c^2 &\sim \text{Inv-}\Gamma\left(\frac{\nu}{2},\frac{\nu}{2}s^2\right) \\ \tau\ &\sim \mathrm{C}^+(0,\tau_0). \end{aligned} \tag{5.10}$$

For small values of λ_i, $\lambda_i\tau \ll c$, and therefore $\tilde{\lambda}_i$ approaches λ, thus approximating the original horseshoe prior. However, for large values of λ_i,$\lambda_i\tau \gg c$, $\tilde{\lambda} \to c/\tau$, and hence β_i is normally distributed about c. This regularizes β_i, constraining it to be on the order of c.

We illustrate the performance of this prior in Figure 5.2, for the example in section 5.2.3. Similar to the spike and slab model the nonzero coefficients have wide distributions. The zero coefficients, on the other hand are more spiked than those of the Laplace prior, thus promoting sparser realizations.

Unlike for the spike and slab prior there is no explicit estimate for the inclusion probabilities. One approach to computing a pseudo-probability is to compute the maximum likelihood with a flat prior (i.e. no prior) and compare it to the maximum likelihood from the regularized horseshoe. The ratio of these two values is called the shrinkage

$$\kappa_i = \frac{\hat{\boldsymbol{\beta}}_i^{RH}}{\hat{\boldsymbol{\beta}}_i^{Flat}}$$

The shrinkage of the coefficients has been used to define inclusion pseudo-probabilities for the model. In general these ratios may not lie between 0 and 1. To improve numerical stability, for our analysis we replace the flat prior with a Gaussian prior.

5.3 UQ-SINDY

In this section we combine advancements in Bayesian inference of ODEs and sparsity promoting priors to propose a UQ-SINDy framework that quantifies both the the uncertainty of coefficients and the inclusion probabilities of the model. In particular, within this framwork we introduce two methods: spike and slab SINDy (ss-SINDy) and a smooth relaxation regularized horseshoe SINDy (rh-SINDy). The ss-SINDy provides state of the art performance for estimating uncertainty of coefficients and inclusion probability, while the rh-SINDy is a smooth approximation which shows comparable performance. We outline this framework below.

5.3.1 Method

We start with a set of time series measurements $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2 \dots \boldsymbol{x}_n]^\intercal \in \mathbb{R}^{n\times d}$. We assume that our data is governed by some underlying equation

$$\frac{d\boldsymbol{x}^\intercal}{dt} = \boldsymbol{\Theta}(\boldsymbol{x}^\intercal)\boldsymbol{\Xi}$$

for some sparse $\boldsymbol{\Xi}$. Our goal is to determine the posterior distribution $p(\boldsymbol{\Xi}, \boldsymbol{x}_0, \boldsymbol{\sigma}, \boldsymbol{\mathcal{D}}|\boldsymbol{X})$.

Step 1: **Construct library**: We first compute a library $\boldsymbol{\Theta} : \mathbb{R}^{1\times d} \to \mathbb{R}^{1\times m}$ of candidate terms:

$$\boldsymbol{\Theta}(\boldsymbol{x}^\intercal) = [x_1, x_2, x_3, x_1^2, x_1x_2, \dots].$$

We emphasize that here $\boldsymbol{\Theta}$ is a symbolic function of the inputs $y_1, \dots, y_d$. This is in constrast to the original SINDy algorithm, in which $\boldsymbol{\Theta(X)}$ is a fixed matrix computed from the initial data.

Step 2: **Construct model priors and model likelihood** In the case of normally distributed noise the likelihood, takes the form

$$p(\boldsymbol{X}|\Xi, \boldsymbol{x}_0, \boldsymbol{\sigma}, \boldsymbol{\mathcal{D}}) = \prod_{j=1}^{d} \frac{1}{\sigma_j\sqrt{2\pi}} e^{\frac{-||\boldsymbol{x}_j - \int_{t_0}^{\boldsymbol{t}} \boldsymbol{\Theta}(\boldsymbol{x}(t')^\intercal)\boldsymbol{\Xi} dt'||^2}{\sigma_j^2}} \tag{5.11}$$

For some cases, the values of $\boldsymbol{X}$ takes nonnegative values, such as for populations, in which case we may choose to use a lognormal likelihood instead.

$$p(\boldsymbol{X}|\Xi, \boldsymbol{x}_0, \boldsymbol{\sigma}, \boldsymbol{\mathcal{D}}) = \prod_{j=1}^{d} \frac{1}{\sigma_j\sqrt{2\pi}} e^{\frac{-||\log(\boldsymbol{x}_j) - \log(\int_{t_0}^{\boldsymbol{t}} \boldsymbol{\Theta}(\boldsymbol{x}(t'))\boldsymbol{\Xi} dt')||^2}{\sigma_j^2}} \tag{5.12}$$

We must choose priors for several parameters, in particular, the noise level σ_j, initial condition $\boldsymbol{x}_0$ in addition to any other model parameters $\boldsymbol{\mathcal{D}}$. These priors are typically chosen using knowledge about about the type of parameter (i.e. the parameter is nonnegative) and the scales of the data.

Step 3: **Choose sparsity promoting prior** Following section 5.2.3, for *spike and slab SINDy (ss-SINDy)* we use the prior

$$\begin{aligned} \Xi_j|\lambda_j, c &\sim \text{Laplace}(0,1)\lambda_j \\ \lambda_j &= \text{Ber}(\pi), \end{aligned}$$

where λ_j is the inclusion of term j in the model. We note that we use a Laplace here instead of a normal distribution due to its wider tails. For *regularized horseshoe SINDy (rh-SINDy)* we use the prior

$$\begin{aligned} \Xi_{i,j}|\tilde{\lambda}_{i,j}\tau &\sim \mathcal{N}(0, \tilde{\lambda}_{i,j}^2\tau^2) \\ \tilde{\lambda}_{i,j} &\sim \frac{c\lambda_{i,j}}{\sqrt{c^2 + \tau^2\lambda_{i,j}^2}} \\ \lambda_i &\sim \text{C}^+(0,1) \\ c^2 &\sim \text{Inv-}\Gamma\left(\frac{\nu}{2}, \frac{\nu}{2}s^2\right) \\ \tau &\sim \text{C}^+(0, \tau_0). \end{aligned}$$

Step 4: **Bayesian Inference** Using the priors and likelihood, we perform Bayesian inference using (5.6) to compute the inclusion probability and posterior distributions for each term in the model. Using (5.7), we may also compute a posterior predictive distribution for model reconstruction or forecasting. For our analysis, we use PyMC3 [177] and sunode (a python wrapper for the CVODES library for efficiently solving ODEs) to efficiently perform Bayesian inference [186].

5.3.2 Synthetic Examples

In this section we apply the spike and slab and regularized horseshoe priors in the UQ-SINDy framework and illustrate their performance on three examples: two synthetic and one real world time series of lynx and hare populations. For each example we compute the likelihood of each term belonging to the underlying dynamical equations providing both an inclusion probability and a distribution of likely values for each parameter. We compare these results to the original SINDy algorithm and show that UQ-SINDy significantly outperforms SINDy in identifying the underlying dynamics and associated parameter values.

Nonlinear Oscillator

To illustrate this method, we first consider a damped nonlinear oscillator, which takes the form

$$\dot{x} = \alpha x^3 + \beta y^3 \tag{5.13}$$

$$\dot{y} = \gamma x^3 + \delta y^3. \tag{5.14}$$

Following [165], we use the values $\alpha = -0.1, \beta = -2, \gamma = 2, \delta = -0.1$. We use an initial condition of $[x_0, y_0] = [2, 0]$. We sample the time series over the interval $t \in [0, 20]$, with a sampling period of $\Delta t = 0.5$.

Here we add a small amount of Gaussian distributed noise $\mathcal{N}(0, 0.02^2)$ to the measurements. The observed trajectory is shown in Figure 5.3. We use a library of polynomial terms

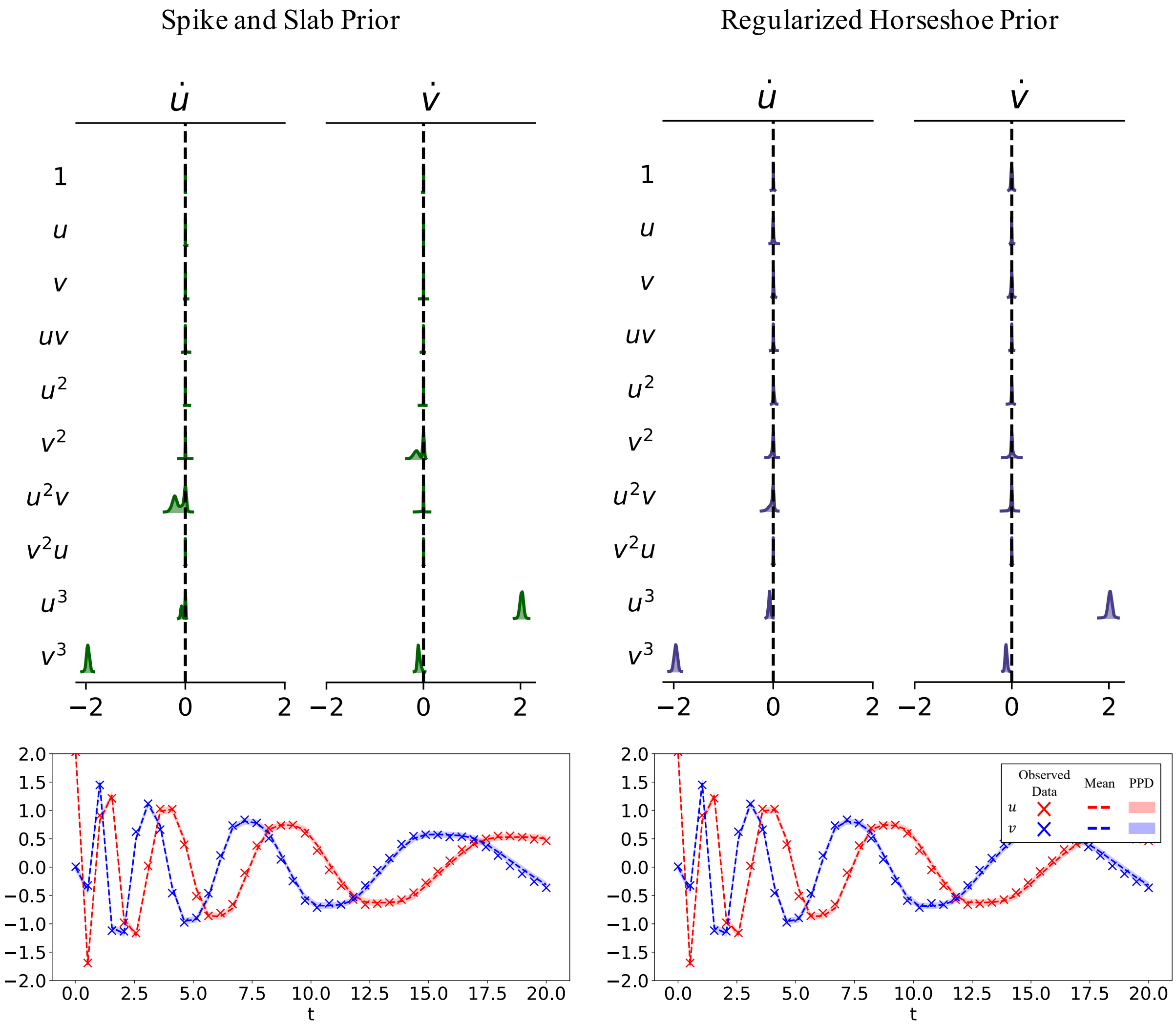


Figure 5.3: UQ-SINDy applied to nonlinear oscillator system with Gaussian noise. Posterior distributions of parameters in model for spike and slab prior and regularized horseshoe prior. For both priors the terms in the model are correctly identified, with nonzero terms having wide distributions and zero terms corresponding to spikes at the origin. We plot the trajectories directly below. We additionally plot the 90% credibility intervals and means for the associated posterior predictive distributions (PPDs).

$\boldsymbol{\Theta}(x, y) = [1, x, y, xy, x^2, y^2, x^2y, y^2x, x^3, y^3]$. Since we have two sets of measurements, Ξ is a 2×10 matrix. Our goal is to perform sparse Bayesian inference on the differential equation

$$\begin{bmatrix} \dot{x} & \dot{y} \end{bmatrix} = \begin{bmatrix} 1 & x & y & x^2 & y^2 & xy & x^2y & y^2x & x^3 & y^3 \end{bmatrix} \Xi.$$

Since we have Gaussian distributed noise we choose to use the likelihood in (5.11). For the noise level and initial condition we use $\boldsymbol{\sigma} \sim \Gamma(1, 0.1)$ and $x_0, y_0 \sim \text{Laplace}(0, 1)$, respectively.

We first apply the spike and slab prior for Ξ as in (5.9), with parameters $\pi = 0.8$, and $c = 1$, and perform Bayesian inference. We plot the posterior distributions for Ξ in Figure 5.3. As we expect from the distributions of Ξ we see that sixteen of the terms correspond to spikes centered on the origin and four of the terms have wider distributions centered away from zero. Further, these four nonzero terms correspond directly to the four terms in the model. The modes of these distributions are shown in Table 4. These values are close in value to the true values and are well within the 90% credibility interval of these values. Next, we present the model inclusion probabilities λ_j in Table 5.1. All four nonzero terms have inclusion probabilities close to 1, while the other 16 terms have values close to zero. We compare the means of the posterior distributions to that of the SINDy model. Note that for this the SINDy model is not sparse and does not learn the correct model.

Next we apply the regularized horseshoe prior for Ξ as in (5.10), with parameters $\tau_0 = 0.1, \nu = 2$ and $s = 4$. We note that these values of ν and s are suggested in [159]. We plot the posterior distributions for Ξ in Figure 5.3. We see that the regularized horseshoe provides comparable distributions and similar maximum values.

Last, in Figure 5.3, we plot the predictive posterior distribution (PPD) of the reconstruction. In particular, for each sampling time we compute the mean values (dashed lines), the 5th and 95th percentiles (shaded regions), in addition to the measured values (crosses). Both the regularized horseshoe and spike and slab show similar confidence intervals. Further, we see that the true data lies well within these predictive posterior distributions, as expected.

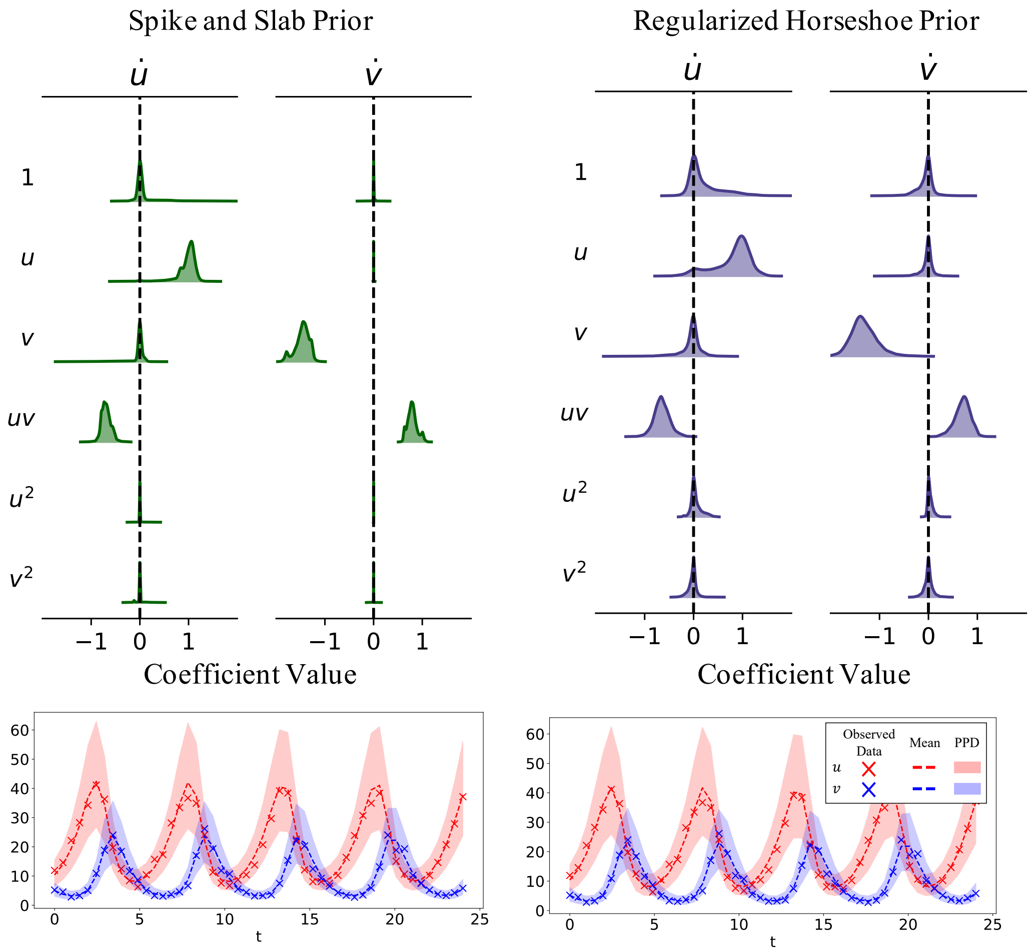


Figure 5.4: UQ-SINDy applied to synthetic Lotka Volterra system with lognormal noise. Posterior distributions of parameters in model for spike and slab prior and regularized horseshoe prior. For both priors the terms in the model are correctly identified, with nonzero terms having wide distributions and zero terms corresponding to spikes at the origin. We plot the trajectories directly below. We additionally plot the 90% credibility intervals and means for the associated posterior predictive distributions (PPDs).

Lotka Volterra Model

The Lotka Volterra model, also commonly refered to as the predator prey model, is a popular system used to model the interaction between two competing groups [71, 216]. Originally developed by Lotka to model chemical reactions [134], the system has also been studied as a model in economics [75] and for biological systems [119, 205, 215]. We explore one biological example in Section 5.3.3. This system is governed by two nonlinear differential equations

$$\begin{aligned}\dot{x} &= \alpha x - \beta xy\\ \dot{y} &= -\gamma y + \delta xy.\end{aligned}$$

For this example, we simulate the system with the commonly used initial condition $[x_0, y_0] = [10, 5]$ and parameters $\alpha = 1, \beta = 0.1, \gamma = 1.5, \delta = 0.075$, which results in a periodic trajectory. We sample 50 snapshots over a time interval of $t \in [0, 24]$. We additionally multiply the trajectory with multiplicative noise following the distribution Lognormal$(0, 0.1)$. The lognormal distribution is nonnegative and thus commonly used for systems, like populations, for which possible values must be greater or equal to zero.

The resulting time series is shown in Figure 5.4 from which we see that the trajectory covers approximately four periods of oscillation. We apply UQ-SINDy for both the spike and slab prior and regularized horseshoe prior. We use a library containing all possible constant, linear and quadratic terms (12 terms in total). In Table 5.2 we show the likelihood of each term belonging to the the model for the spike and slab prior. We see significantly higher probabilities for the four true nonzero terms compared to all other terms. Further we can compute pseudo-probabilities for the regularized horseshoe. Although these values are not constrained between zero and the relevant terms are easily identified with values near to or greater than 1. Next, we plot the posterior distributions. From this we immediately see that for both priors the parameters which belong to the model have broad distributions centered about the true means, while the other 8 terms have narrow peaks centered about 0. In Table 5.2 we compare the peak values of these posterior distributions to the true values. We additionally apply SINDy to the data and compare those values. We see that SINDy is unable to identify the correct dynamics. Furthermore, we note that due to their sparsifying behavior, the peak values for both the spike and slab and regularized horseshoe priors are close to the true values.

Last, in Figure 5.4, we plot the predictive posterior distribution of the reconstruction, computing the mean values (dashed lines), the 5th and 95th percentiles (shaded regions) and the observed values (crosses). Like for the nonlinear oscillator, both the regularized horseshoe and spike and slab show similar confidence intervals. We also note that the range of values in the posterior predictive distribution is much wider than for the nonlinear oscillator, which is expected due to the increased noise present in these measurements.

5.3.3 Lynx Hare Population Model

As a final example, here we illustrate the performance of the sparsifying priors for modeling the populations of two species in Canada. In particular, this data consists of measurements by the Hudson Bay Company of lynx and hare pelts between 1900 and 1920 (see Figure 5.5) [37, 87]. The number of pelts between for these two species is thought to be proportional to the true populations. Hares are a herbivorous relative of the rabbit while the lynx is a type of wildcat whose diet depends heavily on hares. This predator prey interdependence between the two species has been shown to be well characterized to first order by the Lotka Volterra model in Section 5.3.2, where u and v correspond to the population of hares and lynxes, respectively.

In Figure 5.5 we plot the number of pelts recorded for these two species. Modeling this data with SINDy is particularly challenging as we have relatively few samples ($n = 21$) which cover only two cycles. In addition, factors including the weather and the consistency of trapping between years adds a high level of uncertainty to the measurements.

Here we compare the performance of the ss-SINDY, and rh-SINDY. The library as in the previous example contains all constant, linear and quadratic terms. In addition we normalize the data. From the posterior distributions for the spike and slab prior we see the distinct nonzero peaks corresponding for the terms in equation 5.3.2. The likelihood of these four terms belonging to the model are very high. We additionally see a small peak near zero in the u term for the $\dot{u}$. This term is highly correlated with a nonzero term appearing in the constant term. We see a similar peak for the regularized horseshoe model. However, in this

	TRUE	SINDy	SS-Prior	Rh-prior
$\dot{u} : 1$	0	0.44	0.00	0.00
$\dot{v} : 1$	0	0.05	0.00	-0.01
$\dot{u} : u$	0	0.58	0.00	0.00
$\dot{v} : u$	0	-0.60	0.00	0.00
$\dot{u} : v$	0	0.75	0.00	0.00
$\dot{v} : v$	0	-0.12	0.00	0.00
$\dot{u} : uv$	0	-0.33	0.00	0.00
$\dot{v} : uv$	0	-0.07	0.00	0.00
$\dot{u} : u^2$	0	-1.80	0.00	0.01
$\dot{v} : u^2$	0	-0.37	0.00	0.00
$\dot{u} : v^2$	0	0.47	0.00	-0.01
$\dot{v} : v^2$	0	0.34	-0.08	0.00
$\dot{u} : u^2v$	0	0.39	-0.12	-0.03
$\dot{v} : u^2v$	0	0.47	-0.01	-0.01
$\dot{u} : v^2u$	0	0.55	0.00	0.00
$\dot{v} : v^2u$	0	-0.35	0.00	0.00
$\dot{u} : u^3$	-0.1	-1.44	-0.04	-0.08
$\dot{v} : u^3$	2	-0.48	2.03	2.02
$\dot{u} : v^3$	-2	0.02	-1.96	-1.96
$\dot{v} : v^3$	-0.1	-0.09	-0.10	-0.12

	SS-prior	Rh-prior
$\dot{u} : 1$	0.00	-0.05
$\dot{v} : 1$	0.01	-0.33
$\dot{u} : u$	0.10	-0.03
$\dot{v} : u$	0.00	-0.08
$\dot{u} : v$	0.05	0.04
$\dot{v} : v$	0.08	-0.06
$\dot{u} : uv$	0.08	-0.21
$\dot{v} : uv$	0.06	0.00
$\dot{u} : u^2$	0.07	0.01
$\dot{v} : u^2$	0.07	0.00
$\dot{u} : v^2$	0.142	0.03
$\dot{v} : v^2$	0.5	-0.02
$\dot{u} : u^2$	0.70	0.03
$\dot{v} : u^2$	0.17	0.06
$\dot{u} : v^2$	0.01	0.05
$\dot{v} : v^2$	0.02	-0.14
$\dot{u} : u^3$	0.50	1.20
$\dot{v} : u^3$	1.00	1.01
$\dot{u} : v^3$	1.00	1.00
$\dot{v} : v^3$	1.00	0.92

Table 5.1: Left: Posterior distributions $p(\boldsymbol{\Xi})$ for each term in model for nonlinear oscillator. Right: Corresponding inclusion probabilities.

case the peak is even more pronounced. Looking at Table 5.3, we see that the spike and slab model correctly identifies the terms and places high values for the inclusion probabilities

	TRUE	SINDy	SS-Prior	Rh-prior
$\dot{u}:1$	0	0.861	0	0.04
$\dot{v}:1$	0	0	0	0
$\dot{u}:u$	1	0.39	1.06	0.97
$\dot{v}:u$	0	0	0	-0.01
$\dot{u}:v$	0	-0.64	0	-0.01
$\dot{v}:v$	-1.5	-1.24	-1.44	-1.35
$\dot{u}:uv$	-0.68	-0.226	-0.74	-0.66
$\dot{v}:uv$	0.82	0.664	0.78	0.7
$\dot{u}:u^2$	0	0	0	0.01
$\dot{v}:u^2$	0	0	0	0.02
$\dot{u}:v^2$	0	0	0	0
$\dot{v}:v^2$	0	0	0.01	0.01

	SS-prior	Rh-prior
$\dot{u}:1$	0.36	0.26
$\dot{v}:1$	0.17	-0.32
$\dot{u}:u$	1	3.29
$\dot{v}:u$	0.13	0.07
$\dot{u}:v$	0.27	0.15
$\dot{v}:v$	1	1.01
$\dot{u}:uv$	1	1.10
$\dot{v}:uv$	1	1.19
$\dot{u}:u^2$	0.09	0.27
$\dot{v}:u^2$	0.01	0.24
$\dot{u}:v^2$	0.19	-1.92
$\dot{v}:v^2$	0.03	-0.02

Table 5.2: Left: Posterior distributions $p(\Xi)$ for each term in model for Lotka Volterra data. Right: Corresponding inclusion probabilities.

for the four terms in the model. The regularized identifies horseshoe three terms correctly. Last, in Figure 5.5, we plot the predictive posterior distribution (PPD) of the reconstruction, computing the mean values (dashed lines), the 5th and 95th percentiles (shaded regions) and the observed values (crosses). We note that all of the original time series lies within these confidence bounds, as we expect.

5.4 *Conclusions and Future Work*

In this work, we proposed a new framework for identifying governing equations directly from data. We leveraged advances in Bayesian inference of ODE's and sparse regression to identify relevant terms, quantifying the uncertainty in the values of each parameter and its

	Param est	SINDy	SS prior	RH prior
$\dot{u} : 1$	0	0	0	0.01
$\dot{v} : 1$	0	0	0	0
$\dot{u} : u$	0.55	0.384	0.47	0
$\dot{v} : u$	0	0	0	-0.01
$\dot{u} : v$	0	-0.143	0	0
$\dot{v} : v$	-0.84	-0.613	-0.76	-0.74
$\dot{u} : uv$	-0.455	-0.226	-0.51	-0.42
$\dot{v} : uv$	0.5433	0.664	0.52	0.52
$\dot{u} : u^2$	0	0	0	0.01
$\dot{v} : u^2$	0	0	0	0
$\dot{u} : v^2$	0	0	0	0
$\dot{v} : v^2$	0	0	0	-0.01

	SS-prior	Rh-prior
$\dot{u} : 1$	0.47	0.035
$\dot{v} : 1$	0.35	0.00
$\dot{u} : u$	0.85	0.00
$\dot{v} : u$	0.34	0.02
$\dot{u} : v$	0.54	0.00
$\dot{v} : v$	0.99	0.73
$\dot{u} : uv$	0.96	0.78
$\dot{v} : uv$	1	2.01
$\dot{u} : u^2$	0.581	0.03
$\dot{v} : u^2$	0.08	0.02
$\dot{u} : v^2$	0.31	-0.50
$\dot{v} : v^2$	0.35	-0.06

Table 5.3: Left: Posterior distributions $p(\Xi)$ for each term in model for lynx and hare population data. Right: Corresponding inclusion probabilities.

inclusion probability. Further, this method is very robust to noise and can accommodate sparse samples.

Going forward, one of the primary limitations of this method is its scalability to very large libraries of terms. This is primarily due to the computational intensity to sample high dimensional spaces using Markov Chain Monte Carlo. One remedy for this is to use variational inference, which matches classes of functions to the posterior distribution. This method has been particularly effective for high dimensional model, most notably neural networks, with comparable accuracy to sampling methods.

In addition, for this work we are primarily focused on situations in which coordinate which induce a sparse representation is known. However, in general this ”effective” set of

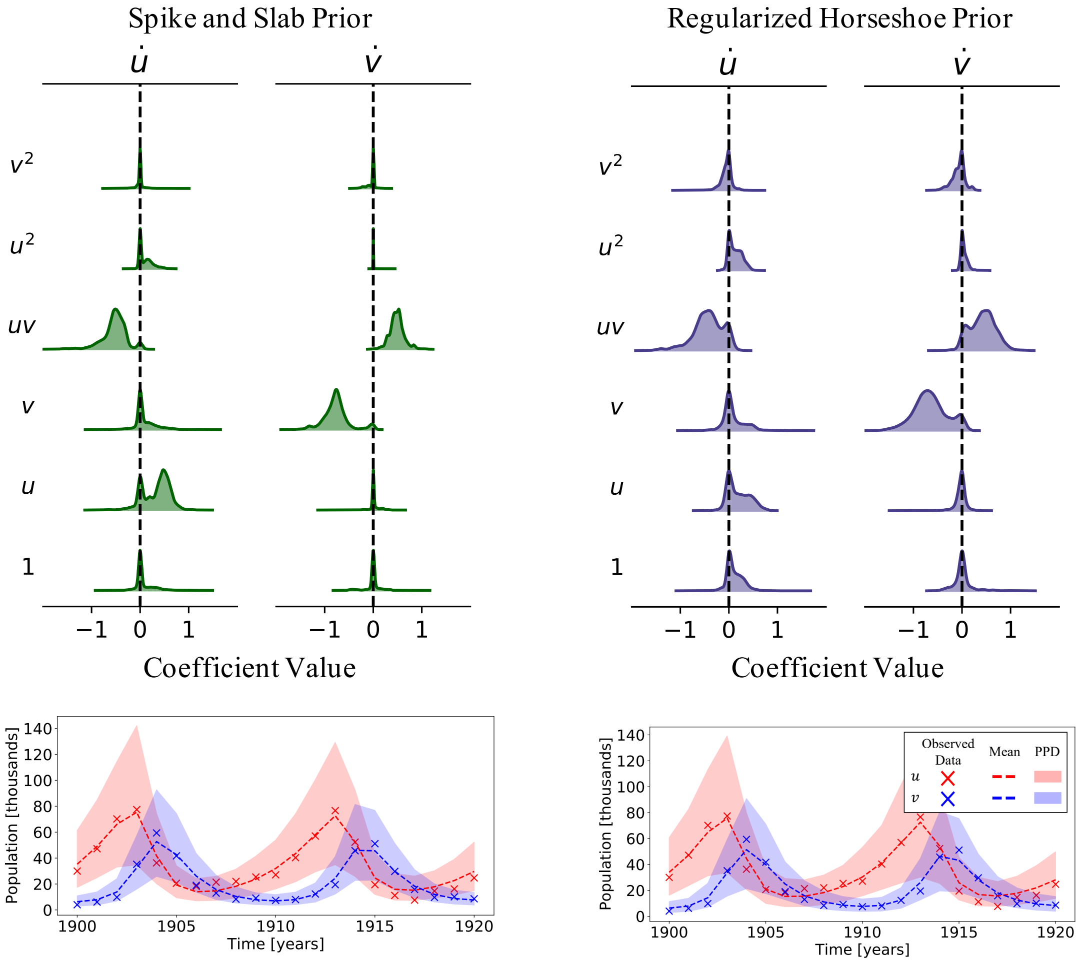


Figure 5.5: UQ-SINDy applied to lynx and hare population data. Posterior distributions of parameters in model for spike and slab prior and regularized horseshoe prior. For both priors the terms in the model are correctly identified, with nonzero terms having wide distributions and zero terms corresponding to spikes at the origin. We plot the trajectories directly below. We additionally plot the 90% credibility intervals and means for the associated posterior predictive distributions (PPDs).

coordinates may be unknown. Recent work has incorporated SINDy into neural network architectures, to simultaneously learn the parsimonious governing equations and associated coordinates [42]. Incorporating UQ-SINDy into these neural network architecture could greatly improve robustness and confidence in these models and the associated forecasts.

Chapter 6

CONCLUSIONS

In this thesis, we discussed four contributions to the theory and methods of data driven modeling of physical systems. We first proposed mean subtraction as a natural and computationally efficient preprocessing step when performing the dynamic mode decomposition. This is defended by rigorous theory and is contrast to previous results that have shown mean subtraction to yield undesirable results. Next, we developed a novel modal decomposition, STIMD, which decomposes spatiotemporal data into time varying linear dynamics and associated spatial modes. We showed that STIMD outperforms comparable methods such as DMD, SVD, and ICA, and illustrated this method on two real world examples, gravitational wave data and neural recordings from a rodent hippocampus. Third, we explored the recently developed HAVOK method, which combines time delay embeddings and dimensionality reduction to decompose nonlinear dynamics into a linear model plus intermittent forcing. Utilizing properties of the Frenet-Serret frame we showed that under certain conditions the linear model yields a specific antisymmetric tridiagonal structure. With this new perspective we provided a simple modification, which we denote structured HAVOK (sHAVOK), which promotes this structure. Fourth, we proposed a UQ-SINDy framework which leverages sparse Bayesian inference to identify governing equations with uncertainty quantification in both parameter values and inclusion probabilities for each term. This method is illustrated on two synthetic examples and a real world dataset consisting of lynx and hare populations. In this chapter we summarize these contributions and propose possible areas of future research.

6.1 DMD with Centering

In chapter 2, we showed that centering data improves the performance of the dynamic mode decomposition (DMD). In particular, we showed that DMD with centering is equivalent to an additional affine term in the DMD model. However, it is not the same as the discrete Fourier transform, contrary to previous results. In the case of affine linear dynamics, we showed that DMD with centering yields the same spectra as DMD without centering in certain cases. However, if the data is full rank, then DMD with centering can recover the dynamics, while DMD without centering cannot. We illustrated this theory on video surveillance data and simulations of the Lorenz system. As predicted by theory, for the video surveillance data the linear model from DMD is low rank, and hence DMD with centering is equivalent to DMD without centering. However, for the Lorenz system, in which the linear dynamics are full rank, DMD with centering provides a better reconstruction in the presence of noise.

Further, by considering the DMD with centering as subtracting the zero frequency mode from the data, we generalized our results to show how to subtract modes of arbitrary frequencies within the DMD framework. This is particularly relevant in the case where there is prior knowledge of a subset of frequencies. We illustrated this on ECOG data, showing that we can remove 60Hz line noise from the signal of interest.

6.2 STIMD

In chapter 3, we introduced a novel modal decomposition, called the spatiotemporal intrinsic mode decomposition (STIMD), which leverages spatial correlations and nonlinear matching pursuit (NMP) to decompose data into linear combinations of intrinsic mode functions (IMFs). IMFs have the beneficial properties that they can accurately model nonstationary dynamics, a key advantage over techniques such as DMD and the Fourier transform. Furthermore, the IMFs have a well-defined Hilbert transform, which can be used to measure the underlying instantaneous frequencies of the data, a key benefit over methods, such as SVD and ICA. By utilizing properties of the NMP method we showed that STIMD may be used for short term forecasting. Furthermore, we showed that STIMD is highly robust to noise and initial conditions required for the method. We illustrated STIMD on two examples, namely neural recordings from a rodent hippocampus and gravitational wave signals from merging black holes in the LIGO experiment. For both examples, we showed that STIMD is able to extract out the relevant frequencies from the data.

6.3 sHAVOK

In chapter 4, we unified results from time delay embeddings, dimensionality reduction techniques, and differential geometry to decompose nonlinear dynamics into a sparse linear dynamical model plus a forcing term. We showed that the linear model has a very specific structure. In particular it is antisymmetric with nonzero elements along the sub- and superdiagonals. We connected this to previous results showing similar structure for the Lorenz system in the HAVOK algorithm.

With this new perspective we explored the limitations of this structure and proposed modifications to HAVOK to induce this structure. We suggested interpolation when the dynamics are sparsely sampled. In the case where the length of data is limited, we proposed a small modification called structured HAVOK (sHAVOK), which is more closely connected to theory. We illustrated sHAVOK on three synthetic examples and two real world examples, a double pendulum and measles outbreak data. In these cases, the sHAVOK model induces more stable dynamics and better reconstructions.

6.4 UQ-SINDy

In chapter 5, we proposed a new uncertainty quantification for sparse identification of nonlinear dynamics (UQ-SINDy) framework, which utilizes sparse priors to identify governing equations. In particular, we proposed to use two sparsifying priors: the spike and slab prior, which is the gold standard for sparse Bayesian inference, and the regularized horseshoe, a smooth relaxation. We illustrated this framework on two synthetic examples, a nonlinear oscillator and a Lotka-Volterra system. We find that UQ-SINDy well outperforms the original SINDy method in accurately identifying dynamics in addition to putting tight bounds on the uncertainty. We further applied these methods to lynx and hare population data, confirming the hypothesis that the data is well modeled using the Lotka-Volterra equations.

6.5 Future Directions

- **Nonsequential time series**

  In chapter 2, many theorems are based on the assumption the measurements were sampled at a fixed frequency. However, one of the most popular DMD methods, exact DMD, can accommodate non-uniformly sampled measurements. Although likely, it remains to be shown that the DMD modes are unique for nonuniformly sampled data and that centering will be beneficial in this general case. One possible starting point for proving these results is the theory of exponential Vandermonde matrices [169, 231].

  Additionally, our analysis focused on the SVD based DMD method, exact DMD. However, many other DMD algorithms exist, including forward backward DMD [50], total least squares DMD [85], and optimized DMD [12]. The effects of centering for these alternative methods remain to be explicitly characterized.

- **Efficient for higher dimensional datasets**

  The STIMD method was shown to be extremely robust to measurement noise and initial conditions, which were both explicitly characterized in chapter 3. One limitation of the method is the optimization problem in algorithm 2. The current optimization does not explicitly use gradient information. Consequently, the method scales exponentially with dimensionality making it prohibitively expensive for high dimensional datasets. We suggest exploring different optimization techniques by leveraging gradient information to improve efficiency without sacrificing robustness.

  We additionally note that this method leverages the NMP method, which extracts signals with interwave frequency and amplitude modulation. In other words, this method requires that the modulation frequency is lower than the frequency of the wave. However, many signals such as electrocardiogram data contain intrawave modulation. Extracting IMFs with intrawave modulation is significantly more difficult, though recent advancements have been made for one dimensional signals [94]. We believe this could be incorporated into the STIMD framework, thus generalizing this model to a greater class of functions.

- **Connecting time delay embeddings, dimensionality reduction and the Frenet-Serret frame**

  In chapter 4, we unified key results from the theory of time delay embeddings, dimensionality reduction techniques, and differential geometry to explain the antisymmetric tridiagonal structure observed in HAVOK. This new perspective enabled us to discover modifications to the method, which were closer connected to theory and yielded improved model reconstruction in practice.

  Differential geometry, dimensionality reduction and time delay embeddings are all well-established fields and by fully exploring these connections greater insights and further improvements could be made. More specifically, by connecting HAVOK and the Frenet-Serret frame, we discovered the significance of $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ being orthogonal matrices, resulting in the development of modifications, which we denote sHAVOK. These modifications resulted in improved stability and model reconstruction. Furthermore, with this new perspective we recognized that HAVOK may be thought of as a first order forward difference scheme, approximating the derivative of $\boldsymbol{V}$ by $\left(\boldsymbol{V}_2 - \boldsymbol{V}_1\right)/\Delta t$. By modifying HAVOK to behave like a central order difference method, we have observed that the dynamic model becomes more antisymmetric, which more closely aligns with Frenet-Serret theory.

- **Scalibility and coordinate discovery in UQ-SINDy framework**

  In chapter 5, we proposed a new framework for identifying governing equations directly from data. We leverage advances in Bayesian inference of ODE's and sparse regression to identify relevant terms, quantifying the uncertainty in the values of each parameter and its inclusion probability. Further, this method is very robust to noise and can accommodate sparse samples.

Going forward, one of the primary limitations of this method is its scalability to very large libraries of terms. This is primarily due to the computational intensity to sample high dimensional spaces using Markov Chain Monte Carlo. One remedy for this is to use variational inference, which matches classes of functions to the posterior distribution. This method has been particuarly effective for high dimensional model, most notably neural networks, with comparable accuracy to sampling methods.

In addition, for this work we are primarily focused on situations in which coordinates which induce a sparse representation is known. However, in general this "effective" set of coordinates may be unknown. Recent work has incorporated SINDy into neural network architectures to simultaneously learn the parsimonous governing equations and associated coordinates [42]. Incorporating UQ-SINDy into these neural network architecture could greatly improve robustness and confidence in these models and the associated forecasts.

Appendix A

A.1 Rank one Update

Here we will derive (2.23) and (2.25). Namely we will show that if $\boldsymbol{X}_2 = \boldsymbol{A}\boldsymbol{X}_1$ then

$$\bar{\boldsymbol{X}}_1^\dagger = \begin{cases} \boldsymbol{X}_1^\dagger \left(\boldsymbol{I} - \frac{\boldsymbol{n}\boldsymbol{n}^\intercal}{\boldsymbol{n}^\intercal\boldsymbol{n}}\right) & \text{if } (\boldsymbol{I} - \boldsymbol{X}_1^\dagger\boldsymbol{X}_1)^\intercal\boldsymbol{1} = \boldsymbol{0} \\ \left(\boldsymbol{I} - \frac{(\boldsymbol{I}-\boldsymbol{X}_1^\dagger\boldsymbol{X}_1)\boldsymbol{1}\boldsymbol{1}^\intercal}{\boldsymbol{1}^\intercal(\boldsymbol{I}-\boldsymbol{X}_1^\dagger\boldsymbol{X}_1)\boldsymbol{1}}\right)\boldsymbol{X}_1^\dagger & \text{otherwise} \end{cases} \tag{A.1}$$

where $\boldsymbol{n} = \boldsymbol{X}_1^{\dagger^\intercal}\boldsymbol{1}$. To derive this we use the rank-one update formula (3.2.7) from [157] to compute $\bar{\boldsymbol{X}}_1^\dagger = (\boldsymbol{X}_1 - \boldsymbol{\mu}_1\boldsymbol{1}^\intercal)^\dagger$.

First, let's assume $\left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger\boldsymbol{X}_1\right)\boldsymbol{1} = \boldsymbol{0}$. Letting $A = \boldsymbol{X}_1$, $c = -\boldsymbol{\mu}_1$, and $d = \boldsymbol{1}$, then

$$\begin{gathered} \beta = 1 - \boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger\boldsymbol{\mu}_1 = 1 - \frac{\boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger\boldsymbol{X}_1\boldsymbol{1}}{\boldsymbol{1}^\intercal\boldsymbol{1}} = 0 \\ \boldsymbol{w} = -(\boldsymbol{I} - \boldsymbol{X}_1\boldsymbol{X}_1^\dagger)\frac{\boldsymbol{X}_1\boldsymbol{1}}{\boldsymbol{1}^\intercal\boldsymbol{1}} = \boldsymbol{0} \\ \boldsymbol{m} = (\boldsymbol{I} - \boldsymbol{X}_1^\dagger\boldsymbol{X}_1)^\intercal\boldsymbol{1} = \left(\boldsymbol{1}^\intercal - \boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger\boldsymbol{X}_1\right)^\intercal = \boldsymbol{0} \\ \boldsymbol{v} = -\boldsymbol{X}_1^\dagger\boldsymbol{\mu}_1 = -\boldsymbol{X}_1^\dagger\frac{\boldsymbol{X}_1\boldsymbol{1}}{\boldsymbol{1}^\intercal\boldsymbol{1}} = -\frac{\boldsymbol{1}}{\boldsymbol{1}^\intercal\boldsymbol{1}} \\ \boldsymbol{n} = \boldsymbol{X}_1^{\dagger^\intercal}\boldsymbol{1} \end{gathered}$$

Note that $\|\boldsymbol{v}\|^2 = \frac{1}{\boldsymbol{1}^\intercal\boldsymbol{1}}$. Since $\beta = \|\boldsymbol{m}\| = \|\boldsymbol{w}\| = 0$, we are in Case 6 and the pseudoinverse is given by

$$\bar{\boldsymbol{X}}_1^\dagger = \boldsymbol{X}_1^\dagger - \frac{1}{\|\boldsymbol{v}\|^2}\boldsymbol{v}\boldsymbol{v}^\intercal\boldsymbol{X}_1^\dagger - \frac{1}{\|\boldsymbol{n}\|^2}\boldsymbol{X}_1^\dagger\boldsymbol{n}\boldsymbol{n}^\intercal + \frac{\boldsymbol{v}^\intercal\boldsymbol{X}_1^\dagger\boldsymbol{n}}{\|\boldsymbol{v}\|^2\|\boldsymbol{n}\|^2}\boldsymbol{v}\boldsymbol{n}^\intercal$$

Noting that in first term $\frac{\boldsymbol{v}^\intercal\boldsymbol{X}_1^\dagger}{\|\boldsymbol{v}\|^2} = \boldsymbol{n}^\intercal$ and in the third term $\boldsymbol{v}^\intercal\boldsymbol{X}_1^\dagger\boldsymbol{n} = \frac{\boldsymbol{1}^\intercal\boldsymbol{X}_1^\dagger\boldsymbol{X}_1^{\dagger^\intercal}\boldsymbol{1}}{\boldsymbol{1}^\intercal\boldsymbol{1}} = \|\boldsymbol{n}\|^2\|\boldsymbol{v}\|^2$, then the first and third terms equal $-\boldsymbol{v}\boldsymbol{n}^\intercal$ and $\boldsymbol{v}\boldsymbol{n}^\intercal$. These cancel, yielding the first case of (A.1). Thus, by Theorem 2.1 in [52], $\hat{\boldsymbol{A}}$ and $\bar{\boldsymbol{A}}$ share all the same eigenvalues and eigenvectors except the eigenvalue of $\hat{\boldsymbol{A}}$ equal to 1 which becomes

$$1 - \frac{\boldsymbol{n}^\intercal \boldsymbol{X}_2 \boldsymbol{X}_1^\dagger \boldsymbol{n}}{\|\boldsymbol{n}\|^2} = 1 - \frac{\boldsymbol{n}^\intercal \hat{\boldsymbol{A}} \boldsymbol{n}}{\|\boldsymbol{n}\|^2} = 1 - \frac{\boldsymbol{n}^\intercal \boldsymbol{n}}{\|\boldsymbol{n}\|^2} = 0.$$

Now, let's assume $\boldsymbol{1}^\intercal \left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger \boldsymbol{X}_1\right) \neq \boldsymbol{0}$. This corresponds to Case 3 in [157].

$$\begin{aligned}
\boldsymbol{w} &= -\left(\boldsymbol{I} - \boldsymbol{X}_1 \boldsymbol{X}_1^\dagger\right) \frac{\boldsymbol{X}_1 \boldsymbol{1}}{\boldsymbol{1}^\intercal \boldsymbol{1}} = \boldsymbol{0} \\
\boldsymbol{m} &= \left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger \boldsymbol{X}_1\right)^\intercal \boldsymbol{1} \neq \boldsymbol{0} \\
\beta &= 1 - \frac{\boldsymbol{1}^\intercal \boldsymbol{X}_1^\dagger \boldsymbol{X}_1 \boldsymbol{1}}{\boldsymbol{1}^\intercal \boldsymbol{1}} = \frac{\|\boldsymbol{m}\|^2}{\boldsymbol{1}^\intercal \boldsymbol{1}} \neq 0 \\
\boldsymbol{v} &= -\frac{\boldsymbol{X}_1^\dagger \boldsymbol{X}_1 \boldsymbol{1}}{\boldsymbol{1}^\intercal \boldsymbol{1}} \\
\boldsymbol{n} &= \boldsymbol{X}_1^{\dagger\intercal} \boldsymbol{1}
\end{aligned}$$

$$\bar{\boldsymbol{X}}_1^\dagger = \boldsymbol{X}_1^\dagger + \frac{1}{\beta} \boldsymbol{m} \boldsymbol{v}^\intercal \boldsymbol{X}_1^\dagger - \frac{\beta}{\|\boldsymbol{v}\|^2 \|\boldsymbol{m}\|^2 + |\beta|^2} \left(\frac{\|\boldsymbol{v}\|^2}{\beta} \boldsymbol{m} + \boldsymbol{v}\right) \left(\frac{\|\boldsymbol{m}\|^2}{\beta} \left(\boldsymbol{X}_1^\dagger\right)^\intercal \boldsymbol{v} + \boldsymbol{n}\right)^\intercal$$

Now

$$\frac{\|\boldsymbol{m}\|^2}{\beta} \left(\boldsymbol{X}_1^\dagger\right)^\intercal \boldsymbol{v} + \boldsymbol{n} = -\boldsymbol{1}^\intercal \boldsymbol{1} \boldsymbol{X}_1^{\dagger\intercal} \boldsymbol{X}_1^\dagger \frac{\boldsymbol{X}_1 \boldsymbol{1}}{\boldsymbol{1}^\intercal \boldsymbol{1}} + \boldsymbol{X}_1^{\dagger\intercal} \boldsymbol{1} = \boldsymbol{0}$$

since $\boldsymbol{X}_1^\dagger \boldsymbol{X}_1$ is symmetric. Hence,

$$\begin{aligned}
\bar{\boldsymbol{X}}_1^\dagger &= \boldsymbol{X}_1^\dagger + \frac{1}{\beta} \boldsymbol{m} \boldsymbol{v}^\intercal \boldsymbol{X}_1^\dagger \\
&= \left(\boldsymbol{I} - \frac{\left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger \boldsymbol{X}_1\right) \boldsymbol{1}\boldsymbol{1}^\intercal}{\boldsymbol{1}^\intercal \left(\boldsymbol{I} - \boldsymbol{X}_1^\dagger \boldsymbol{X}_1\right) \boldsymbol{1}}\right) \boldsymbol{X}_1^\dagger.
\end{aligned}$$

A.2 Efficient Computation of Fixed Point

We provide an efficient algorithm for computing a fixed point in the case where range($\boldsymbol{X}_1$) = range($\boldsymbol{X}_2$). In the SVD-based DMD algorithm, we efficiently compute the DMD with centering modes by projecting $\bar{\boldsymbol{A}}$ onto the singular vectors $\boldsymbol{U}_r$ of $\boldsymbol{X}_1$: $\bar{\boldsymbol{A}}_r = \boldsymbol{U}_r^\intercal \bar{\boldsymbol{A}} \boldsymbol{U}_r = \boldsymbol{U}_r^\intercal \boldsymbol{X}_2 \boldsymbol{V}_r \boldsymbol{\Sigma}^{-1}$. The matrix $\bar{\boldsymbol{A}}_r$ satisfies the linear model

$$\tilde{\boldsymbol{x}}_{j+1} = \bar{\boldsymbol{A}}_r \tilde{\boldsymbol{x}}_j,$$

where $\tilde{\boldsymbol{x}}_j = \boldsymbol{U}_r^\intercal \boldsymbol{x}_j$. The means $\tilde{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\mu}}_2$ in the lower-dimensional space are similarly related to $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ by $\tilde{\boldsymbol{\mu}}_i = \boldsymbol{U}_r^\intercal \boldsymbol{\mu}_i$ for $i = 1, 2$.

Consider the solution $\tilde{\boldsymbol{c}}$ to the background mode in the lower dimensional space:

$$\tilde{\boldsymbol{c}} = \left(\boldsymbol{I} - \bar{\boldsymbol{A}}_r\right)^{-1} \tilde{\boldsymbol{\mu}}_2 - \bar{\boldsymbol{A}}_r \tilde{\boldsymbol{\mu}}_{\mathbf{1}}.$$

Like before, $\boldsymbol{I} - \bar{\boldsymbol{A}}_r$ is invertible, since $\bar{\boldsymbol{A}}_r$ and $\bar{\boldsymbol{A}}$ have the same nonzero eigenvalues, and $\bar{\boldsymbol{A}}$ does not have an eigenvalue equal to 1. This computation is done in the lower r dimensional space in contrast to the original n dimensional space, so $\tilde{\boldsymbol{c}}$ can be efficiently computed by solving an $r \times r$ linear system. We will show that $\boldsymbol{c}$ is related to $\tilde{\boldsymbol{c}}$ by $\boldsymbol{c} = \boldsymbol{U}_r \tilde{\boldsymbol{c}}$.

Multiplying both sides by $\boldsymbol{I} - \bar{\boldsymbol{A}}_r$,

$$\left(\boldsymbol{I} - \bar{\boldsymbol{A}}_r\right) \tilde{\boldsymbol{c}} = \tilde{\boldsymbol{\mu}}_2 - \bar{\boldsymbol{A}}_r \tilde{\boldsymbol{\mu}}_{\mathbf{1}}.$$

Plugging in the relations between $\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{A}}_r$, $\boldsymbol{\mu}_1$ and $\tilde{\boldsymbol{\mu}}_1$, and $\boldsymbol{\mu}_2$ and $\tilde{\boldsymbol{\mu}}_2$,

$$\left(\boldsymbol{I} - \boldsymbol{U}_r^\intercal \bar{\boldsymbol{A}} \boldsymbol{U}_r\right) \tilde{\boldsymbol{c}} = \boldsymbol{U}_r^\intercal \boldsymbol{\mu}_2 - \boldsymbol{U}_r^\intercal \bar{\boldsymbol{A}} \boldsymbol{U}_r \boldsymbol{U}_r^\intercal \boldsymbol{\mu}_1.$$

Note that since $\operatorname{range}(\boldsymbol{X}_1) = \operatorname{range}(\boldsymbol{X}_2)$, then $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \operatorname{range}(\boldsymbol{X}_1)$ and $\operatorname{range}(\bar{\boldsymbol{A}}) \subseteq \operatorname{range}(\boldsymbol{X}_1)$. Hence, $\boldsymbol{U}_r \boldsymbol{U}_r^\intercal \boldsymbol{\mu}_1 = \boldsymbol{\mu}_1$, $\boldsymbol{U}_r \boldsymbol{U}_r^\intercal \boldsymbol{\mu}_2 = \boldsymbol{\mu}_2$, and $\boldsymbol{U}_r \boldsymbol{U}_r^\intercal \bar{\boldsymbol{A}} = \bar{\boldsymbol{A}}$. Multiplying both sides by $\boldsymbol{U}_r$ and using the previous identities

$$\left(\boldsymbol{I} - \bar{\boldsymbol{A}}\right) \boldsymbol{U}_r \tilde{\boldsymbol{c}} = \boldsymbol{\mu}_2 - \bar{\boldsymbol{A}} \boldsymbol{\mu}_1.$$

Since $\boldsymbol{I} - \bar{\boldsymbol{A}}$ is invertible,

$$\boldsymbol{U}_r \tilde{\boldsymbol{c}} = \left(\boldsymbol{I} - \bar{\boldsymbol{A}}\right)^{-1} \left(\boldsymbol{\mu}_2 - \bar{\boldsymbol{A}}\right) \boldsymbol{\mu}_1 = \boldsymbol{c}.$$

Appendix B

B.1 NMP Minimization Implementation

For the NMP method we must solve the minimization problem

$$\min_a \|x(t) - a(\theta(t))\cos(\theta(t))\|_2^2, \text{ where } a(\theta) \in V(\theta). \tag{B.1}$$

In Ref. [93], Shi and Hou solve this by noting that constraining $a(\theta)$ to lie in $V(\theta)$ is equivalent to applying a low-pass filter in θ-space. The corresponding algorithm is shown in

Algorithm 3 Minimization Algorithm for Nonlinear Matching Pursuit (NMP) Method for Periodic Data

```
Input: measured signal $x(t)$ and phase function of IMFs $\theta(t)$.
Output: $a(t), b(t)$
Define the normalized phase function $\bar{\theta}(t) = \frac{\theta(t)-\theta(0)}{\theta(T)-\theta(0)}$
$L_\theta = \frac{\theta(T)-\theta(0)}{2\pi}$
$x(\theta) :=$ Interpolate($x(t), \theta(t)$) {Reexpress $x(t)$ in terms of the $\theta$ coordinate}.
$\hat{x}(\omega) = \mathcal{F}\left(x(\theta)\right)$
$a(\theta) := \mathcal{F}^{-1}\left(\hat{x}(\omega + L_\theta) + \hat{x}(\omega - L_\theta) \cdot \chi_\lambda(\omega/L_\theta)\right)$
$b(\theta) := \mathcal{F}^{-1}\left(i \cdot (\hat{x}(\omega + L_\theta) - \hat{x}(\omega - L_\theta) \cdot \chi_\lambda(\omega/L_\theta))\right)$
$a(t) =$ Interpolate($a(\theta), t$)
$b(t) =$ Interpolate($b(\theta), t$)
return $a, b$
```

In this algorithm $\mathcal{F}$, and $\mathcal{F}^{-1}$ denote the Fourier Transform, and inverse Fourier Transform respectively

$$\mathcal{F}(r) = \frac{1}{N}\sum_{j=1}^{M} r_j e^{-i2\pi\omega\bar{\theta}_j}, \omega = -N/2+1, \cdots, N/2$$

$$\mathcal{F}^{-1}(\hat{r}) = \frac{1}{N}\sum_{\omega=-N/2+1}^{N/2} \hat{r} e^{i2\pi\omega\bar{\theta}^n_{k,j}}, j = 0, \ldots, N-1.$$

$\chi_\lambda(\omega)$ is the cutoff function used in the low-pass filter. Here we use the function

$$\chi_\lambda(\omega) = \begin{cases} 1 + \cos(\pi\omega/\lambda) & -\lambda < \omega < \lambda \\ 0 & \text{otherwise} \end{cases}.$$

Appendix C

C.1 Discrete Orthogonal Polynomials

In Section 4.3.3 we introduced a set of orthogonal polynomials that appear in HAVOK, and listed these polynomials in the continuous case. The first five polynomials in discrete case are listed below.

$$p_1(n) = \frac{n}{c_1}$$
$$p_2(n) = \frac{n^2}{c_2}$$
$$p_3(n) = \frac{1}{c_3}\left(n^3 - \frac{n\,(3\,p^2+3\,p-1)}{5}\right)$$
$$p_4(n) = \frac{1}{c_4}\left(n^4 - \frac{5\,n^2\,(3\,p^4+6\,p^3-3\,p+1)}{7\,(3\,p^2+3\,p-1)}\right)$$
$$p_5(n) = \frac{1}{c_5}\left(\frac{5\left(\frac{n\left(3\,p^2+3\,p-1\right)}{5}-n^3\right)(2\,p^2+2\,p-3)}{9} - \frac{n\,(3\,p^4+6\,p^3-3\,p+1)}{7} + n^5\right)$$

$$c_1 = \sqrt{\frac{p\,(2\,p+1)\,(p+1)}{3}}$$
$$c_2 = \sqrt{\frac{p\,(2\,p+1)\,(p+1)\,(3\,p^2+3\,p-1)}{15}}$$
$$c_3 = \sqrt{\frac{p\,(2\,p-1)\,(2\,p+1)\,(2\,p+3)\,(p-1)\,(p+1)\,(p+2)}{175}}$$
$$c_4 = \sqrt{\frac{p\,(2\,p-1)\,(2\,p+1)\,(2\,p+3)\,(p-1)\,(p+1)\,(p+2)\,(15\,p^4+30\,p^3-35\,p^2-50\,p+12)}{2205\,(3\,p^2+3\,p-1)}}$$
$$c_5 = \sqrt{\frac{4\,p\,(2\,p-1)\,(2\,p+1)\,(2\,p-3)\,(2\,p+3)\,(2\,p+5)\,(p-1)\,(p+1)\,(p-2)\,(p+2)\,(p+3)}{43659}}$$

C.2 Column Rule for Synthetic Example

In Section 4.4, we state that when applying HAVOK to the synthetic example in 5.3.2 in the limit as the number of columns n in the Hankel matrix $\boldsymbol{H}$ goes to infinity, the derivatives in (4.12) converges to fixed values. Here we prove that the first ratio in the series $\frac{2||\boldsymbol{h}_0'||}{||\boldsymbol{h}_0''||}$ approaches a constant as $n \to \infty$. Further terms in the sequence, can be shown to have the same behavior using a similar proof.

We start with the system $x(t) = \sin(t) + \sin(2t)$. The central row of the matrix $\boldsymbol{h}_0$ will be of the form $x(t)$ for some $a, b \in \mathbb{Z}$ such that

$$t = \begin{bmatrix} a\Delta t & (a+1)\Delta t & (a+2)\Delta t & \dots & b\Delta t \end{bmatrix}.$$

In particular $b = n + a$. Thus, showing that the limit as $n \to \infty$ is equivalent to the limit as $b \to \infty$.

$$\begin{aligned}
\frac{||\boldsymbol{h}_0''||}{||\boldsymbol{h}_0'||} &= \frac{||-\sin(t) - 4\sin(2t)||}{||\cos(t) + 2\cos(2t)||} \\
&= \frac{||\begin{bmatrix} -\sin(a\Delta t) - 4\sin(2a\Delta t) & \dots & -\sin(b\Delta t) - 4\sin(2b\Delta t) \end{bmatrix}||}{||\begin{bmatrix} \cos(a\Delta t) + 2\cos(2a\Delta t) & \dots & \cos(b\Delta t) + 2\cos(2b\Delta t) \end{bmatrix}||} \\
&= \sqrt{\frac{\sum_{k=a}^{b}(\sin(k\Delta t) + 4\sin(2k\Delta t))^2}{\sum_{k=a}^{b}(\cos(k\Delta t) + 2\cos(2k\Delta t))^2}} \\
&= \sqrt{\frac{\sum_{k=a}^{b}(\sin^2(k\Delta t) + 8\sin(k\Delta t)\sin(2k\Delta t) + 16\sin^2(2k\Delta t))}{\sum_{k=a}^{b}(\cos^2(k\Delta t) + 4\cos(k\Delta t)\cos(2k\Delta t) + 4\cos^2(2k\Delta t))}} \\
&= \sqrt{\frac{\sum_{k=a}^{b}(\frac{17}{2} + 4\cos(k\Delta t) - \frac{1}{2}\cos(2k\Delta t) - 4\cos(3k\Delta t) - 8\cos(4k\Delta t))}{\sum_{k=a}^{b}(\frac{5}{2} + 2\cos(k\Delta t) + \frac{1}{2}\cos(2k\Delta t) + 2\cos(3k\Delta t) + 2\cos(4k\Delta t))}}
\end{aligned}$$

In the last step we have used the trigonometric identities $\sin^2(a) = \frac{1}{2}(1 - \cos(2a))$, and $\cos^2(a) = \frac{1}{2}(1 + \cos(2a))$.

Using [120], we have the identity

$$\sum_{k=0}^{q} \cos(Bk) = \frac{\sin\left[(\frac{q+1}{2})B\right]\cos\left[(\frac{q}{2})B\right]}{\sin(\frac{B}{2})}, \quad B, q \in \mathbb{R}.$$

$$\sum_{k=a}^{b}\cos(Bk) = \frac{\sin\left[(\frac{b+1}{2})B\right]\cos\left[(\frac{b}{2})B\right] - \sin\left[(\frac{a}{2})B\right]\cos\left[(\frac{a-1}{2})B\right]}{\sin(\frac{B}{2})}, \quad B, a, b \in \mathbb{R}.$$

Defining $g(b)$ and $h(b)$ as the numerator and denominator under the radical,

$$\begin{aligned} g(b) &= \sum_{k=a}^{b}(4\cos(k\Delta t) - \frac{1}{2}\cos(2k\Delta t) - 4\cos(3k\Delta t) - 8\cos(4k\Delta t)) \\ &= \frac{4(\sin[(b+1)(\frac{\Delta t}{2})]\cos[b(\frac{\Delta t}{2})] - \sin[a(\frac{\Delta t}{2})]\cos[(a-1)(\frac{\Delta t}{2})])}{\sin[\frac{\Delta t}{2}]} \\ &- \frac{\sin[(b+1)\Delta t]\cos[b\Delta t] - \sin[a\Delta t]\cos[(a-1)\Delta t]}{2\sin[\Delta t]} \\ &- \frac{4(\sin[(b+1)(\frac{3\Delta t}{2})]\cos[b(\frac{3\Delta t}{2})] - \sin[a(\frac{3\Delta t}{2})]\cos[(a-1)(\frac{3\Delta t}{2})])}{\sin[\frac{3\Delta t}{2}]} \\ &- \frac{8(\sin[(b+1)(2\Delta t)]\cos[b(2\Delta t)] - \sin[a(2\Delta t)]\cos[(a-1)(2\Delta t)])}{\sin[2\Delta t]} \end{aligned}$$

$$\begin{aligned} h(b) &= \sum_{k=a}^{b}(2\cos(k\Delta t) + \frac{1}{2}\cos(2k\Delta t) + 2\cos(3k\Delta t) + 2\cos(4k\Delta t)) \\ &= \frac{2(\sin[(b+1)(\frac{\Delta t}{2})]\cos[b(\frac{\Delta t}{2})] - \sin[a(\frac{\Delta t}{2})]\cos[(a-1)(\frac{\Delta t}{2})])}{\sin[\frac{\Delta t}{2}]} \\ &+ \frac{\sin[(b+1)\Delta t]\cos[b\Delta t] - \sin[a\Delta t]\cos[(a-1)\Delta t]}{2\sin[\Delta t]} \\ &+ \frac{2(\sin[(b+1)(\frac{3\Delta t}{2})]\cos[b(\frac{3\Delta t}{2})] - \sin[a(\frac{3\Delta t}{2})]\cos[(a-1)(\frac{3\Delta t}{2})])}{\sin[\frac{3\Delta t}{2}]} \\ &+ \frac{2(\sin[(b+1)(2\Delta t)]\cos[b(2\Delta t)] - \sin[a(2\Delta t)]\cos[(a-1)(2\Delta t)])}{\sin[2\Delta t]} \end{aligned}$$

Note that we have the following:

$$\lim_{b\to\infty}\frac{g(b)}{b} = 0 \quad \text{and} \quad \lim_{b\to\infty}\frac{h(b)}{b} = 0$$

. Using this fact, then

$$\begin{aligned}
\lim_{b\to\infty}\frac{2||\boldsymbol{h}_0''||}{||\boldsymbol{h}_0'||} &= 2\lim_{b\to\infty}\sqrt{\frac{\sum_{k=a}^{b}\frac{17}{2}+\sum_{k=a}^{b}(4\cos(k\Delta t)-\frac{1}{2}\cos(2k\Delta t)-4\cos(3k\Delta t)-8\cos(4k\Delta t))}{\sum_{k=a}^{b}\frac{5}{2}+\sum_{k=a}^{b}(2\cos(k\Delta t)+\frac{1}{2}\cos(2k\Delta t)+2\cos(3k\Delta t)+2\cos(4k\Delta t))}}\\
&= 2\lim_{b\to\infty}\sqrt{\frac{\frac{17}{2}(b-a+1)+g(b)}{\frac{5}{2}(b-a+1)+h(b)}}\\
&= 2\lim_{b\to\infty}\sqrt{\frac{\frac{17}{2}-\frac{17a}{2b}+\frac{17}{2b}+\frac{g(b)}{b}}{\frac{5}{2}-\frac{5a}{2b}+\frac{5}{2b}+\frac{h(b)}{b}}}\\
&= 2\sqrt{\frac{17}{5}}.
\end{aligned}$$

C.3 *Structured HAVOK (sHAVOK) algorithm*

Here we present pseudocode for the sHAVOK algorithms with and without forcing terms.

Algorithm 4 Structured HAVOK (sHAVOK) without forcing

```
  Input: Measured signal $x(t)$, number of delays $m$, and rank of Hankel Matrix $\boldsymbol{r}$.
  Output: Dynamics matrix $\hat{\boldsymbol{A}} \in \mathbb{R}^{r\times r}$.
  $\boldsymbol{H} := \text{Hankel}(x(t), m)$
  $\boldsymbol{H}_1 := \boldsymbol{H}[:, 1 : n-1]$
  $\boldsymbol{H}_2 := \boldsymbol{H}[:, 2 : n]$
  $\boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^\intercal := \text{SVD}(\boldsymbol{H}_1, r)$
  $\boldsymbol{U}_2\boldsymbol{\Sigma}_2\boldsymbol{V}_2^\intercal := \text{SVD}(\boldsymbol{H}_2, r)$
  $\hat{\boldsymbol{A}} := \boldsymbol{V}_2^\intercal\boldsymbol{V}_1$
```

Algorithm 5 Structured HAVOK (sHAVOK) with forcing

```
Input: Measured signal $x(t)$, number of delays $m$, and rank of Hankel Matrix $\boldsymbol{r}$.
Output: Dynamics matrix $\hat{\boldsymbol{A}} \in \mathbb{R}^{r-1 \times r-1}$ and forcing term $\hat{\boldsymbol{B}} \in \mathbb{R}^{r-1}$.
$\boldsymbol{H} := \text{Hankel}(x(t), m)$
$\boldsymbol{H}_1 := \boldsymbol{H}[:, 1 : n-1]$
$\boldsymbol{H}_2 := \boldsymbol{H}[:, 2 : n]$
$\boldsymbol{U}_1 \boldsymbol{\Sigma}_1 \boldsymbol{V}_1^{\intercal} := \text{SVD}(\boldsymbol{H}_1, r)$
$\boldsymbol{U}_2 \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^{\intercal} := \text{SVD}(\boldsymbol{H}_2, r-1)$
$[\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}] := \boldsymbol{V}_2^{\intercal} \boldsymbol{V}_1$
```

BIBLIOGRAPHY

[1] Benjamin P Abbott, Richard Abbott, TD Abbott, MR Abernathy, Fausto Acernese, Kendall Ackley, Carl Adams, Thomas Adams, Paolo Addesso, RX Adhikari, et al. Observation of gravitational waves from a binary black hole merger. *Physical Review Letters*, 116(6):061102, 2016.

[2] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1948.

[3] Bruce Abramson, John Brown, Ward Edwards, Allan Murphy, and Robert L Winkler. Hailfinder: A bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1):57–71, 1996.

[4] Orly Alter, Patrick O Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.

[5] J Álvarez-Vizoso, Robert Arn, Michael Kirby, Chris Peterson, and Bruce Draper. Geometry of curves in rn from the local singular value decomposition. *Linear Algebra and its Applications*, 571:180–202, 2019.

[6] Charles W Anderson, James N Knight, Tim O'Connor, Michael J Kirby, and Artem Sokolov. Geometric subspace methods and time-delay embedding for eeg artifact removal and classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):142–146, 2006.

[7] Hassan Arbabi and Igor Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM J. Appl. Dyn. Syst.*, 16(4):2096–2126, 2017.

[8] J Arino and P Van Den Driessche. Time delays in epidemic models. In *Delay differential equations and applications*, pages 539–578. Springer, 2006.

[9] Vladimir Igorevich Arnol'd. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013.

[10] Walter Edwin Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Q. Appl. Math.*, 9(1):17–29, 1951.

[11] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[12] Travis Askham and J Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018.

[13] Andrew D Back and Andreas S Weigend. A first application of independent component analysis to extracting structure from stock returns. *International journal of neural systems*, 8(04):473–484, 1997.

[14] Shervin Bagheri. Effects of small noise on the dmd/koopman spectrum. *Bull. Am. Phys. Soc.*, 58(18):H35, 2013.

[15] Ying-Wen Bai, Wen-Yang Chu, Chien-Yu Chen, Yi-Ting Lee, Yi-Ching Tsai, and Cheng-Hung Tsai. Adjustable 60hz noise reduction by a notch filter for ecg signals. In *Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference*, volume 3, pages 1706–1711. IEEE, 2004.

[16] Zhe Bai, Thakshila Wimalajeewa, Zachary Berger, Guannan Wang, Mark Glauser, and Pramod K Varshney. Low-dimensional approach for reconstruction of airfoil data via compressive sensing. *AIAA journal*, 53(4):920–933, 2015.

[17] Bernhard Beckermann and Alex Townsend. Bounds on the singular values of matrices with displacement structure. *SIAM Review*, 61(2):319–344, 2019.

[18] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[19] Gianpaolo Bellini, J Benziger, D Bick, G Bonfini, D Bravo, M Buizza Avanzini, B Caccianiga, L Cadonati, F Calaprice, P Cavalcante, et al. Final results of borexino phase-i on low-energy solar neutrino spectroscopy. *Physical Review D*, 89(11):112007, 2014.

[20] Adel Belouchrani, Karim Abed-Meraim, J-F Cardoso, and Eric Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, 45(2):434–444, 1997.

[21] Nicky Best, Sylvia Richardson, and Andrew Thomson. A comparison of bayesian spatial models for disease mapping. *Statistical methods in medical research*, 14(1):35–59, 2005.

[22] Anindya Bhadra, Jyotishka Datta, Nicholas G Polson, Brandon Willard, et al. The horseshoe+ estimator of ultra-sparse signals. *Bayesian Analysis*, 12(4):1105–1131, 2017.

[23] Anindya Bhadra, Jyotishka Datta, Nicholas G Polson, Brandon Willard, et al. Lasso meets horseshoe: A survey. *Statistical Science*, 34(3):405–427, 2019.

[24] Anirban Bhattacharya, Debdeep Pati, Natesh S Pillai, and David B Dunson. Dirichlet–laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, 110(512):1479–1490, 2015.

[25] Donato Bini, Fernando de Felice, and Robert T Jantzen. Absolute and relative frenet-serret frames and fermi-walker transport. *Classical and Quantum Gravity*, 16(6):2105, 1999.

[26] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[27] Enrico Bozzo, Roberto Carniel, and Dario Fasino. Relationship between singular spectrum analysis and fourier analysis: Theory and application to the monitoring of volcanic activity. *Computers & Mathematics with Applications*, 60(3):812–820, 2010.

[28] David S Broomhead and Roger Jones. Time-series analysis. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 423(1864):103–121, 1989.

[29] Bingni W Brunton, Lise A Johnson, Jeffrey G Ojemann, and J Nathan Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of neuroscience methods*, 258:1–15, 2016.

[30] 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):19, 2017.

[31] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[32] Steven L Brunton, Jonathan H Tu, Ido Bright, and J Nathan Kutz. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1716–1732, 2014.

[33] Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.

[34] Philip G Burke and Harry M Schey. Elastic scattering of low-energy electrons by atomic hydrogen. *Physical Review*, 126(1):147, 1962.

[35] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):11, 2011.

[36] Jean-François Cardoso and Antoine Souloumiac. Blind beamforming for non-gaussian signals. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 362–370. IET, 1993.

[37] Bob Carpenter. Predator-prey population dynamics: The lotka-volterra model in stan. *Pridobljeno s https://mc-stan. org/users/documentation/case-studies/lotka-volterra-predator-prey. html [28. 8. 2019]*, 62, 2018.

[38] Carlos M Carvalho, Nicholas G Polson, and James G Scott. Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics*, pages 73–80, 2009.

[39] Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

[40] Enrique Castillo, José María Menéndez, and Santos Sánchez-Cambronero. Predicting traffic flow using bayesian networks. *Transportation Research Part B: Methodological*, 42(5):482–509, 2008.

[41] Ismaël Castillo, Johannes Schmidt-Hieber, Aad Van der Vaart, et al. Bayesian linear regression with sparse priors. *The Annals of Statistics*, 43(5):1986–2018, 2015.

[42] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[43] Kevin K Chen, Jonathan H Tu, and Clarence W Rowley. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *J. Nonlinear Sci.*, 22(6):887–915, 2012.

[44] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[45] C Colijn and ER Vrscay. Spin-dependent bohm trajectories for hydrogen eigenstates. *Physics Letters A*, 300(4-5):334–340, 2002.

[46] Julián Colorado, Antonio Barrientos, Alexander Martinez, Benjamin Lafaverges, and João Valente. Mini-quadrotor attitude control based on hybrid backstepping & frenet-serret theory. In *2010 IEEE International Conference on Robotics and Automation*, pages 1617–1622. IEEE, 2010.

[47] 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.

[48] Ingrid Daubechies. *Ten lectures on wavelets*, volume 61. SIAm, 1992.

[49] Ingrid Daubechies, Jianfeng Lu, and Hau-Tieng Wu. Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Applied and Computational Harmonic Analysis*, 30(2):243–261, 2011.

[50] Scott TM Dawson, Maziar S Hemati, Matthew O Williams, and Clarence W Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Exp. Fluids*, 57(3):42, 2016.

[51] Pierre S De Laplace. Mémoire sur la probabilité des causes par les événements. *Mém. de math. et phys. présentés à l'Acad. roy. des sci*, 6:621–656, 1774.

[52] Jiu Ding and Aihui Zhou. Eigenvalues of rank-one updated matrices with some applications. *Appl. Math. Lett.*, 20(12):1223–1226, 2007.

[53] Manfredo P Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition.* Courier Dover Publications, 2016.

[54] Michael F Doherty and Julio M Ottino. Chaos in deterministic systems: strange attractors, turbulence, and applications in chemical engineering. *Chem. Eng. Sci.*, 43(2):139–183, 1988.

[55] Konstantin Dragomiretskiy and Dominique Zosso. Variational mode decomposition. *IEEE transactions on signal processing*, 62(3):531–544, 2013.

[56] Daniel Dylewsky, Eurika Kaiser, Steven L Brunton, and J Nathan Kutz. Principal component trajectories (pct): Nonlinear dynamics as a superposition of time-delayed periodic orbits. *arXiv preprint arXiv:2005.14321*, 2020.

[57] James B Elsner and Thomas H Jagger. A hierarchical bayesian approach to seasonal hurricane modeling. *Journal of Climate*, 17(14):2813–2827, 2004.

[58] N Benjamin Erichson, Steven L Brunton, and J Nathan Kutz. Compressed dynamic mode decomposition for background modeling. *Journal of Real-Time Image Processing*, 16(5):1479–1492, 2019.

[59] Jiaxin Feng, Zhaohua Wu, and Guosheng Liu. Fast multidimensional ensemble empirical mode decomposition using a data compression technique. *Journal of Climate*, 27(10):3492–3504, 2014.

[60] Harold C Fritts, Terence J Blasing, Bruce P Hayden, and John E Kutzbach. Multivariate techniques for specifying tree-growth and climate relationships and for reconstructing anomalies in paleoclimate. *J. Appl. Meteorol. Climatol.*, 10(5):845–864, 1971.

[61] Nicholas Galioto and Alex Gorodetsky. Bayesian system id: Optimal management of parameter, model, and measurement uncertainty. *arXiv preprint arXiv:2003.02359*, 2020.

[62] Matan Gavish and David L Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE T. Inform. Theory*, 60(8):5040–5053, 2014.

[63] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[64] Andrew Gelman, Xiao-Li Meng, and Hal Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica sinica*, pages 733–760, 1996.

[65] Richard H Gerlach, Cathy WS Chen, and Nancy YC Chan. Bayesian time-varying quantile forecasting for value-at-risk in financial markets. *Journal of Business & Economic Statistics*, 29(4):481–492, 2011.

[66] Zoubin Ghahramani. Unsupervised learning. In *Advanced lectures on machine learning*, pages 72–112. Springer, 2004.

[67] Dimitrios Giannakis and Andrew J Majda. Nonlinear laplacian spectral analysis for time series with intermittency and low-frequency variability. *Proceedings of the National Academy of Sciences*, 109(7):2222–2227, 2012.

[68] John F Gibson, J Doyne Farmer, Martin Casdagli, and Stephen Eubank. An analytic approach to practical state space reconstruction. *Physica D: Nonlinear Phenomena*, 57(1-2):1–30, 1992.

[69] Jerome Gilles. Empirical wavelet transform. *IEEE transactions on signal processing*, 61(16):3999–4010, 2013.

[70] Ryan Givens, OF de Alcantara Bonfim, and Robert B Ormond. Direct observation of normal modes in coupled oscillators. *American Journal of Physics*, 71(1):87–90, 2003.

[71] Narendra S Goel, Samaresh C Maitra, and Elliott W Montroll. On the volterra and other nonlinear models of interacting populations. *Reviews of modern physics*, 43(2):231, 1971.

[72] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.

[73] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer, 1971.

[74] Michael M Goodwin. Multichannel matching pursuit and applications to spatial audio coding. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pages 1114–1118. IEEE, 2006.

[75] Richard M Goodwin. A growth cycle. In *Essays in economic dynamics*, pages 165–170. Springer, 1982.

[76] David J Griffiths. *Introduction to electrodynamics.* Prentice Hall New Jersey, 1962.

[77] Jacob Grosek and J Nathan Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.

[78] Andres D Grosmark and György Buzsáki. Diversity in neural firing dynamics supports both rigid and learned hippocampal sequences. *Science*, 351(6280):1440–1443, 2016.

[79] Long J. Grosmark, A.D. and G. Buzsáki. Recordings from hippocampal area ca1, pre, during and post novel spatial learning. CRCNS.org. http://dx.doi.org/10.6080/K0862DC5, 2016.

[80] Eugene Gutkin. Curvatures, volumes and norms of derivatives for curves in riemannian manifolds. *Journal of Geometry and Physics*, 61(11):2147–2161, 2011.

[81] Fredric J Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *P. IEEE*, 66(1):51–83, 1978.

[82] Robert E Hart and Richard H Grumm. Using normalized climatological anomalies to rank synoptic-scale events objectively. *Mon. Weather Rev.*, 129(9):2426–2442, 2001.

[83] Jun He, Laura Balzano, and Arthur Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1568–1575. IEEE, 2012.

[84] Maziar S Hemati, Clarence W Rowley, Eric A Deem, and Louis N Cattafesta. De-biasing the dynamic mode decomposition for applied koopman spectral analysis of noisy datasets. *Theor. Comp. Fluid Dyn.*, 31(4):349–368, 2017.

[85] Neyram Hemati. Strange attractors in brushless dc motors. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 41(1):40–45, 1994.

[86] Herbert W Hethcote, Harlan W Stech, and Pauline Van Den Driessche. Nonlinear oscillations in epidemic models. *SIAM Journal on Applied Mathematics*, 40(1):1–9, 1981.

[87] Charles Gordon Hewitt. *The conservation of the wild life of Canada.* New York: C. Scribner, 1921.

[88] Seth M Hirsh, Kameron Decker Harris, J Nathan Kutz, and Bingni W Brunton. Centering data improves the dynamic mode decomposition. *arXiv preprint arXiv:1906.05973*, 2019.

[89] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[90] Moritz Hoffmann, Christoph Fröhner, and Frank Noé. Reactive sindy: Discovering governing reactions from concentration data. *The Journal of Chemical Physics*, 150(2):025101, 2019.

[91] Jonathan Horrocks and Chris T Bauch. Algorithmic discovery of dynamic models from infectious disease data. *Scientific Reports*, 10(1):1–18, 2020.

[92] Thomas Y Hou and Zuoqiang Shi. Adaptive data analysis via sparse time-frequency representation. *Advances in Adaptive Data Analysis*, 3(01n02):1–28, 2011.

[93] Thomas Y Hou and Zuoqiang Shi. Data-driven time-frequency analysis. *Applied and Computational Harmonic Analysis*, 35(2):284–308, 2013.

[94] Thomas Y Hou and Zuoqiang Shi. Extracting a shape function for a signal with intra-wave frequency modulation. *Philosophical Transactions of the Royal Society*, 374(2065):20150194, 2016.

[95] Thomas Y Hou and Zuoqiang Shi. Sparse time-frequency decomposition for multiple signals with same frequencies. *Advances in Data Science and Adaptive Analysis*, 9(04):1750010, 2017.

[96] Thomas Y Hou, Zuoqiang Shi, and Peyman Tavallali. Convergence of a data-driven time-frequency analysis method. *Applied and Computational Harmonic Analysis*, 37(2):235–270, 2014.

[97] Thomas Y Hou, Zuoqiang Shi, and Peyman Tavallali. Sparse time frequency representations and dynamical systems. *Communications in Mathematical Sciences*, 13(3):673–694, 2015.

[98] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*, volume 454, pages 903–995. The Royal Society, 1998.

[99] Norden E Huang, Man-Li Wu, Wendong Qu, Steven R Long, and Samuel SP Shen. Applications of Hilbert-Huang transform to non-stationary financial time series analysis. *Applied Stochastic Models in Business and Industry*, 19(3):245–268, 2003.

[100] Norden E. Huang and Zhaohua Wu. A review on Hilbert-Huang transform: Method and its applications to geophysical studies. *Reviews of Geophysics*, 46(2), Jun 2008.

[101] Norden Eh Huang. *Hilbert-Huang transform and its applications*, volume 16. World Scientific, 2014.

[102] Aapo Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE T. Neural Networ.*, 10(3):626–634, 1999.

[103] Aapo Hyvärinen. Independent component analysis of images. *Encyclopedia of Computational Neuroscience*, pages 1–5, 2013.

[104] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

[105] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[106] Hemant Ishwaran, J Sunil Rao, et al. Spike and slab variable selection: frequentist and bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.

[107] Bala R Iyer and CV Vishveshwara. Frenet-serret description of gyroscopic precession. *Physical Review D*, 48(12):5706, 1993.

[108] Charles Jaffé and Paul Brumer. Local and normal modes: A classical perspective. *The Journal of Chemical Physics*, 73(11):5646–5658, 1980.

[109] IT Joliffe and BJT Morgan. Principal component analysis and exploratory factor analysis. *Statistical methods in medical research*, 1(1):69–95, 1992.

[110] Ian Jolliffe. *Principal component analysis*. Springer, 2011.

[111] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

[112] Mihailo R Jovanović, Peter J Schmid, and Joseph W Nichols. Sparsity-promoting dynamic mode decomposition. *Phys. Fluids*, 26(2):024103, 2014.

[113] MR Jovanovic, PJ Schmid, and JW Nichols. Low-rank and sparse dynamic mode decomposition. *Center for Turbulence Research Annual Research Briefs*, 2012:139–152, 2012.

[114] J-N Juang and Richard S Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of guidance, control, and dynamics*, 8(5):620–627, 1985.

[115] Kadierdan Kaheman, Steven L Brunton, and J Nathan Kutz. Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data. *arXiv preprint arXiv:2009.08810*, 2020.

[116] Kadierdan Kaheman, Eurika Kaiser, Benjamin Strom, J Nathan Kutz, and Steven L Brunton. Learning discrepancy models from experimental data. *arXiv preprint arXiv:1909.08574*, 2019.

[117] Kadierdan Kaheman, J Nathan Kutz, and Steven L Brunton. Sindy-pi: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *arXiv preprint arXiv:2004.02322*, 2020.

[118] Mason Kamb, Eurika Kaiser, Steven L Brunton, and J Nathan Kutz. Time-delay observables for koopman: Theory and applications. *arXiv preprint arXiv:1810.01479*, 2018.

[119] Sharon E Kingsland and Sharon E Kingsland. *Modeling nature*. University of Chicago Press, 1995.

[120] Michael P Knapp. Sines and cosines of angles in arithmetic progression. *Mathematics magazine*, 82(5):371, 2009.

[121] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the national academy of sciences of the united states of america*, 17(5):315, 1931.

[122] J Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

[123] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.

[124] J Nathan Kutz, Xing Fu, and Steven L Brunton. Multiresolution dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.*, 15(2):713–735, 2016.

[125] Jake Nathan Kutz, J Grosek, and Steven L Brunton. Dynamic mode decomposition for robust pca with applications to foreground/background subtraction in video streams and multi-resolution analysis. *CRC Handbook on Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*, 2016.

[126] Cornelius Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office Los Angeles, CA, 1950.

[127] Andrew B Lawson. *Bayesian disease mapping: hierarchical modeling in spatial epidemiology*. CRC press, 2013.

[128] Ju-Won Lee and Gun-Ki Lee. Design of an adaptive filter with a dynamic structure for ecg signal processing. *Int. J. Control Autom.*, 3(1):137–142, 2005.

[129] Te-Won Lee, Mark Girolami, and Terrence J Sejnowski. Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural computation*, 11(2):417–441, 1999.

[130] GA Leonov, NV Kuznetsov, and TN Mokaev. Homoclinic orbits, and self-excited and hidden attractors in a lorenz-like system describing convective fluid motion. *Eur. Phy. J. Spec. Top.*, 224(8):1421–1458, 2015.

[131] Qianxiao Li, Felix Dietrich, Erik M Bollt, and Ioannis G Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103111, 2017.

[132] Wayne P London and James A Yorke. Recurrent outbreaks of measles, chickenpox and mumps: I. seasonal variation in contact rates. *American journal of epidemiology*, 98(6):453–468, 1973.

[133] Edward N Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20(2):130–141, 1963.

[134] Alfred J Lotka. Contribution to the theory of periodic reactions. *The Journal of Physical Chemistry*, 14(3):271–274, 2002.

[135] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018.

[136] Alan Mackey, Hayden Schaeffer, and Stanley Osher. On the compressive spectral method. *Multiscale Modeling & Simulation*, 12(4):1800–1827, 2014.

[137] Lucia Maddalena and Alfredo Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Trans. Image Process.*, 17(7):1168–1177, 2008.

[138] David Madigan and Adrian E Raftery. Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.

[139] Danilo P Mandic, Naveed ur Rehman, Zhaohua Wu, and Norden E Huang. Empirical mode decomposition-based time-frequency analysis of multivariate signals: The power of adaptive data analysis. *IEEE signal processing magazine*, 30(6):74–86, 2013.

[140] Niall M Mangan, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):52–63, 2016.

[141] Jordan Mann and J Nathan Kutz. Dynamic mode decomposition for financial trading strategies. *Quant. Financ.*, 16(11):1643–1655, 2016.

[142] Leonard Meirovitch. *Methods of analytical dynamics.* Courier Corporation, 2010.

[143] Igor Mezić. Analysis of fluid flows via spectral properties of the koopman operator. *Annu. Rev. Fluid Mech.*, 45:357–378, 2013.

[144] Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the american statistical association*, 83(404):1023–1032, 1988.

[145] Neil Muller, Lourenço Magaia, and Ben M Herbst. Singular value decomposition, eigenfaces, and 3d reconstructions. *SIAM review*, 46(3):518–545, 2004.

[146] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[147] David P Nguyen, Riccardo Barbieri, Matthew A Wilson, and Emery N Brown. Instantaneous frequency and amplitude modulation of eeg in the hippocampus reveals state dependent temporal structure. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 1711–1715. IEEE, 2008.

[148] Robert K Niven, Ali Mohammad-Djafari, Laurent Cordier, Markus Abel, and Markus Quade. Bayesian identification of dynamical systems. *Multidisciplinary Digital Publishing Institute Proceedings*, 33(1):33, 2020.

[149] Bernd R Noack, Konstantin Afanasiev, MAREK MORZYŃSKI, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *J. Fluid Mech.*, 497:335–363, 2003.

[150] Barrett O'neill. *Elementary differential geometry.* Academic press, 2014.

[151] Samuel E Otto and Clarence W Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

[152] Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, and Stanley Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.

[153] Vincent Pagneux, N Amir, and Jean Kergomard. A study of wave propagation in varying cross-section waveguides by modal decomposition. part i. theory and validation. *The Journal of the Acoustical Society of America*, 100(4):2034–2048, 1996.

[154] Trevor Park and George Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[155] Jonathan R Partington, James Riddick Partington, et al. *An introduction to Hankel operators*, volume 13. Cambridge University Press, 1988.

[156] Roger Penrose. A generalized inverse for matrices. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 51, pages 406–413. Cambridge University Press, 1955.

[157] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook, 2008.

[158] Andrea Pigorini, Adenauer G Casali, Silvia Casarotto, Fabio Ferrarelli, Giuseppe Baselli, Maurizio Mariotti, Marcello Massimini, and Mario Rosanova. Time-frequency spectral analysis of tms-evoked eeg oscillations by means of Hilbert-Huang transform. *Journal of neuroscience methods*, 198(2):236–245, 2011.

[159] Juho Piironen, Aki Vehtari, et al. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018–5051, 2017.

[160] Marion Pilté, Silvère Bonnabel, and Frédéric Barbaresco. Tracking the frenet-serret frame associated to a highly maneuvering target in 3d. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1969–1974. IEEE, 2017.

[161] Douglas Poland. Cooperative catalysis and chemical chaos: a chemical model for the lorenz equations. *Physica D: Nonlinear Phenomena*, 65(1-2):86–99, 1993.

[162] Joshua L Proctor, Steven L Brunton, Bingni W Brunton, and JN Kutz. Exploiting sparsity and equation-free architectures in complex systems. *The European Physical Journal Special Topics*, 223(13):2665–2684, 2014.

[163] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM J. Appl. Dyn. Syst.*, 15(1):142–161, 2016.

[164] Joshua L Proctor and Philip A Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *Int. Health*, 7(2):139–145, 2015.

[165] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.

[166] Reza Ravani and Ali Meghdari. Velocity distribution profile for robot arm motion using rational frenet–serret curves. *Informatica*, 17(1):69–84, 2006.

[167] Gabriel Rilling, Patrick Flandrin, Paulo Goncalves, et al. On empirical mode decomposition and its algorithms. In *IEEE-EURASIP workshop on nonlinear signal and image processing*, volume 3, pages 8–11. NSIP-03, Grado (I), 2003.

[168] Gabriel Rilling, Patrick Flandrin, Paulo Gonçalves, and Jonathan M Lilly. Bivariate empirical mode decomposition. *IEEE signal processing letters*, 14(12):936–939, 2007.

[169] Joelw Robbin and Dietmara Salamon. The exponential vandermonde matrix. *Linear Algebra Appl.*, 317:225–226, 2000.

[170] Reinhardt Mathias Rosenberg. Normal modes of nonlinear dual-mode systems. 1960.

[171] OE Rossler. An equation for hyperchaos. *Physics Letters A*, 71(2-3):155–157, 1979.

[172] Otto E Rössler. An equation for continuous chaos. *Physics Letters A*, 57(5):397–398, 1976.

[173] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.

[174] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18(2):643–660, 2019.

[175] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[176] Jun John Sakurai and Eugene D Commins. Modern quantum mechanics, revised edition, 1995.

[177] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

[178] O Santolík, M Parrot, and F Lefeuvre. Singular value decomposition methods for wave propagation analysis. *Radio Science*, 38(1), 2003.

[179] William M Schaffer and Mark Kot. Do strange attractors govern ecological systems? *BioScience*, 35(6):342–350, 1985.

[180] Peter Schmid and Joern Sesterhenn. Dynamic mode decomposition of numerical and experimental data. *APS*, 61:MR–007, 2008.

[181] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[182] Peter J Schmid, Larry Li, Matthew P Juniper, and O Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.

[183] PJ Schmid and J Sesterhenn. Sixty-first annual meeting of the aps division of fluid dynamics. *San Antonio, Texas, USA*, 2008.

[184] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[185] Erwin Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Physical review*, 28(6):1049, 1926.

[186] Radu Serban and Alan C Hindmarsh. Cvodes: An ode solver with sensitivity analysis capabilities. Technical report, Technical Report UCRL-JP-200039, Lawrence Livermore National Laboratory, 2003.

[187] J-A Serret. Sur quelques formules relatives à la théorie des courbes à double courbure. *Journal de mathématiques pures et appliquées*, pages 193–207, 1851.

[188] Ofer Shapira, Ayman F Abouraddy, John D Joannopoulos, and Yoel Fink. Complete modal decomposition for optical waveguides. *Physical review letters*, 94(14):143902, 2005.

[189] Steven W Shaw and Christophe Pierre. Normal modes for non-linear vibratory systems. *Journal of sound and vibration*, 164(1):85–124, 1993.

[190] Daniel E Shea, Steven L Brunton, and Nathan Kutz. Sindy-bvp: Sparse identification of nonlinear dynamics for boundary value problems. *arXiv preprint arXiv:2005.10756*, 2020.

[191] Jack Sherman and Winifred J Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Stat.*, 21(1):124–127, 1950.

[192] Xingjian Shi and Dit-Yan Yeung. Machine learning for spatiotemporal sequence forecasting: A survey. *arXiv preprint arXiv:1808.06865*, 2018.

[193] Troy Shinbrot, Celso Grebogi, Jack Wisdom, and James A Yorke. Chaos in a double pendulum. *American Journal of Physics*, 60(6):491–499, 1992.

[194] Andrews Sobral and Antoine Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Comput. Vis. Image Und.*, 122:4–21, 2014.

[195] Mariia Sorokina, Stylianos Sygletos, and Sergei Turitsyn. Sparse identification for nonlinear optical communication systems: Sino method. *Optics express*, 24(26):30433–30443, 2016.

[196] Michael D Spivak. *A comprehensive introduction to differential geometry.* Publish or perish, 1970.

[197] Bruce Stephenson. *Kepler's physical astronomy*, volume 13. Springer Science & Business Media, 2012.

[198] George Sugihara and Robert M May. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, 344(6268):734–741, 1990.

[199] Shiliang Sun, Changshui Zhang, and Guoqiang Yu. A bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, 7(1):124–132, 2006.

[200] Gilead Tadmor, Oliver Lehmann, Bernd R Noack, and Marek Morzyński. Mean field representation of the natural and actuated cylinder wake. *Phys. Fluids*, 22(3):034102, 2010.

[201] Naoya Takeishi, Yoshinobu Kawahara, Yasuo Tabei, and Takehisa Yairi. Bayesian dynamic mode decomposition. In *IJCAI*, pages 2814–2821, 2017.

[202] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pages 1130–1140, 2017.

[203] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[204] John R Taylor et al. *Classical mechanics*. University Science Books, 2005.

[205] T Frede Thingstad. Elements of a theory for the mechanisms controlling abundance, diversity, and biogeochemical role of lytic bacterial viruses in aquatic systems. *Limnology and Oceanography*, 45(6):1320–1328, 2000.

[206] Ying-Li Tian, Max Lu, and Arun Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1182–1187. IEEE, 2005.

[207] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[208] Jonathan L Ticknor. A bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, 40(14):5501–5506, 2013.

[209] Dustin Tran, Alp Kucukelbir, Adji B Dieng, Maja Rudolph, Dawen Liang, and David M Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

[210] Giang Tran and Rachel Ward. Exact recovery of chaotic systems from highly corrupted data. *Multiscale Modeling & Simulation*, 15(3):1108–1129, 2017.

[211] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[212] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

[213] Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: theory and applications. *arXiv preprint arXiv:1312.0041*, 2013.

[214] L Richard Turner. Inverse of the vandermonde matrix with applications. 1966.

[215] M Varon and BP Zeigler. Bacterial predator-prey interaction at low prey density. *Applied and Environmental Microbiology*, 36(1):11–17, 1978.

[216] Vito Volterra. *Variazioni e fluttuazioni del numero d'individui in specie animali conviventi.* C. Ferrari, 1927.

[217] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 37(3):328–339, 1989.

[218] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[219] Nancy XR Wang, Ali Farhadi, Rajesh PN Rao, and Bingni W Brunton. Ajile movement prediction: Multimodal deep learning for natural human neural recordings and video. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[220] Tong Wang, Mingcai Zhang, Qihao Yu, and Huyuan Zhang. Comparing the applications of emd and eemd on time–frequency analysis of seismic signal. *Journal of Applied Geophysics*, 83:29–34, 2012.

[221] Wen-Xu Wang, Rui Yang, Ying-Cheng Lai, Vassilios Kovanis, and Celso Grebogi. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical review letters*, 106(15):154101, 2011.

[222] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: an expanded change detection benchmark dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 387–394, 2014.

[223] Zhen Wang, Xia Huang, and Guodong Shi. Analysis of nonlinear dynamics and chaos in a fractional order financial system with time delay. *Computers & Mathematics with Applications*, 62(3):1531–1539, 2011.

[224] CO Weiss and J Brock. Evidence for lorenz-type chaos in a laser. *Phys. Rev. Lett.*, 57(22):2804, 1986.

[225] Mike West and Jeff Harrison. *Bayesian forecasting and dynamic models.* Springer Science & Business Media, 2006.

[226] Edmund Taylor Whittaker and George Neville Watson. *A course of modern analysis.* Cambridge university press, 1996.

[227] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *J. Nonlinear Sci.*, 25(6):1307–1346, 2015.

[228] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometr. Intell. Lab.*, 2(1-3):37–52, 1987.

[229] Jonathan H Wright. Bayesian model averaging and exchange rate forecasts. *Journal of Econometrics*, 146(2):329–341, 2008.

[230] Zhaohua Wu, Norden E Huang, Steven R Long, and Chung-Kang Peng. On the trend, detrending, and variability of nonlinear and nonstationary time series. *Proceedings of the National Academy of Sciences*, 104(38):14889–14894, 2007.

[231] Shang-jun Yang, Hua-zhang Wu, and Quan-bing Zhang. Generalization of vandermonde determinants. *Linear Algebra Appl.*, 336(1-3):201–204, 2001.

[232] Yibo Yang, Mohamed Aziz Bhouri, and Paris Perdikaris. Bayesian differential programming for robust systems identification under uncertainty. *arXiv preprint arXiv:2004.06843*, 2020.

[233] Rongjie Yu, Mohamed Abdel-Aty, and Mohamed Ahmed. Bayesian random effect models incorporating real-time weather and traffic data to investigate mountainous freeway hazardous factors. *Accident Analysis & Prevention*, 50:371–376, 2013.

[234] JE Yuen and GJPP Hughes. Bayesian analysis of plant disease prediction. *Plant Pathology*, 51(4):407–412, 2002.

[235] Hao Zhang, Clarence W Rowley, Eric A Deem, and Louis N Cattafesta. Online dynamic mode decomposition for time-varying systems. *arXiv preprint arXiv:1707.02876*, 2017.

[236] Yan Zhang, Brian J Reich, and Howard D Bondell. High dimensional linear regression via the r2-d2 shrinkage prior. *arXiv preprint arXiv:1609.00046*, 2016.

[237] Weizhong Zheng, Der-Horng Lee, and Qixin Shi. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of transportation engineering*, 132(2):114–121, 2006.