Modified dynamic mode decomposition (mDMD)

Amani E. Ngowi

Supervisor:
Montaz Ali

A research thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

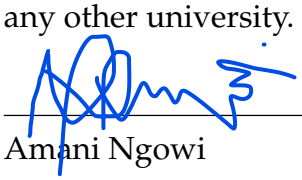
in the

School of Computational Applied Mathematics and Computer Science.
University of the Witwatersrand, Johannesburg.

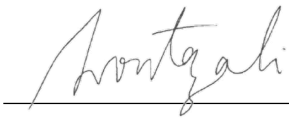
8 November 2024

Declaration

I, Amani E. Ngowi, declare that this proposal is my own, unaided work. It is being submitted for the degree of Master of Science at the University of the Witwatersrand, Johannesburg.. It has not been submitted for any degree or examination at any other university.



Amani Ngowi
Candidate
8 November 2024



Prof. Montaz Ali
Supervisor
8 November 2024

Abstract

Dynamical systems provide valuable insights into natural phenomena, classically modeled by mathematical equations. Recently, the analysis and characterization of dynamical systems is transitioning towards data-driven methods, such as Dynamic Mode Decomposition (DMD). The transition is propelled by the availability of large dynamic datasets from complex systems, often in the absence of explicit governing equations. The infinite dimensional linear Koopman operator offers a global linear approximation of nonlinear dynamics, providing a foundation for DMD-based approximations through finite-dimensional linear operators.

In this thesis, we propose a nonhomogeneous extension to the existing DMD framework by incorporating a regularizer into the traditional homogeneous linear model. The proposed regularization enhances model flexibility, enabling the capture of external influences within dynamical systems. To implement this, we have solved an optimization problem, by minimizing the Frobenius norm of the finite dimensional Koopman operator with the added regularizer, thus finding optimal solutions to the nonhomogeneous linear model. Our proposed approach aims to achieve two primary data driven objectives: identifying coherent patterns within dynamic systems and extending the applicability of DMD to a wider range of nonlinear systems. Furthermore, we have numerically evaluated the influence of regularization within DMD, demonstrating that our modified DMD consistently outperforms the standard DMD methodology in a range of tested scenarios.

Acknowledgements

I would like to thank God. This journey has occurred through His Grace.

My sincere gratitude to my Supervisor, Prof. Montaz Ali. I would love to acknowledge the valuable skills he shared throughout.

I greatly acknowledge the support by the staff members from the school of Computer Science and Applied Mathematics at Wits University.

I deeply thank my family and colleagues for the love and support.

I am sincerely grateful to everyone that played and still play a role in my life ...

Contents

Chapter 1

Introduction

Interactions between quantities of physical (dynamical) systems have been classically (equation based) described by mathematical models. The description of physical systems often involve systems state estimation and prediction. The traditional mathematical models are formulated based on first principles, spanning a wide range of phenomena. However, due to increasing complexities in many modern physical systems, it is not getting easier to formulate mathematical models. Alternatively, data based methods, from taking snapshots, simulations or measurements of the physical system, are increasingly used to construct models. In this research, we have considered a new yet well established data dependent method called the dynamic mode decomposition (DMD) [17]. The DMD method involves taking up high dimensional data, decomposing the data into relevant (tractable) linear dynamic structures, thereby constructing linear dynamic models using data of a potentially nonlinear dynamical system [17] .

DMD constructs a linear dynamic model through the mining of spatiotemporal features, leveraging inherent low dimensionality, in the high dimensional data. Similar methods include, proper orthogonal decomposition (POD) [9], which only concentrates on capturing spatial features. In this research, we have considered high dimensional data of nonlinear dynamical systems (NLDS). NLDS often exhibit a type of complexity that cannot be easily comprehended to facilitate system state estimation and prediction. Our proposed study uses DMD, benefiting from its strong connection to the analysis of NLDS through the linear Koopman operator, appreciating the fruitful work done in [14].

1.1 Research focus

DMD seeks to identify low dimensional coherent structures or patterns (modes) in the high dimensional dynamic data. DMD stands out among its counterparts due to its unique capability to not only identify the spatial patterns or modes within the data but also to capture their temporal evolution. This distinguishing feature enables DMD to furnish dynamic modes through its decomposition process, thereby providing a comprehensive insight into the dynamic behavior inherent within the dynamic system. Hence, the method is called "dynamic mode decomposition" [17]. For any measurement function (observable) of the state of a nonlinear dynamical system (NLDS) in the Hilbert space, its evolution is governed by the infinite dimensional linear Koopman operator. The spectral decomposition of the linear Koopman operator can fully characterize the dynamics in NLDS [10]. The Koopman operator dictates the evolution of all measurement functions of the state via a composition operation onto the underlying nonlinear dynamics. The advantage being, the dictating process of the measurement functions is linear [10] regardless of the nature of the underlying dynamics. Therefore, the Koopman operator theory of dynamical systems provides a linear dynamical system in terms of measurement functions on the state space.
The Hilbert space considered here is the space of all square integrable measurement functions on the state space, thus it is an infinite dimensional function space. Therefore, the infinite dimensional linear Koopman operator evolves measurement functions of the state forward in time, providing the next measurement function. The relation between DMD and Koopman operator theory of dynamical systems has gained popularity in recent years. The DMD computes a finite and low dimensional approximation of the Koopman operator by solving an optimization problem, yielding a smaller matrix representation of the high or infinite dimensional Koopman operator. The smaller Koopman matrix's eigendecomposition allows for the characterization of the dynamical system, facilitating the extraction of spatiotemporal behaviors. This feature positions DMD favorably among modal reduction techniques.

1.1.1 Research Aim and Objectives

The infinite dimensional linear Koopman operator cannot be directly represented or computed in practice. The DMD algorithm computes a best possible (optimal) Koopman matrix, in the least squares sense, to approximate the infinite dimensional linear Koopman operator based on a homogeneous linear model of a dynamical system. In this work, we propose a nonhomogeneous linear model of a dynamical system, resulting to a modified DMD framework. This modified DMD is presented as an optimization problem, where the resulting Koopman matrix and the constant offset (nonhomogeneous) matrix are solutions that minimize the Frobenius norm of the proposed linear model. Numerical results demonstrate that the modified DMD offers improved performance over the standard DMD.

The DMD algorithm is extensively applied in modern data driven analysis of spatiotemporal data of high dimensional dynamical systems. DMD approximates an optimal linear matrix operator that governs the evolution of measurement data of a dynamic system. By defining a measurement subspace and taking direct measurements of the state of a dynamical system, DMD can approximate a Koopman operator restricted to that measurement subspace. DMD is entirely data dependant thus, does not require prior knowledge of the underlying governing equations. Furthermore, due to its simplicity DMD is highly amenable to problems from control theory, compressed sensing and multi resolution. The research endeavor is principally directed towards addressing the challenges associated to DMD. These include: longer time future state estimation, prediction and model stability of DMD. Consequently, the aim and objectives of this study can be succinctly encapsulated as follows:

Research Aim:

- To identify dynamic coherent patterns in dynamic data
- To predict future state of a dynamical system using linear dynamic models

Research Objectives:

- Stable predictions of system states over extended time duration
- Improve computational stability
- Minimize error in system state identification and state prediction
- Capture dominant patterns and their time dynamics faster and reliably

The contents of this thesis are structured as follows:

- Chapter 1: We have introduced the research topic, defined our objectives and aim.
- Chapter 2: We have reviewed and presented related research studies and the contributions made on the research topic. Highlighting the research gap for our proposed contribution to the research topic.
- Chapter 3: We have defined the theoretical underpinnings and the building blocks for this research study, including Koopman operator theory.
- Chapter 4: We have defined the primary mathematical tool essential for the DMD framework. The singular value decomposition is introduced as the best matrix factorization used in DMD.
- Chapter 5: We have introduced and defined the mathematical framework upon which this research topic is founded, namely, data driven analysis of dynamical systems, while also formulating our proposed optimization model and its solution.
- Chapter 6: We have conducted numerical tests to demonstrate the effectiveness of our proposed method compared to the standard approach.
- Chapter 7: We have concisely highlighted the key concepts of DMD and mDMD. A summary of our findings and out future work is given.

Chapter 2

Literature

Here, we present a general review of DMD-related data driven methods. The review is structured as follows: DMD and its applications are reviewed in Section 2.1, as well as connections to the theory of the Koopman operator. In Section 2.2, the DMD and its variants are briefly presented.

2.1 Dynamic Mode Decomposition and Dynamical systems

Exploiting low dimensionality in high dimensional systems to formulate data driven models through DMD is cheaper, manageable, and computationally efficient. Fundamentally, like POD, DMD utilises the singular value decomposition (SVD) [9]. The SVD/POD, specifically mines spatial features (modes) from a data decomposition. Remarkably, DMD extracts modes from a data decomposition with each mode having a corresponding temporal behaviour [4].
Following the original formulation of DMD [17], Rowley, Mezic, and their collaborators established a significant link between DMD and linear Koopman operator theory [14] [10]. In this approach, a system's state can be represented using a set of high dimensional measurement functions that are advanced linearly in time by the Koopman operator. This defines a linear dynamical system in terms of the measurement functions, governed by the linear Koopman operator [12]. Therefore, DMD computes the best-fit finite dimensional Koopman operator of the linear dynamical system [17].
DMD is a versatile algorithm for characterizing high-dimensional dynamical systems data. It works well with both experimental and measurement data. The DMD

algorithm may be perceived as optimally combining the SVD for the identification of low dimensional spatial features and the Fast Fourier Transforms (as discussed in Chapter 2 of [4]) for associated time dynamics. Thus, each DMD mode corresponds to a specific complex eigenvalue, denoted $\lambda = c + id$. Where d, being the imaginary component, denotes a distinct frequency of oscillation. The real component, c indicates the associated growth or decay.

2.2 Variants of DMD

The standard DMD algorithm uses direct (linear) measurements [4]. In the extended DMD (EDMD), William [20] considers an augmented vector that contains nonlinear measurements of the state. The objective in the EDMD is to enhance the basis for approximating the Koopman operator. It was found that unless the augmented vector of measurements spans the subspace that contain the Koopman eigenmeasurements, it is not possible to construct a finite dimensional Koopman operator. Baddoo [2] demonstrates how physical knowledge can be integrated into the DMD, calling his contribution physics informed DMD (piDMD). The approximate Koopman operator in this case is restricted to respect physical properties of the system, prior knowledge of the dynamical system is assumed to be known in this approach. Colbrook [8] introduced measure preserving EDMD, mpEDMD. The mpEDMD does not assume any prior physical knowledge like piDMD, instead, a measure is preserved whenever one computes the matrix approximation of the Koopman operator.

These recent variants of DMD discussed here aim to construct a matrix that best approximates the Koopman operator by using the homogeneous (standard) linear relation. However, there's a realm yet unexplored critically here, the nonhomogeneous (modified) linear relation of the underlying dynamics. We have approximated the linear operator based on the new linear dynamic model. Our aim is to primarily explore and generalize the standard linear model. Furthermore, the research will explore an alternative mathematical treatment of Koopman operator theory and DMD, through optimization techniques.

Chapter 3

Dynamical Systems

3.1 General overview, objectives and challenges

Dynamical systems describe the evolution of variables or states over time, often governed by interacting components. These interactions can be complex or nonlinear, especially in real-world scenarios where co-existing, co-evolving interactions are involved. Nonlinear dynamical systems (NLDS) provide a mathematical framework for capturing and studying this complexity, offering insights into behaviors such as chaos and bifurcation that are not easily handled by linear models [4]. However, the study of dynamical systems is rapidly evolving, prompting our exploration of both the established and emerging methodologies. These methods contribute to ongoing advancements while addressing current objectives and challenges associated with the study of dynamical systems. Among the objectives include: future state prediction, design and optimization, estimation and control, interpretability and physical understanding [4]. To these mentioned objectives, the challenges in the scenario include: real-world dynamical systems often operate across multiple scales in both space and time. Therefore, accurately measuring such complex systems, defining equations of motion, and identifying state parameters is uncertain. Furthermore, although it may be possible to derive equations of motion from first principles for certain real-world dynamical systems, for most, this approach is impractical [4].

Given the challenges and goals associated with studying dynamical systems, this thesis explores recent data-driven methods that fully attempt the discovery of governing equations by relying on data. The data-driven approach focuses on addressing two major challenges in analyzing dynamical systems: first, the nonlinearity of real world dynamical systems and the need to globally characterize their inherent

complexities. Second, the absence of governing equations due to the basic lack of well-established physical laws for many real world dynamical systems.
All models, whether classically constructed using first principals or data-driven, are inherently approximations and often become subjects of scrutiny. For example, for data-driven models, there are hidden variables crucial to the dynamics that frequently go unmeasured or unobserved, impacting on the model's accuracy. Uncovering these hidden variables is a central challenge in data-driven methods. It is also important to note that a third significant challenge is the high dimensionality inherent in many modern real world dynamical systems. In addition, despite formulating governing equations from high dimensional real world dynamical systems data, identifying patterns where dominant behaviours evolve, in order to construct reduced order models, remains significantly challenging [4]. However, identifying the unknown dynamics and defining intrinsic coordinates that can linearly represent a complex system are two major objectives in modern dynamical systems.
There are two key approaches in data-driven methods for the analysis of modern or real world dynamical systems: the operator-theoretic framework and data-driven regression and machine learning. This thesis explores the former, focusing on the implementation of the Koopman operator [10], in which, the evolution of direct measurements or observations from real world dynamical systems by the operator is approximated. Furthermore, the Koopman operator theory of dynamical systems provides a method to identify intrinsic coordinate systems that allows for a linear representation of nonlinear dynamics.
The layout in this chapter is in the following order: first, we present the benefits of representing nonlinear dynamical systems as linear dynamical systems in Section 3.2. Second, the role of the Koopman operator theory in dynamical systems, in view of providing a linear representation, is presented in Section 3.3.

3.2 Preliminaries

The section provides an overview of both continuous-time and discrete-time dynamics. In conclusion, this section highlights a linear dynamical system and the associated analytical advantages from its spectral decomposition.

3.2.1 Mathematical introduction

Continuous-time Systems

Our primary focus has been on real-world dynamical systems which are frequently classified as nonlinear, and classically characterized by ordinary differential equations. Mathematically, a dynamical system is represented as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t). \tag{3.1}$$

In this context, the state of the dynamical system, represented by $\mathbf{x}$ is assumed to be a vector in some smooth submanifold $\mathbb{X}$ such that $\mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^d$. The vector field $\mathbf{f}$, which describes the dynamics may depend on both $\mathbf{x}$ and time t, is assumed to be a continuous (nonlinear) function defined on the finite dimensional state space $\mathbb{X}$ such that $\mathbf{f} : \mathbb{X} \to \mathbb{X}$ [4]. Such a dynamical system is typically referred to as a non-autonomous system. For simplicity, we have focused on the autonomous system, one without an explicit dependency on t, such that:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)). \tag{3.2}$$

The existence of unique solutions to the dynamical system (3.2) is ensured under the assumptions made about $\mathbf{x}$ and $\mathbf{f}$. For a more detailed and general discussion, refer to [1]. Next, we have presented the autonomous system (3.2) in the discrete-time setting [14].

Discrete-time Systems

In practice, one usually has access to discrete-time measurements (data) of the dynamical system. Thus, it is natural to adopt a discrete-time dynamical system.

Discrete-time dynamical systems encompass both hybrid and discontinuous systems, offering a broader representation of real-world dynamics compared to continuous-time dynamics [4]. The discrete-time dynamical system, also referred to as a map $\mathbf{F} : \mathbb{X} \to \mathbb{X}$, advances a state $\mathbf{x}_k$, at $t = k$, to a future state $\mathbf{x}_{k+1}$, at $t = k + 1$:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k), \tag{3.3}$$

discrete-time dynamics may be derived from continuous-time dynamics by sampling the trajectory of (3.2) at discrete time points, where the state at a discrete time $t = k$ is now obtained as $\mathbf{x}(t = k) = \mathbf{x}(k\Delta t)$. The discrete-time propagator or map $\mathbf{F}$ is now parameterized by a time step Δt, $\mathbf{F}^{\Delta t}$. For this continuous to discrete-time relation and any time t arbitrarily chosen, we can define a time $t-$map that governs the evolution of the trajectories according to $\mathbf{F}^t : \mathbb{X} \to \mathbb{X}$, given as :

$$\mathbf{F}^{t_0+t}(\mathbf{x}(t_0)) = \mathbf{x}(t_0 + t) = \mathbf{x}(t_0) + \int_{t_0}^{t_0+t} \mathbf{f}(\mathbf{x}(\mu))d\mu, \tag{3.4}$$

where the map $\mathbf{F}^{t_0+t}$ is a repeated $(t_0 + t)$-fold composition of $\mathbf{F}$, $\mathbf{F}^{t_0+t} = \mathbf{F}(\mathbf{F}(\mathbf{F}(\mathbf{F})))$. Thus, the generator of this composition semigroup is denoted by $\mathbf{F}$. Nonetheless, in (3.4), the future state of the dynamical system at time $t_0 + t$, is defined by $\mathbf{F}^{t_0+t}$, which advances the initial states forward along the trajectory by a t over the time interval $[t_0, t_0 + t]$.

The aim in modern dynamical studies analysis is to provide a coordinate transformation. For the continuous time systems (3.2), we seek a linearizing coordinate system : $\mathbf{c} = \varphi(\mathbf{x})$ or $\mathbf{x} = \varphi(\mathbf{c})$ such that the dynamics are linear $\dot{\mathbf{c}} = \mathbf{A}\mathbf{c}$. Similarly for the discrete time system (3.3) we seek a linearizing coordinate system such that the dynamics are linear: $\mathbf{c}_{k+1} = \mathbf{K}\mathbf{c}_k$. Here, the matrix $\mathbf{K}$ is a discrete time analog of the continuous time matrix $\mathbf{A}$ and $\mathbf{c}$ is a vector of linearizing coordinates. Such linear coordinate transformations may be given by the eigenfunctions of the discrete time Koopman operator, to be discussed in Section 3.3. For now, we provide an overview of the linear dynamical system and appreciate its advantages when used to represent dynamical systems.

Linear Dynamics and Spectral Decomposition

The thesis particularly explores the linear dynamical system constructed by DMD based on the Koopman operator theory of dynamical systems. Before getting into the DMD formulation, we first motivate this approach by presenting and discussing the advantages of constructing linear dynamic systems for dynamical systems. A linear dynamical system is typically expressed in the form:

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} \quad \text{subject to} \quad \mathbf{x}(t_0) = \mathbf{x}_0. \tag{3.5}$$

Here, $\mathbf{A}$ is the system matrix, determining the systems state dynamics. The initial condition of the system is given as $\mathbf{x}(t_0)$. The advantages of working with the form in (3.5) include guaranteed closed-form solutions and access to a wide range of analytical methods for prediction, numerical simulation, estimation, and control [9]. By taking an educated guess and using the initial condition of the system. The solution to (3.5) is:

$$\mathbf{x}(t_0 + t) = e^{\mathbf{A}t}\mathbf{x}(t_0), \tag{3.6}$$

where the matrix exponential $e^{\mathbf{A}t}$ is the solution map and is defined as follows:

$$e^{\mathbf{A}t} = \sum_{k=0}^{\infty} \frac{\mathbf{A}^k t^k}{k!} \quad = \quad (\mathbb{I} + \mathbf{A}t + \frac{\mathbf{A}^2 t^2}{2!} + \frac{\mathbf{A}^3 t^3}{3!} + \dots), \tag{3.7}$$

where ! denotes a factorial. A major advantage of linear dynamical systems is that the dynamics of the system can be completely characterized by the eigenvalues and eigenvectors of the matrix $\mathbf{A}$. The eigenvalues and eigenvectors are obtained from the spectral decomposition of the matrix $\mathbf{A}$:

$$\mathbf{A}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}. \tag{3.8}$$

In the simplest scenario where $\mathbf{A}$ is of full rank, we can assumed that the matrix $\mathbf{A}$ possesses d distinct eigenvalues. These eigenvalues, denoted by λ_j, are arranged in the diagonal matrix $\mathbf{\Lambda}$. The columns of the matrix $\mathbf{W}$ correspond to the linearly independent eigenvectors ω_j associated with the eigenvalues λ_j. Consequently, applying the spectral decomposition in equation (3.8), we have assumed that $\mathbf{A}$ is square and diagonalizable, such that $\mathbf{A} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}$. Thus, substituting the spectral decomposition of $\mathbf{A}$ into (3.7), the matrix exponential gives the result:

$$
\begin{aligned}
e^{(\mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1})t} &= (\mathbb{I} + (\mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1})t + \frac{(\mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1})^2 t^2}{2!} + \frac{(\mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1})^3 t^3}{3!} + \dots), \\
&= (\mathbb{I} + (\mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1})t + \frac{(\mathbf{W}\mathbf{\Lambda}^2\mathbf{W}^{-1})t^2}{2!} + \frac{(\mathbf{W}\mathbf{\Lambda}^3\mathbf{W}^{-1})t^3}{3!} + \dots), \\
&= (\mathbf{W}\mathbf{W}^{-1} + (\mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1})t + \frac{(\mathbf{W}\mathbf{\Lambda}^2\mathbf{W}^{-1})t^2}{2!} + \frac{(\mathbf{W}\mathbf{\Lambda}^3\mathbf{W}^{-1})t^3}{3!} + \dots), \\
&= \mathbf{W}(\mathbb{I} + (\mathbf{\Lambda})t + \frac{(\mathbf{\Lambda}^2)t^2}{2!} + \frac{(\mathbf{\Lambda}^3)t^3}{3!} + \dots)\mathbf{W}^{-1}, \\
&= \mathbf{W}(e^{\mathbf{\Lambda}t})\mathbf{W}^{-1}.
\end{aligned}
$$

Thus,

$$\mathbf{x}(t_0 + t) = e^{(\mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1})t}\mathbf{x}(t_0) = \mathbf{W}e^{\mathbf{\Lambda}t}\mathbf{W}^{-1}\mathbf{x}(t_0). \tag{3.9}$$

Here, $e^{\mathbf{\Lambda}}$ is the matrix containing the exponential of each diagonal element in $\mathbf{\Lambda}$
The solution map $e^{\mathbf{A}t}$ in (3.6) defines the flow map (discrete time) $\mathbf{F}_t$. The discrete eigenvalues will be given by $e^{\lambda t}$. In this simple context, the system is described in the standard or original coordinate system. These coordinate systems are typically coupled: the system's dynamics are interconnected and may be difficult to analyze directly. The matrix $\mathbf{W}^{-1}$, defines a coordinate change or change of basis, moving from the standard coordinate system to intrinsic eigenvector coordinates $\mathbf{c}$ given as, $\mathbf{c} = \mathbf{W}^{-1}\mathbf{x}$. The coordinate change transforms the underlying coupled dynamics to a new basis defined by the eigenvectors of $\mathbf{A}$. In this new basis, the system's behavior becomes much easier to analyze because the dynamics associated with each eigenvector (or eigenvalue) are independent of the others, effectively decoupling the system:

$$\frac{d}{dt}\mathbf{c} = \mathbf{\Lambda}\mathbf{c}. \tag{3.10}$$

Thus, each coordinate, denoted as c_j, evolves independently, with simplified dynamics:

$$\frac{d}{dt}c_j = \lambda_j c_j. \tag{3.11}$$

Thus, it is advantageous to attain a linear dynamic system to represent a dynamical system: the linear dynamic system has closed form solutions and makes it possible

to decouple the dynamic system through simple linear transformations of coordinates. Furthermore, providing a robust framework for analytical prediction of dynamical systems future state, optimal estimation and control methods. At the heart of the approach taken in this research is a data-driven computation of an associated linear Koopman operator that governs the linear dynamic system of a dynamical system. The next section gives a theoretical introduction to the theory of Koopman operators in dynamical systems and data driven analysis.

3.3 Koopman operator theory

The Koopman operator theory of dynamical systems provides a mathematical framework for a global and explicit linearization of nonlinear dynamics. We explore an application of the Koopman operator in enhancing data-driven linear dynamic models—a concept pioneered by Mezic [14]. Recently, Koopman operator theory of dynamical systems has been extensively connected to dynamic mode decomposition (DMD), a data driven method that constructs a linear dynamic model which approximates and characterizes nonlinear dynamics [16]. Originally, the Koopman operator, as developed by Bernard O. Koopman is unitary and used to characterize Hamiltonian systems [10]. A year after its inception, Koopman and Von Neuman generalized the Koopman operator theory to encompass dynamical systems with a continuous eigenvalue spectrum [10]. For the past two decades, the Koopman operator theory of dynamical systems has been extended from smooth and measure preserving dynamics to non-smooth and dissipative dynamics in more complex systems [14].

The Koopman operator theory of dynamical systems establishes that a dynamical system, typically governed by a nonlinear known or unknown function $\mathbf{f}$ in terms of the state $\mathbf{x}$ on a finite dimensional state space, can be represented as a linear dynamical system governed by the linear Koopman operator $\mathcal{K}$ in terms of measurement functions (observables) of the state, denoted $g(\mathbf{x})$, but on an infinite-dimensional observables space. In this thesis, the Koopman operator acts on the measurement functions of the system's state, which are chosen to satisfy the space of square-integrable functions within a Hilbert space, $\mathcal{H}$. Specifically, the linear Koopman operator governs the evolution of these observables as they reflect the underlying dynamics. [14]. The Koopman operator is a linear operator that acts on all possible

measurement functions, whether countably infinite or uncountable. Interestingly, its spectral decomposition allows for the full(global) characterization of nonlinear dynamical systems. However, to capture the complexity of a nonlinear dynamical system with all possible measurement functions, the linear Koopman operator must be infinite-dimensional. In one end, the Koopman operator theory of dynamical systems offers an alternative perspective to traditional point linearization methods. The Koopman operator enables the formulation and complete linear characterization of nonlinear dynamics through the evolution of state measurement functions or state observables, providing deeper insights into the system's behavior without the limitations associated to point or local linear approximations of nonlinear dynamics. How one computes a Koopman operator is the ultimate focus our research. In modern data-driven practices aimed at constructing linear dynamic models for nonlinear dynamics, finite-dimensional approximations of the Koopman operator are typically computed using the DMD algorithm. This approach highlights the relevance of DMD in tackling nonlinear dynamical systems, particularly in the era of big data. In particular, DMD computes the approximate Koopman operator matrix, which provides Koopman eigenfunctions and eigenvalues. These DMD constructed Koopman eigenquantities capture both spatial and temporal information associated to the dynamical system unlike traditional data driven methods which often only provide spatial features. Although computing these finite approximations of the Koopman operator (Koopman matrices) present its own challenges, Koopman matrices have become a central focus in modern data driven research on dynamical systems. Together, the Koopman operator theory and its approximation via DMD offer practical applications in addressing challenges of stability assessments and control of nonlinear dynamical systems, further cementing their importance in data-driven nonlinear analysis.

In Koopman operator theory applied to dynamical systems, scalar-valued observables of the system's state, also known as measurement functions $g(\mathbf{x})$ are linearly advanced forward in time by the Koopman operator. These measurement functions $g(\mathbf{x}) : \mathbb{X} \to \mathbb{C}$ takes the state $\mathbf{x}$ from a finite-dimensional state space $\mathbb{X}$ and maps it into a potentially infinite-dimensional space of observables, denoted $\mathcal{G}(\mathbb{X})$ in $\mathcal{H}$. The Koopman operator $\mathcal{K}$ then acts on these observables: $\mathcal{K} : \mathcal{G}(\mathbb{X}) \to \mathcal{G}(\mathbb{X})$, advancing them forward in time.

The space of observables $\mathcal{G}(\mathbb{X})$ is carefully selected from spaces with specific mathematical properties, such as completeness and inner product structure, to ensure analytical stability. These observables are continuous functions of the state on $\mathbb{X}$, defined within a function space. In this thesis, we focus on Hilbert spaces, specifically L^2-integrable functions. The choice of a Hilbert space guarantees a structured, well-behaved setting, enabling a consistent analysis of complex, nonlinear dynamics through a linear operator framework. Thus, $\mathcal{G}(\mathbb{X})$ provides a higher-dimensional representation compared to $\mathbb{X}$, enhancing the capacity to capture and study the nonlinear dynamics in a linearized form.
In discrete time dynamics, the discrete time family of maps $\mathbf{F}^t$ for the autonomous dynamical system is a t-times (t-fold) repeated mapping, $\mathbf{F}^t(\mathbf{x}_0) = \mathbf{x}_t$. The corresponding discrete-time family of Koopman operators $\mathcal{K}^t$, is defined as:

$$\mathcal{K}^t g(\mathbf{x}_0) = g(\mathbf{x}_0) \circ \mathbf{F}^t = g(\mathbf{F}^t(\mathbf{x}_0)) = g(\mathbf{x}_t) \;\; \forall g \in \mathcal{G}(\mathbb{X}), \tag{3.12}$$

where $\circ$ represents the composition operator, the discrete-time Koopman operator composes the measurement function with the underlying discrete time map. The result (3.12) holds true for all measurement functions in $\mathcal{G}(\mathbb{X})$. The one parameter family of functions (3.12) is a generalization of the one time step forward of the discrete Koopman operator:

$$\mathcal{K}^{t_1} g(\mathbf{x}_{t_0}) = g(\mathbf{x}_{t_1}), \quad \mathcal{K}^{t_2} g(\mathbf{x}_{t_1}) = g(\mathbf{x}_{t_2}), \ldots, \mathcal{K}^{t} g(\mathbf{x}_{t-1}) = g(\mathbf{x}_t), \tag{3.13}$$

where, an observation g of the state at time t_i given as $g(\mathbf{x}_{t_i})$ is linearly advanced to a next time step. Thus:

$$g(\mathbf{x}_t) = \mathcal{K}^t g(\mathbf{x}_{t_0}), \tag{3.14}$$

here, the Koopman operator advances the initial observable of the system's state to an observable in the future time t. This defines a discrete time linear dynamical system governed by the Koopman operator. Furthermore, since the addition operation in function spaces ($\mathcal{G}(\mathbb{X})$) is linear, the Koopman operator is linear:

$$\begin{aligned} \mathcal{K}^t(c_1 g_1(\mathbf{x}) + c_2 g_2(\mathbf{x})) &= c_1 g_1(\mathbf{F}^t(\mathbf{x})) + c_1 g_2(\mathbf{F}^t(\mathbf{x})), && (3.15) \\ &= c_1 \mathcal{K}^t g_1(\mathbf{x}) + c_2 \mathcal{K}^t g_2(\mathbf{x}). && (3.16) \end{aligned}$$

This is true for all observables or measurement functions regardless of $\mathbf{F}^t$ being linear or nonlinear. Notably, the discrete time $\mathcal{K}^t$ in (3.14) is a repeated application t-times of the Koopman operator $\mathcal{K}$. Thus, *the* Koopman operator $\mathcal{K}$ is the generator of the (countable) composition semigroup, giving rise to a discrete-time linear dynamical system:

$$g(\mathbf{x}_{k+1}) = \mathcal{K} g(\mathbf{x}_k), \tag{3.17}$$

the dynamical system is linear and infinite dimensional.
In continuous time dynamics, the flow map $\mathbf{F}^t$ satisfies some properties of a semigroup $\mathbf{F}^{t+s}(\mathbf{x}) = \mathbf{F}^t(\mathbf{F}^s(\mathbf{x}))$ for all $\mathbf{x}$ and $t, s \geq 0$. These properties can be extended to a group, given that the flow map is invertible. These properties are inherited by the Koopman operator family $\mathcal{K}^t$ as well. Additionally, for sufficiently smooth dynamics, we take the limit as $t \to 0$, defining an infinitesimal generator of $\mathcal{K}^t$ family [7]. The infinitesimal generator, denoted as $\mathcal{L}$, is defined as:

$$\begin{aligned} \mathcal{L} g &= \lim_{t \to 0} \frac{\mathcal{K}^t g - g}{t}, \\ &= \lim_{t \to 0} \frac{g \circ \mathbf{F}^t - g}{t}. \end{aligned}$$

Considering the dynamics given by (3.2) and taking the chain rule:

$$\frac{d}{dt} g(\mathbf{x}(t)) = \nabla g \cdot \dot{\mathbf{x}}(t) = \nabla g \cdot \mathbf{f}(\mathbf{x}(t)), \tag{3.18}$$

where $\mathcal{L}$ is the Lie operator since it is the Lie derivative of g along $\mathbf{f}$, but:

$$\frac{d}{dt} g(\mathbf{x}(t)) = \lim_{\tau \to 0} \frac{g(\mathbf{x}(t+\tau)) - g(\mathbf{x}(t))}{\tau} = \mathcal{L}(g(\mathbf{x}(t))). \tag{3.19}$$

Thus, using (3.18) and (3.19), we have that:

$$\mathcal{L}g(\mathbf{x}(t)) = \nabla g \cdot \mathbf{f}(\mathbf{x}(t)) \tag{3.20}$$

where the continuous-time linear dynamical system, is induced by $\mathcal{L}$:

$$\frac{d}{dt}g(\mathbf{x}(t)) = \mathcal{L}g(\mathbf{x}(t)), \tag{3.21}$$

the dynamical system is linear and infinite dimensional. Notably, the linear dynamical systems (3.17) and (3.21) are each an analogue to the dynamical systems (3.3) and (3.2) respectively. The measurement functions typically project specific components of the actual system's state, such that $g(\mathbf{x}) = x_i$. However, for simplicity, we have adopted the notation $g(\mathbf{x}) = \mathbf{x}$ [7]. In which case, the left hand side of (3.21) is $\dot{\mathbf{x}}$ but the right hand side $\mathcal{L}\mathbf{x}$ remains a challenge to represent in a chosen basis for the space of measurement functions. In practice, and only for certain structures, representing $\mathcal{L}\mathbf{x}$ typically requires an infinite number of measurements of the state. In essence, the Koopman operator theory of dynamical systems offers an appealing framework to analyze nonlinear dynamics which are originally defined on a finite dimensional state space to defining linear dynamics and an infinite-dimensional observable space. However, this approach comes with two major challenges: (1) representing the Koopman operator and (2) computing it within the space of observables. Additionally, when the underlying state space dynamics are nonlinear, individual measurement functions may exhibit nonlinear behavior over time. This makes it impractical to attempt working with all possible measurement functions to linearly capture the nonlinear dynamical system.

Modern Koopman analysis addresses these challenges by focusing on Koopman eigenfunctions, $\varphi_j(\mathbf{x})$, which are special observables that evolve linearly in time, regardless of the nonlinear nature of the underlying dynamics. The Koopman operator's primary advantage lies in its eigendecomposition, which provides a linear yet meaningful characterization of complex dynamical behavior. Measurement functions that might otherwise evolve nonlinearly in time can often be expressed as linear combinations of Koopman eigenfunctions, which maintain linear evolution under the action of the Koopman operator regardless of the system's nonlinear dynamics. This is because Koopman eigenfunctions are directly tired to the Koopman operator's eigenvalues, providing a stable, linearized framework for capturing

nonlinear behavior.
However, the representation of measurement functions using Koopman eigenfunctions is only possible if the measurement function can be expressed in the eigenfunction basis. Not all measurement functions have a straightforward Koopman decomposition, especially in complex systems. In some cases, a small number of dominant eigenfunctions suffice to approximate the system's behavior effectively, offering a practical linear approximation for complex dynamics. In more complex systems, however, a large number of eigenfunctions may be necessary, which increases the difficulty of achieving a reliable approximation.
In summary, a central goal in Koopman operator theory for dynamical systems is to represent measurement functions as a linear combination of Koopman eigenfunctions. This representation provides the following key benefits:

- Linear analysis of Nonlinear Dynamics:

  Representing measurement functions in terms of Koopman eigenfunctions enables the analysis of nonlinear dynamics within a linear framework. This representation provides a simplified approach to analyze and predict nonlinear dynamics.

- Reduced complexity through a finite eigenfunction basis :

  In practical applications, a nonlinear dynamical system is often approximated by a finite set of dominant Koopman eigenfunctions. This finite basis captures the primary dynamics of the system in a lower-dimensional, computationally feasible, and linear framework.

- Enhanced control and prediction capabilities in nonlinear dynamics:

  Expressing measurement functions as a combination of Koopman eigenfunctions allows for linear control strategies and predictive modeling in systems with nonlinear dynamics.

Each Koopman eigenfunction evolves over time with a specific frequency or rate given by the associated eigenvalue. By considering a discrete-time eigenvalue λ, the corresponding Koopman eigenfunction satisfies:

$$\varphi(\mathbf{x}_{k+1}) = \mathcal{K}^t \varphi(\mathbf{x}_k) = \lambda^t \varphi(\mathbf{x}_k). \tag{3.22}$$

Similarly, in the continuous time setting:

$$\frac{d}{dt}\varphi(\mathbf{x}) = \mathcal{L}\varphi(\mathbf{x}) = \omega\varphi(\mathbf{x}). \tag{3.23}$$

The eigenfunctions of $\mathcal{L}$ are also eigenfunctions of $\mathcal{K}$ with different eigenvalues representations, where $\lambda^t = exp(\omega t)$. In [7], it is shown that eigenfunctions of $\mathcal{K}$ corresponding to $\lambda \neq 0$ are likewise of $\mathcal{L}$. The eigenfunctions define linear coordinate transformations so that the Koopman matrix $\mathbf{A}$ can be constructed. The next section reviews the second challenge: a finite dimensional approximation of the infinite dimensional Koopman operator.

3.3.1 Finite dimensional representations of the Koopman operator

In practice, multiple measurements of a dynamical system's state are taken and arranged in a vector, as opposed to the previously considered scalar measurements or eigenmeasurements. Therefore, it is common to consider a vector $\mathbf{g}(\mathbf{x})$ consisting of p measurement functions, where $p \in \mathbb{R}$:

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_p(\mathbf{x}) \end{bmatrix}. \tag{3.24}$$

Each measurement function in (3.24) $g_i(\mathbf{x})$ can be represented as a linear combination of the Koopman eigen functions $\varphi_j(\mathbf{x})$. To all Koopman eigenfunctions, there are Koopman modes v_{ij}, coefficients that provide a quantifiable information as to how much each Koopman eigenfunction overlaps with an observable quantity in the system (measurement function). Thus, the Koopman eigenfunctions and Koopman modes offer a structured way to represent measurement functions, forming a basis of eigenfunctions for the Hilbert space. Mathematically:

$$g_i(\mathbf{x}) = \sum_{j=1}^{\infty} v_{ij}\varphi_j(\mathbf{x}), \quad i = [1, \ldots, p]. \tag{3.25}$$

Thus, using (3.25), (3.24) becomes:

$$\mathbf{g}(\mathbf{x}) = \sum_{j=1}^{\infty} \varphi_j(\mathbf{x})\mathbf{v}_j, \tag{3.26}$$

where φ_j is a j-th Koopman eigenfunction which captures the underlying linear dynamics. Associated to the Koopman eigenfunction is a Koopman mode denoted by $\mathbf{v}_j$. The Koopman modes are in essence, the dynamic modes to be discussed in Chapter 5. This expansion of the measurement functions may be perceived as a change of basis into eigenfunction coordinates.
The Koopman operator is unitary on the space $\mathcal{G}(\mathbb{X})$ of square-integrable functions with respect to the conserved measure. Thus, for conservative systems, the Koopman eigenfunctions are orthonormal. Each eigenfunction represents a unique, independent mode of the system's behavior and is scaled to have a unit magnitude. Therefore, the Koopman modes are obtained by computing the inner product of the Koopman eigenfunctions and the measurement functions of the state as follows:

$$\mathbf{v}_j = \begin{bmatrix} \langle \varphi_j, g_1 \rangle \\ \langle \varphi_j, g_2 \rangle \\ \vdots \\ \langle \varphi_j, g_p \rangle \end{bmatrix}, \tag{3.27}$$

where $\langle \cdot, \cdot \rangle$ is inner product of functions on $\mathcal{H}$. Applying the representation (3.26) and (3.14), the dynamics of the measurement vector (3.24) can be given as follows:

$$\begin{aligned} \mathbf{g}(\mathbf{x}) &= \mathcal{K}_1 \mathbf{g}(\mathbf{x}_0), \\ \mathbf{g}(\mathbf{x}_2) &= \mathcal{K}\mathbf{g}(\mathbf{x}_1) = \mathcal{K}\mathcal{K}\mathbf{g}(\mathbf{x}_0), \\ \mathbf{g}(\mathbf{x}_3) &= \mathcal{K}\mathbf{g}(\mathbf{x}_2) = \mathcal{K}\mathcal{K}\mathcal{K}\mathbf{g}(\mathbf{x}_0), \\ \vdots &= \vdots, \end{aligned}$$

thus:

$$\mathbf{g}(\mathbf{x}_t) = \mathcal{K}^t\mathbf{g}(\mathbf{x}_0) = \mathcal{K}^t\sum_{j=1}^{\infty}\varphi_j(\mathbf{x}_0)\mathbf{v}_j, \tag{3.28}$$

$$= \sum_{j=1}^{\infty}\mathcal{K}^t\varphi_j(\mathbf{x}_0)\mathbf{v}_j, \tag{3.29}$$

$$= \sum_{j=1}^{\infty}\lambda_j^t\varphi_j(\mathbf{x}_0)\mathbf{v}_j. \tag{3.30}$$

Thus, the vector of measurements of the state at t, denoted $\mathbf{g}(\mathbf{x}_t)$ is equivalent to applying the discrete time Koopman operator t times to the vector of measurements of the state at t_0, denoted $\mathbf{g}(\mathbf{x}_0)$. In particular, resulting to the Koopman mode decomposition, which is a sum of an infinite sequence of triplets, $\{\varphi_j, \lambda_j, \mathbf{v}_j\}_{j=1}^{\infty}$. It is not practical to capture the space of all measurement functions in $\mathcal{H}$. Thus, a Koopman-invariant subspace given by a span of measurement functions (3.24) is constructed. In this invariant subspace, a finite dimensional matrix approximation $\mathbf{A}$ of the Koopman operator is attainable such that $\mathbf{A}$ acts on the space $\mathbb{R}^p$ [13]. With the coordinates given by $g_i(\mathbf{x})$, the matrix $\mathbf{A}$ governs a finite dimensional linear dynamical system. A finite set of Koopman eigenfunctions will definitely span an invariant subspace, however, obtaining these eigenfunctions is challenging. In practice, approximations of the invariant subspace are made, from a finite set of measurement functions $\{g_j(\mathbf{x})\}_{j=1}^{p}$ where each measurement function of the state is approximated by a finite number of Koopman eigenfunctions $g_j(\mathbf{x}) \approx \sum_{k=1}^{p} v_{jk}\varphi_k(\mathbf{x})$. The measurement functions of the state considered in this thesis are the direct measurements of the state, $\mathbf{g}(\mathbf{x}) = \mathbf{x}$. With these direct measurements of the state, Koopman modes are coherent spatial structures (modes) with the same temporal linear behaviour. Mezic in 2005 [14] formally connected the infinite sequence of eigenquantities (eigenfunctions, eigenvalues and modes) from the Koopman mode decomposition to data-driven regression through the DMD. DMD of the data matrices approximate dominant terms of the Koopman mode decomposition. In particular, DMD mode amplitudes approximate the Koopman eigenfunctions at $\varphi(\mathbf{x}_0)$, DMD modes approximate the Koopman modes and DMD eigenvalues approximate the Koopman eigenvalues, this is shown in Chapter 5, DMD . Before delving into the DMD, the next section reviews data matrix factorizations, crucial to DMD.

Chapter 4

Matrix decomposition

4.1 Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) is a key matrix factorization technique with broad computational utility in the big data era. SVD offers a stable and guaranteed decomposition of matrices which are applicable in diverse data-driven purposes. In this thesis, the SVD is used for two purposes: first, to provide low-rank approximations of high dimensional data matrices. Second, to enable the computation of pseudo-inverses for non-square matrices, providing solutions to overdetermined and underdetermined data matrix equations [4]. Other applications of SVD include as an underlying algorithm in Principal Component Analysis (PCA). In PCA, dominant statistically descriptive factors are extracted from the high dimensional data. Furthermore, SVD is a more systems tailored data driven technique as opposed to the fast Fourier transform, FFT, which is applied in a more general setting [4].
Practically, one often has access to data of a dynamical system, through experiment or simulation. The data is arranged in a data matrix, where each column represents all measurements taken at a particular time. Given that each column is often multi dimensional, the data column is flattened out (reshaped) so that it is now high dimensional. Remarkably, the high dimensional columns can be explained by some dominant patterns within them. This means that the high dimensional data matrix often exhibits low rank approximations. SVD is numerically robust in capturing the low rank approximations in the high dimensional data matrices of dynamical systems.

The data matrices are generally a time series of data and high dimensional, $\mathbf{X1} \in \mathbb{R}^{d\times m}$. The columns of the input data matrix are tall, $d << m$, given as:

$$\mathbf{X1} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t=1) & \mathbf{x}(t=2) & \dots & \mathbf{x}(t=m) \\ | & | & & | \end{bmatrix}. \tag{4.1}$$

The columns $\mathbf{x} \in \mathbb{R}^d$ represent the measurements of the state space where each column represents a distinct time point t. The longer the column, the bigger the system's state size is for that time. Thus, for a time series data set, m denotes the time at which the snapshots or measurements of a systems state are taken. Thus,the column with index $\mathbf{x}_k$ is the snapshot taken at the k^{th} time point. Thus, for (4.1), we have m snapshots.

Recall that the matrix $\mathbf{A}$ in (3.8) was assumed to be square, full rank and diagonalizable so that $\mathbf{A} = \mathbf{W}\Lambda\mathbf{W}^{-1}$. Unfortunately, here we have a data matrix (4.1) which is not necessarily a square matrix or full rank. SVD provides a diagonalization for any such matrix using two matrices, $\mathbf{U}$ and $\mathbf{V}$ whose columns are orthonormal bases. Using these matrices, the data matrix will become a diagonal matrix Σ. The diagonalization provided by SVD provides more information than that by (3.8). In addition, the eigenvectors in $\mathbf{W}$ are often not orthogonal, not enough and to solve (3.8) requires that $\mathbf{A}$ is a square matrix. These issues are addressed by singular vectors of $\mathbf{X1}$ in the SVD diagonalization.

The SVD diagonalization of the data matrix (4.1) with rank r begins by defining two orthogonal matrices $\mathbf{U} \in \mathbb{R}^{d\times d}$ and $\mathbf{V} \in \mathbb{R}^{m\times m}$, which are composed of orthonomal columns called the left singular vectors $\mathbf{u} \in \mathbb{R}^d$ of (4.1) and right singular vectors $\mathbf{v} \in \mathbb{R}^m$ respectively. The singular vectors provide bases for the four main subspaces: the orthonomal basis for the column space given as $\mathbf{u}_1, \dots, \mathbf{u}_r$, the orthonomal basis for the row space given as $\mathbf{v}_1, \dots, \mathbf{v}_r$, the orthonomal basis for the left null space given as $\mathbf{u}_{r+1}, \dots, \mathbf{u}_d$ and the orthonomal basis for the null space given as $\mathbf{v}_{r+1}, \dots, \mathbf{v}_m$.

First, we have presented the diagonalization of (4.1) using singular vectors. In addition to orthogonality, the singular vectors diagonalize (4.1) as follows:

$$\mathbf{X1}\mathbf{v}_1 = \sigma_1\mathbf{u}_1, \quad \mathbf{X1}\mathbf{v}_2 = \sigma_2\mathbf{u}_2, \quad \dots \quad , \mathbf{X1}\mathbf{v}_r = \sigma_r\mathbf{u}_r,$$

where $\sigma_1 \dots \sigma_r$ are called singular values such that σ_i will represent the length of $\mathbf{X1v}_i$. The σ_i are positive numbers contained in a diagonal matrix Σ.
We can also present the diagonalization of (4.1) using the orthogonal matrices (unitary matrices). Since they have orthonomal columns, the matrices $\mathbf{V}^* = \mathbf{V}^{-1}$ and $\mathbf{U}^* = \mathbf{U}^{-1}$ where the conjugate transpose is denoted by *. Therefore, from $\mathbf{X1v}_i = \sigma_i \mathbf{u}_i$ we can approximate $\mathbf{X1}$ using its own low rank matrix given by the r truncated SVD or economy SVD as follows:

$$\mathbf{X1} \approx \hat{\mathbf{U}}_r \hat{\Sigma}_r \hat{\mathbf{V}}_r^* \tag{4.2}$$

where

$$\hat{\mathbf{U}}_r \in \mathbb{C}^{d \times r}, \tag{4.3}$$
$$\hat{\mathbf{V}}_r \in \mathbb{C}^{m \times r}, \tag{4.4}$$
$$\hat{\Sigma}_r \in \mathbb{R}^{r \times r}. \tag{4.5}$$

Here, the singular vectors in the columns of (4.3) and (4.4) represent, respectively, the column space and row space of (4.1). However, we still have more orthonomal $(d - r)$ of $\mathbf{u}$ from the left null space and more orthonomal $(m - r)$ of $\mathbf{v}$ from the null space. These orthogonal null spaces can be included so that we have $\mathbf{U}$ and $\mathbf{V}$ as square orthogonal matrices, giving the untruncated (full) SVD, thereby $\mathbf{X1V} = \mathbf{U}\Sigma$. Thus, using (4.1), the full SVD is a unique matrix decomposition for the complex-valued data matrix $\mathbf{X1}$:

$$\mathbf{X1} = \mathbf{U}\Sigma\mathbf{V}^*, \tag{4.6}$$

where

$$\mathbf{U} \in \mathbb{C}^{d \times d}, \tag{4.7}$$
$$\mathbf{V} \in \mathbb{C}^{m \times m}, \tag{4.8}$$
$$\Sigma \in \mathbb{R}^{d \times m}. \tag{4.9}$$

Here, the full Σ is constructed by adding $(m - r)$ extra zero columns and $(d - r)$ extra zero rows to (4.5).
Therefore, (4.2) is the truncated SVD with bases for the column and row spaces.

The full SVD includes the null spaces and is given by (4.6). Both (4.2) and (4.6) decomposes (4.1) into r matrices of rank one. The truncated matrix (4.5) or full matrix (4.9) is a diagonal matrix of singular values, the singular values are real entries, non negative and decreasing down the diagonal. Therefore, the SVD splits the data matrix according to its dominant low-dimensional forms and variances in the measurements are ordered based on their significance [5]. Furthermore, the number of singular values not equal to zero is the rank of the matrix (4.1).
The SVD is connected to the symmetric eigenvalue problem (3.8), relating the singular values to the eigenvalues. The data matrix $\mathbf{X1}$ is not a square matrix but the matrix $\mathbf{X1}^*\mathbf{X1}$ is. By denoting $\mathbf{X1}^*\mathbf{X1}$ by $\mathbf{S}$, its eigenvalue problem is:

$$\mathbf{SW} = \mathbf{W\Lambda}, \tag{4.10}$$

where the eigenvalues of $\mathbf{S}$ denoted by λ_i are in the diagonal matrix $\mathbf{\Lambda}$. The squares of the singular values σ_i in Σ are eigenvalues of $\mathbf{X1}^*\mathbf{X1}$ and $\mathbf{X1X1}^*$. Thus, $\sigma_i^2 = \lambda_i$. Also, the orthonomal columns of $\mathbf{U}$ and $\mathbf{V}$ are eigenvectors of $\mathbf{X1X1}^*$ and $\mathbf{X1}^*\mathbf{X1}$. Furthermore, if $\mathbf{X1}$ is positive semidefinite(definite), then $\mathbf{X1} = \mathbf{U\Sigma V}^*$ coincides with $\mathbf{X1} = \mathbf{W\Lambda W}^{-1}$.
The singular values are linked to the Frobenius norm of $\mathbf{X1}$. The Frobenius norm calculates the size of the matrix, the square of the Frobenius norm is exactly the inner product of the matrix with itself:

$$\begin{aligned} \|\mathbf{X1}\|_F^2 = Tr(\mathbf{X1X1}^*) &= \langle \mathbf{X1}, \mathbf{X1} \rangle, && (4.11)\\ &= \sum_{i=1}^{d}\sum_{j=1}^{m} |\mathbf{x}_{ij}|^2, && (4.12)\\ &= \sum_{i=1}^{r} \sigma_i^2 && (4.13) \end{aligned}$$

where σ_i denotes the ith singular value of $\mathbf{X1}$ and $Tr(\cdot)$ denotes the trace of a matrix. The largest singular value of $\mathbf{X1}$ defines the operator norm (spectral norm), given as $\|\mathbf{X1}\| = \sigma_1$. Furthermore, the nuclear norm is defined in terms of singular values as $\|\mathbf{X1}\|_* = \sum_{i=1}^{r} \sigma_i$. Importantly, SVD provides an optimal, the least squares sense, hierarchy of low rank approximation of (4.1), returning the leading r singular values and vectors and discarding the rest [4]. The SVD tools are crucial in the next chapter.

Chapter 5

Dynamic Mode Decomposition

5.1 Introduction

In Chapter 3, the Koopman operator theory of dynamical systems gives a mathematical framework for the linear representation of dynamical systems in terms of the evolution of measurement functions of the state. In as much as this may sound mathematically enlightening, computing a finite dimensional Koopman operator and selecting the right measurement functions is not straight forward. Here, Dynamic mode decomposition (DMD) has rapidly become a standard method to approximate the infinite dimensional Koopman operator entirely from measurement data.

DMD was originally developed by Schmid in the fluid dynamics community where it is used to mine spatio-temporal coherent structures from high dimensional time dependent data [17]. In fluid dynamics, DMD is based on a data matrix decomposition called proper orthogonal decomposition (POD) which enables DMD to provide quantifiable dimensionality reductions. POD is now generally called SVD, as reviewed in Chapter 4. The hierarchical ordering of SVD modes is according to the variance that each mode captures from the original measurement data of the dynamical system. SVD's numerical robustness ensures that the ordered modes remain unchanged even when the data matrix is shuffled in time.

In contrast to SVD, a DMD mode is a linear combination of SVD modes. The linear combination is chosen particularly to extract spatial structures with the same coherent linear behavior in time which is given by growing or decaying oscillations of equal frequency. Thus, in practice DMD modes will be fewer compared to SVD modes, giving a dimensionality reduction in terms of a low dimensional set of spatial modes. In addition to providing a dimensionality reduction, DMD modes

provides a linear model as to how the amplitudes of the spatial modes evolve in time. This fundamental feature of DMD is remarkable and perceived as combining both the SVD/POD in space and fast Fourier transform (FFT) in time [4].
Koopman theory of dynamical systems and DMD based methods provide a perspective that is beyond the high applicability in data driven analytics of dynamical systems, adding that the robustness of the method can sufficiently account for numerical aspects involved. The DMD is currently decorated as a major algorithm for mining spatiotemporal structures from high dimensional dynamic data. Several factors have contributed to the DMD success in modern data driven methods, these include: firstly, the DMD algorithm approximates a best fit linear matrix that maps measurements of the state of a dynamical system forward in time. Therefore, DMD essentially computes a finite dimensional Koopman operator which is restricted to a state measurement subspace of a dynamical system (direct measurements of the state are are considered in our research). Secondly, since DMD is entirely based on measurement data, requires no knowledge about the underlying governing equations. Thus, DMD is flexible to processing both simulated and experimental data. Thirdly, DMD's formulation is founded on simple linear algebra. Thus, one can easily extend DMD to control theory, compressed sensing and other related applications. Based on the attributes, DMD is applicable beyond fluid dynamics, to include, neural science, signal processing and video processing as we have demonstrated in our numerical results, Chapter 6.
The rest of this chapter is presented as follows: in Section 5.1, we have introduced the original idea and formulation of DMD [17]. The original formulation of DMD requires regularly (uniformly) sampled time dynamics or data collected from a sequential time series $\{\mathbf{x}(t=k)\}_{k=1}^{m}$. In Section 5.2, we study the exact DMD method, a variant of the original DMD. In the exact DMD framework, the formulation can handle data from irregularly sampled time dynamics. Thus, the measurement data is given as pairs $\{\mathbf{x}(t=k), \mathbf{x}(t'=k)\}_{k=1}^{m}$ where each pair is a correspondence of a measurement. In Section 5.3, we consider an alternative approach to computing the inverse of a data matrix, in contrast to the classical SVD approach which assumes that the data matrix is not fully ranked. We have used matrix optimization and the Lagrangian multiplier. Finally, in Section 5.4, a modification on the exact DMD's linear approximation of the dynamical system is presented, where a regularizer is added and a resulting Koopman operator matrix is approximated.

DMD's original perspective

A historical overview is provided to introduce the DMD framework in analyzing high dimensional dynamical systems data. The DMD framework starts with a collection of snapshots of a dynamical system from a single trajectory that are uniformly sampled in time. A snapshot is typically a full measurement of a state. We have considered snapshots as direct measurements of the state which we denote as $g(\mathbf{x}) = \mathbf{x}$ where m snapshots of the dynamical system will be given as $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$. Each snapshot $\mathbf{x}_k$ is a high dimensional element of the vector space $\mathbb{R}^d$ where, $\mathbf{x}_k = \mathbf{x}(k\Delta t)$ represents a d dimensional data column at time $t = k$, where d is very large and Δt is chosen to be very small to account for the highest frequencies in the dynamical system. DMD utilises the snapshots to construct a linear dynamic model. The dynamic model is governed by a linear operator, mapping the snapshots from $t = k$ to $t = k + 1$. Therefore, the snapshots are regressed onto a local dynamic linear approximation:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k, \tag{5.1}$$

where the matrix $\mathbf{A}$ is DMD's best fit linear operator that approximately advances the snapshots, approximating the Koopman operator $\mathcal{K}$ restricted to the measurement subspace. In this case, the measurement subspace is spanned by $\mathbf{x}$. Thus, DMD computes the best fit Koopman operator for the underlying unknown dynamics.

DMD's original perspective proceeds by rearranging the snapshots into two data matrices, **X1** and **X2**, known as the input and output data matrices, respectively. By supposing that $m + 1$ snapshots are collected, we obtain:

$$\mathbf{X1} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t=1) & \mathbf{x}(t=2) & \ldots & \mathbf{x}(t=m) \\ | & | & & | \end{bmatrix}, \tag{5.2}$$

$$\mathbf{X2} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t=2) & \mathbf{x}(t=3) & \ldots & \mathbf{x}(t=m+1) \\ | & | & & | \end{bmatrix}. \tag{5.3}$$

The data matrix **X2** is a forward shift of the data matrix **X1** by a fixed Δt in time.

Since the snapshots are sequential and evenly sampled from the same trajectory, one can begin by representing (5.2) in terms of iterations of $\mathbf{A}$, according to (5.1):

$$\mathbf{X1} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t=1) & \mathbf{A}\mathbf{x}(t=1) & \dots & \mathbf{A}^{m-1}\mathbf{x}(t=1) \\ | & | & & | \end{bmatrix}. \tag{5.4}$$

The columns in (5.4) belong to the Krylov subspace. They are obtained using the linear operator $\mathbf{A}$ and the initial snapshot $\mathbf{x}(t=1)$. The matrix $\mathbf{A}$ hereby acts as the best fit of the first m snapshots of the dynamic system. By excluding the first column of (5.4), the rest of the columns equates to the first $m-1$ columns in (5.3) leaving out the last snapshot or state measurement column unfitted. Thus, DMD originally represented the final measurement $\mathbf{x}_{m+1} = \mathbf{x}(t=m+1)$ in terms of the krylov basis as a best fit combination of m columns of (5.2) that produces a smallest possible residual [17], relating the original DMD algorithm to Krylov subspaces and the Arnoldi algorithm:

$$\mathbf{x}_{m+1} = \sum_{k=1}^{m} b_m \mathbf{x}_m + \mathbf{r}. \tag{5.5}$$

Here b_m are coefficients of the columns of (5.4), while $\mathbf{r}$ represents the residual that will lie outside of the Krylov space [6].

However, the modern DMD algorithm according to [17] constructs the best fit of the state measurement points in an L^2 sense, the Euclidean norm of a vector, using a pseudo-inverse. The approach is aided by the SVD of the data matrix (5.4). Thus, the SVD based DMD computes the leading eigendecomposition of the matrix $\mathbf{A}$. The drawback to the SVD based DMD is that if the original data matrix (5.2) has no low dimensional or low rank dynamic structures, then the formulation fails immediately. Otherwise, we can generalize (5.1) in terms of the data matrices (5.2) and (5.3):

$$\mathbf{X2} \approx \mathbf{AX1}. \tag{5.6}$$

The linear operator $\mathbf{A}$ propagates each snapshot column in (5.2) by a Δt to a future snapshot column in (5.3). The matrix $\mathbf{A}$ is the solution that minimizes the Frobenius norm of $\|\mathbf{X2} - \mathbf{AX1}\|_F$ across all state snapshots. Thus, the matrix $\mathbf{A}$ is the best fit linear approximation for the data matrices (5.2) and (5.3). Schmid explored (5.6) and showed that the matrix $\mathbf{A}$ approximates the infinite dimensional Koopman operator

[11]. In [17], Schmid has shown that instead of computing the whole matrix $\mathbf{A}$ directly, a low-rank matrix $\tilde{\mathbf{A}}$ is computed based on SVD.
In Section 5.2, next, we have presented the modern definition of DMD namely, exact DMD [19]. From this point, we have explored the exact DMD formulation for our proposed modified linear approximation to considered in Section 5.4.

5.2 Exact DMD

The previous section introduced DMD, in which a uniform sampling of snapshots from a single trajectory is implemented. In exact DMD, these constraints have been relaxed. Snapshots are not uniformly sampled but each snapshot $\mathbf{x}(t = k)$ corresponds to a snapshot $\mathbf{x}(t' = k)$ one time step in the future. The time step Δt from t_k to t'_k is not necessarily evenly spaced or sequential. Each snapshot correspondence constitute a data pair, expressed as $\{\mathbf{x}(t = k), \mathbf{x}(t' = k)\}_{k=1}^{m}$, where $t = k \Longrightarrow t_k$ and $t = k + \Delta t \Longrightarrow t'_k$. As in the DMD, two data matrices **X1** and **X2** are arranged where **X1** will contain $\{\mathbf{x}(t = k)\}_{k=1}^{m}$ and **X2** will contain $\{\mathbf{x}(t' = k)\}_{k=1}^{m}$. The data matrices are given as:

$$\mathbf{X1} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t=1) & \mathbf{x}(t=2) & \dots & \mathbf{x}(t=m) \\ | & | & & | \end{bmatrix}, \tag{5.7}$$

$$\mathbf{X2} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t'=1) & \mathbf{x}(t'=2) & \dots & \mathbf{x}(t'=m) \\ | & | & & | \end{bmatrix}. \tag{5.8}$$

Here, the data matrices (5.7) and (5.8) can approximate (5.1) as follows:

$$\mathbf{X2} \approx \mathbf{A}\mathbf{X1}, \tag{5.9}$$

where $\mathbf{X1} \in \mathbb{R}^{d \times m}$, $\mathbf{X2} \in \mathbb{R}^{d \times m}$ and $\mathbf{A}$ is the best fit linear operator that approximately advances data between **X1** and **X2**.
The result (5.9) is formulated as an optimization problem.

For data sets given as (5.7) and (5.8), the operator $\mathbf{A}$ is a solution of the least squares minimization problem:

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X2} - \mathbf{AX1}\|_F = \mathbf{X2X1}^{\dagger}, \tag{5.10}$$

where † denotes the Moor-Penrose pseudoinverse, $\|\cdot\|_F$ is the Frobenius norm and $\mathbf{A} \in \mathbb{R}^{(d\times d)}$ is supposed to minimize the Frobenius norm of the difference. To find the minimizer $\mathbf{A}$ to (5.10), DMD computes the inverse of the non square data matrix $\mathbf{X1}$ using the Moore–Penrose pseudoinverse given by SVD of $\mathbf{X1}$: $\mathbf{X1} = \mathbf{U\Sigma V}^*$ and $\mathbf{X1}^{\dagger} = \mathbf{V\Sigma}^{-1}\mathbf{U}^*$ so that $\mathbf{A} = \mathbf{X2V\Sigma}^{-1}\mathbf{U}^*$.
In the next section, we have presented an approach based on matrix optimization to compute the minimizer $\mathbf{A}$, circumventing the Moore-Penrose pseudoinverse given by SVD. The proposed approach is presented based on the assumption that we can compute the regular matrix inverse $\mathbf{X1}^{-1}$, even for non square matrices. This methodology is based on the assumption that we can construct a full rank data matrix $\mathbf{X1}$ and in view of some of the drawbacks to relying on the pseudoinverse if the regular matrix inverse can exist, these include: loss of exactness, increased computational costs by requiring SVD. The pseudoinverse provides a least squares solution while the regular inverse provides a unique solution, the pseudoinverse tend to be numerically unstable for matrices with a near singularity and can amplify errors while the regular inverse is generally more stable.

5.3 Matrix optimization

In this approach, we have considered the concept and application of Lagrangian multipliers in perspective of matrix optimization. Lagrangian multipliers are common tools used in optimization problems. We have utilised the principals of Lagrange multipliers in our optimization problem, integrating the constraints into the objective function. We have applied matrix optimization firstly to the standard linear approximation (5.6), then to our proposed linear approximation in Section 5.4
We have presented the constrained matrix optimization (minimization) problem as follows:

$$\min_{\mathbf{A}} \|\mathbf{A}\|_F^2 \quad \text{subject to} \quad \mathbf{X2} - \mathbf{AX1} = 0, \tag{5.11}$$

where the objective is minimizing the Frobenius norm $\|\mathbf{A}\|_F^2 = Tr(\mathbf{A}\mathbf{A}^*)$, which equates to the Trace operator $Tr(\cdot)$. Based on the constraint $\mathbf{X2} - \mathbf{AX1} = 0$, the matrix $\mathbf{A}$ should satisfy $\mathbf{X2} = \mathbf{AX1}$.
The concept of Lagrange multipliers transforms (5.11) to an unconstrained matrix minimization problem. We have defined a Lagrangian function $l(\mathbf{A}, \mathcal{L})$:

$$l(\mathbf{A}, \mathcal{L}) = Tr(\mathbf{A}\mathbf{A}^*) + Tr(\mathcal{L}(\mathbf{AX1} - \mathbf{X2})^*), \tag{5.12}$$

where $\mathcal{L} \in \mathbb{R}^{d \times m}$ is the Lagrangian multiplier matrix, which is implemented to penalize deviations from the linear constraint. In particular, the Lagrange multiplier incorporates the constraint into the objective function. According to (4.11), then (5.12) becomes:

$$\begin{aligned} l &= \langle \mathbf{A}, \mathbf{A} \rangle - \langle \mathcal{L}, \mathbf{X2} \rangle + \langle \mathcal{L}, (\mathbf{AX1}) \rangle, \\ &= Tr(\mathbf{A}\mathbf{A}^*) - Tr(\mathcal{L}\mathbf{X2}^*) + Tr(\mathcal{L}(\mathbf{AX1})^*), \\ &= Tr(\mathbf{A}\mathbf{A}^*) - Tr(\mathcal{L}\mathbf{X2}^*) + Tr(\mathcal{L}(\mathbf{X1}^*\mathbf{A}^*)). \end{aligned}$$

The Lagrangian function is the objective function for the unconstrained optimization problem. For the purpose of optimization, the Lagrangian function is given in terms of the Frobenius norm or the trace of a matrix. Thus, the derivative of the Lagrangian function with respect to a matrix is equivalent to the derivative of the Frobenius norm of a matrix with respect to a matrix.
For example, consider $f(\mathbf{P}) = \|\mathbf{P}\|_F^2$, then:

$$\frac{\partial f}{\partial \mathbf{P}} = \frac{\partial \|\mathbf{P}\|_F^2}{\partial \mathbf{P}} = 2\mathbf{P},$$

generally, consider $f(\mathbf{P}, \mathbf{D}) = \|\mathbf{P} - \mathbf{D}\|_F^2$, then:

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{D}} &= \frac{\partial \|\mathbf{P} - \mathbf{D}\|_F^2}{\partial \mathbf{D}}, \\ &= \frac{\partial Tr\left((\mathbf{P} - \mathbf{D})(\mathbf{P} - \mathbf{D})^*\right)}{\partial \mathbf{D}}, \\ &= \frac{\partial [Tr(\mathbf{P}\mathbf{P}^*) - 2Tr(\mathbf{D}\mathbf{P}^*) + Tr(\mathbf{D}\mathbf{D}^*)]}{\partial \mathbf{D}}, \\ &= -2(\mathbf{P} - \mathbf{D}), \end{aligned}$$

where f is a function of matrices $\mathbf{P}$ and $\mathbf{D}$ given in terms of the Frobenis norm of the matrices. The gradient of the Frobenius norm gives a direction that minimizes the difference between the matrices.
Thus, we have applied first order optimality conditions to minimize the matrix function (5.12). Taking the derivative of $l(\mathbf{A}, \mathcal{L})$ with respect to $\mathbf{A}$ and $\mathcal{L}$ independently, we obtain a minimization of the Lagrangian function at the stationary points:

$$\frac{\partial l}{\partial \mathbf{A}} = 2\mathbf{A} + \mathcal{L}\mathbf{X1}^* = 0, \tag{5.13}$$

$$\frac{\partial l}{\partial \mathcal{L}} = -\mathbf{X2} + \mathbf{AX1} = 0. \tag{5.14}$$

By considering (5.13), we have $\mathbf{A} = -\frac{1}{2}\mathcal{L}\mathbf{X1}^*$ which is substituted in (5.14) to solve for $\mathcal{L}$. We find that $\mathcal{L} = -2\mathbf{X2}(\mathbf{X1}^*\mathbf{X1})^{-1}$. The L^2 norm or least squares solution to the optimization problem (5.11) is found by substituting the equation for $\mathcal{L}$ into (5.13):

$$\mathbf{A} = \mathbf{X2}(\mathbf{X1}^*\mathbf{X1})^{-1}\mathbf{X1}^*. \tag{5.15}$$

Thus, comparing the result (5.15) to (5.10), we have an explicit expression for the regular inverse which is apprximated by the Moore-Penrose pseudoinverse:

$$(\mathbf{X1}^*\mathbf{X1})^{-1}\mathbf{X1}^* = \mathbf{X1}^{-1}, \tag{5.16}$$

where $\mathbf{X1}$ is non-square, $(\mathbf{X1}^*\mathbf{X1})^{-1}$ exists assuming that $\mathbf{X1}^*\mathbf{X1}$ is invertible. The matrix product $\mathbf{X1}^*\mathbf{X1}$ is invertible if and only if the rank $r = m$, full rank and m is the number of columns. Otherwise, the matrix product (square matrix) $\mathbf{X1}^*\mathbf{X1}$ is singular and we resort to using the Moore-Penrose pseudoinverse. We have decomposed $(\mathbf{X1}^*\mathbf{X1})^{-1}\mathbf{X1}^*$ using SVD of the input data matrix $\mathbf{X1}$. The right hand side of (5.15) becomes:

$$
\begin{aligned}
\mathbf{A} &= \mathbf{X2}((\mathbf{U}\Sigma\mathbf{V}^*)^*\mathbf{U}\Sigma\mathbf{V}^*)^{-1}(\mathbf{U}\Sigma\mathbf{V}^*)^*, \\
&= \mathbf{X2}(\mathbf{V}\Sigma^*\mathbf{U}^*\mathbf{U}\Sigma\mathbf{V}^*)^{-1}(\mathbf{U}\Sigma\mathbf{V}^*)^*, \\
&= \mathbf{X2}((\mathbf{V}^*)^{-1}(\Sigma^*\Sigma)^{-1}\mathbf{V}^{-1})(\mathbf{U}\Sigma\mathbf{V}^*)^*, \\
&= \mathbf{X2}((\mathbf{V}^*)^\dagger(\Sigma^*\Sigma)^{-1}\mathbf{V}^\dagger)(\mathbf{U}\Sigma\mathbf{V}^*)^*, \\
&= \mathbf{X2}((\mathbf{V}^*\mathbf{V})^{-1}\mathbf{V}^*)^*(\Sigma^*\Sigma)^{-1}(\mathbf{V}^*\mathbf{V})^{-1}\mathbf{V}^*(\mathbf{U}\Sigma\mathbf{V}^*)^* \quad \text{using} \quad (\mathbf{V}^*)^\dagger = (\mathbf{V}^\dagger)^*, \\
&= \mathbf{X2}((\mathbf{V}^*)^*(\Sigma^*\Sigma)^{-1})\mathbf{V}^*(\mathbf{U}\Sigma\mathbf{V}^*)^*, \\
&= \mathbf{X2}(\mathbf{V}(\Sigma^*\Sigma)^{-1}\mathbf{V}^*)(\mathbf{U}\Sigma\mathbf{V}^*)^*, \\
&= \mathbf{X2V}(\Sigma^*\Sigma)^{-1}\mathbf{V}^*\mathbf{V}\Sigma^*\mathbf{U}^*, \\
&= \mathbf{X2V}(\Sigma^*\Sigma)^{-1}\Sigma^*\mathbf{U}^*.
\end{aligned}
$$

Therefore, we have the same expression for $\mathbf{A}$, when we apply SVD of $\mathbf{X1}$ on the regular inverse given in (5.16), as the one obtained when we apply a pseudoinverse $\mathbf{X1}^\dagger$. Thus the SVD for the regular inverse is given by $\mathbf{X1}^{-1} = \mathbf{V}\Sigma^{-1}\mathbf{U}^*$ and (5.15) becomes:

$$\mathbf{A} = \mathbf{X2V}\Sigma^{-1}\mathbf{U}^* \approx \mathbf{X2}\hat{\mathbf{V}}_r\hat{\Sigma}_r^{-1}\mathbf{U}_r^*. \tag{5.17}$$

The result (5.17) are projected onto the leading r, when $r < m$, POD modes in $\mathbf{U}_r^*$ to give a reduced dimension approximation of $\mathbf{A}$. In essence, capturing only the first r eigenvalues and eigenvectors of (5.15). By multiplying (5.17) by the transpose of $\hat{\mathbf{U}}_r$ from the left and $\hat{\mathbf{U}}_r$ from right, we obtain the result:

$$\tilde{\mathbf{A}} = \hat{\mathbf{U}}_r^*\mathbf{A}\hat{\mathbf{U}}_r = \hat{\mathbf{U}}_r^*\mathbf{X2}\hat{\mathbf{V}}_r\hat{\Sigma}_r^{-1}\hat{\mathbf{U}}_r^*\hat{\mathbf{U}}_r = \hat{\mathbf{U}}_r^*\mathbf{X2}\hat{\mathbf{V}}_r\hat{\Sigma}_r^{-1}, \tag{5.18}$$

where the smaller-dimensional matrix $\tilde{\mathbf{A}}$ is an approximation of the full matrix $\mathbf{A}$. The leading eigendecomposition of $\mathbf{A}$ is now approximated by the eigendecomposition of $\tilde{\mathbf{A}}$ [18]. The eigenvalue problem of the reduced matrix (5.18) is given by:

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}. \tag{5.19}$$

The eigendecomposition of $\tilde{\mathbf{A}}$ provides with DMD eigenvalues stored in the diagonal matrix $\mathbf{\Lambda}$. These DMD eigenvalues correspond to eigenvalues of the finite

dimensional Koopman operator, in this case $\mathbf{A}$. The columns in $\mathbf{W}$ are eigenvectors of small $\tilde{\mathbf{A}}$ where each column is a linear combination of how much each POD mode in the column (a mode amplitude) contributes to the overall system. These linearly combined mode amplitudes have a linear behaviour exhibited by a single temporal pattern, given by a corresponding eigenvalue.
Recall that the DMD modes, denoted Φ, approximate the modes of the Koopman operator or the eigenvectors of $\mathbf{A}$. Φ is a reconstruction using the matrix $\mathbf{X2}$ and the matrix $\mathbf{W}$:

$$\mathbf{\Phi} = \mathbf{X2}\hat{\mathbf{V}}_r\hat{\Sigma}_r^{-1}\mathbf{W}. \tag{5.20}$$

The DMD modes (5.21) are the leading columns for the eigenvector matrix of $\mathbf{A}$, corresponding to the leading eigenvalues in $\mathbf{\Lambda}$. In [18], it is shown that the Φ are eigenvectors of the matrix $\mathbf{A}$ under certain conditions:

$$\begin{aligned}
\mathbf{A\Phi} &= (\mathbf{X2}\hat{\mathbf{V}}_r\hat{\Sigma}_r^{-1}\hat{\mathbf{U}}_r^*)(\mathbf{X2}\hat{\mathbf{V}}_r\hat{\Sigma}_r^{-1}\mathbf{W}) \quad \text{using (5.17) and (5.20)},\\
&= \mathbf{X2}\hat{\mathbf{V}}_r\hat{\Sigma}_r^{-1}\tilde{\mathbf{A}}\mathbf{W} \quad \text{using (5.18)},\\
&= \mathbf{X2}\hat{\mathbf{V}}_r\hat{\Sigma}_r^{-1}\mathbf{W}\Lambda \quad \text{using (5.19)},\\
&= \mathbf{\Phi}\Lambda \quad \text{using (5.20)}.
\end{aligned}$$

The next section, we have presented the DMD spectral decomposition's utility in representing the data snapshots.

5.3.1 DMD eigen expansion

Having constructed the smaller dimension matrix from (5.18), the DMD eigenvalues from (5.19) and the DMD modes from (5.20), the DMD framework represents the measurements of the state of a dynamical system in terms of the DMD eigenquantities.
For the discrete time dynamics, the snapshot $\mathbf{x}_k$ in terms of the DMD eigen expansion is given as:

$$\mathbf{x}_k = \sum_{j=1}^{r} \boldsymbol{\phi}_j \lambda_j^{k-1} b_j = \mathbf{\Phi}\mathbf{\Lambda}^{k-1}\mathbf{b}, \tag{5.21}$$

where the columns $\boldsymbol{\phi}_j \in \mathbf{\Phi}$ are the DMD modes or eigenvectors of $\mathbf{A}$, $\lambda_j \in \mathbf{\Lambda}$ are DMD eigenvalues or eigenvalues of $\mathbf{A}$ and $b_j \in \mathbf{b}$ are DMD mode amplitudes, a finite sequence of triples $\{\boldsymbol{\phi}_j, \lambda_j, b_j\}_{j=1}^{r}$ constituting the DMD. DMD modes, which

are the spatial structures or patterns within the data matrix with each mode being characterized by coherent patterns having a specific temporal behaviour. The temporal behaviour of a DMD mode is given by its corresponding DMD eigenvalue. The contribution of each DMD mode to the overall dynamics is prescribed by an associated DMD mode amplitude. The DMD expansion (5.21) is similar in a sense (analogous) to the Koopman mode decomposition (3.30). We recall that, DMD approximates Koopman mode decomposition. Therefore, DMD mode amplitudes approximate the Koopman eigenfunctions evaluated at $\varphi(\mathbf{x}_1)$, initial state, DMD modes approximate the Koopman modes and DMD eigenvalues approximate the Koopman eigenvalues. To demonstrate the DMD approximation of the Koopman mode decomposition, the reader is reminded that we are considering direct measurements of the state, $g(\mathbf{x}(t)) = \mathbf{x}(t)$ and the associated Koopman mode decomposition. Thus, the Koopman mode decomposition (3.28) with direct measurements of the state is finitely approximated by DMD (5.21). The matrix representations are given as follows:

$$\mathbf{x}_k \approx \sum_{j=1}^{r} \lambda_j^{k-1} \varphi_j(\mathbf{x}_1)\mathbf{v}_j, \tag{5.22}$$

$$= \underbrace{\begin{bmatrix}\mathbf{v}_1 & \dots & \mathbf{v}_r\end{bmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{bmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_r\end{bmatrix}}_{\mathbf{\Lambda}} \underbrace{\begin{bmatrix}\varphi_1(\mathbf{x}_1) \\ \vdots \\ \varphi_r(\mathbf{x}_1).\end{bmatrix}}_{\mathbf{b}} \tag{5.23}$$

$$= \underbrace{\begin{bmatrix}\mathbf{v}_1 & \dots & \mathbf{v}_r\end{bmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{bmatrix}\varphi_1(\mathbf{x}_1) & & \\ & \ddots & \\ & & \varphi_r(\mathbf{x}_1)\end{bmatrix}}_{\mathbf{b}} \underbrace{\begin{bmatrix}\lambda_1(\mathbf{x}_1) \\ \vdots \\ \lambda_r(\mathbf{x}_1).\end{bmatrix}}_{\mathbf{\Lambda}}. \tag{5.24}$$

Thus, each column (snapshot) of the data matrix (5.7) can be represented by the Koopman mode decomposition (5.24). Using DMD, then (5.7) is given as:

$$\mathbf{X1} = \begin{bmatrix}\boldsymbol{\phi}_1 & \dots & \boldsymbol{\phi}_r\end{bmatrix} \begin{bmatrix}b_1 & & \\ & \ddots & \\ & & b_r\end{bmatrix} \begin{bmatrix}\lambda_1 & \dots & \lambda_1^{m-1} \\ \vdots & \ddots & \vdots \\ \lambda_r & \dots & \lambda_r^{m-1}.\end{bmatrix} \tag{5.25}$$

For the continuous-time dynamics, a transformation is introduced to compute the continuous eigenvalues, $\omega = \log(\lambda)/\Delta t$. Thus (5.21) becomes:

$$\mathbf{x}(t) = \sum_{j=1}^{r} \boldsymbol{\phi}_j exp(\omega_j t) b_j = \boldsymbol{\Phi} exp(\boldsymbol{\Omega} t)\mathbf{b} \tag{5.26}$$

here the continuous eigenvalues ω_j are contained in the diagonal matrix given by $exp(\boldsymbol{\Omega} t)$ and (5.25) in continuous time dynamics is given as:

$$\mathbf{X1} \approx \begin{bmatrix} \boldsymbol{\phi}_1 & \dots & \boldsymbol{\phi}_r \end{bmatrix} \begin{bmatrix} b_1 & & \\ & \ddots & \\ & & b_r \end{bmatrix} \begin{bmatrix} e^{(\omega_1 t_1)} & \dots & e^{(\omega_1 t_m)} \\ \vdots & \ddots & \vdots \\ e^{(\omega_1 t_1)} & \dots & e^{(\omega_1 t_m)} \end{bmatrix}. \tag{5.27}$$

The vector $\mathbf{b}$ contains the DMD mode amplitudes b_j, each corresponds to a DMD mode. The vector $\mathbf{b}$ is calculated using the first snapshot. Using the first snapshot quantifies each DMD mode's overall contribution to the dynamics, computed as:

$$\mathbf{b} = \boldsymbol{\Phi}^{\dagger}\mathbf{x}_1, \tag{5.28}$$

where the first DMD snapshot $\mathbf{x}_1$ is equivalent to the initial condition $\mathbf{x}(0)$ used in (3.30) for the Koopman eigenfunction in the Koopman mode decomposition.
So far, modern DMD seeks a best fit linear operator $\mathbf{A}$ based on $\mathbf{X2} = \mathbf{AX1}$. The classical relation imply that $\mathbf{X2}$ entirely relates to $\mathbf{X1}$ via the linear operator $\mathbf{A}$. Thus, DMD constructs a homogeneous linear ordinary differential equation to model the evolution of nonlinear dynamics. The solution (state) forms, (5.21) for the discrete or (5.26) for the continuous time dynamics, stems from linearity, where the superposition principle allowing the solution to be constructed as a weighted sum of the system's dynamic modes. Each term in the sum represents an independent "mode" of the system's behavior, scaled by the initial condition b_j and evolved in time according to $exp(\omega_j t)$. This superposition captures the nonlinear system's dynamics through a linear combination of evolving modes, providing a computationally feasible approximation of nonlinear behavior.
However, this approach may fall short when dealing with systems that cannot be fully represented by a single linear operator $\mathbf{A}$. In such cases, a nonhomogeneous

linear ordinary differential equation might provide a more suitable model, accounting for external influences or inputs that **A** alone does not capture. The following section introduces this proposed nonhomogeneous DMD framework and presents the mathematical foundation for its formulation. This new approach aims to address the limitations of traditional DMD and extend its applicability to a broader class of nonlinear systems.

5.4 Modified DMD

While DMD framework depends on **A** to characterise the dynamical system, it is an approximation and may not fully account for all possible behaviours. Thus, **A** may perform poorly or predict inaccurately in cases where the system exhibits significant nonlinearities or complex dynamics beyond the scope of linear homogenous approximation. Real world dynamical systems often exhibit behaviors far from the origin (baseline or offset behaviours). However, the classical linear approximation unrealistically assumes that **X2** always cuts through the origin whenever $\mathbf{X1} = 0$, which may fail to capture such offset dynamics.
The linear approximation $\mathbf{X2} = \mathbf{AX1}$ is sensitive to noisy data sets. Even small amounts of noisy inputs can significantly alter **A**, leading to poor model performance. For systems with strong nonlinear interactions, the classical linear approximation may over simplify the underlying dynamics. In addition, the matrix **A** may fail to meaningfully prescribe the data matrices, thus compromising its usefulness to capture and interpret the behaviour of the dynamical system.
Modern DMD also faces challenges in dealing with high dimensional data sets. Thus, leading to inaccuracies due to its inability to fully capture complex interactions across multiple dimensions. Thereby, hindering the model's ability to generalize. Additionally, obtaining the solution **A** is often challenging when the data matrices are not well scaled, further reducing the effectiveness of the DMD's framework.
Furthermore, the model $\mathbf{X2} = \mathbf{AX1}$ excludes the explicit addition of a regularizer, which could control the complexity of the model. Without a regularization term, the model may produce unstable or overly sensitive results relative to the data set, limiting its robustness and reliability in real-world applications.

In view of some drawbacks from applying DMD to compute $\mathbf{X2} = \mathbf{AX1}$, we have proposed an alternative formulation: applying DMD to compute $\mathbf{X2} = \mathbf{AX1} + \mathbf{B}$. The matrix $\mathbf{B}$ is regarded as a bias or offset matrix, allows the model to better capture offset dynamics and baseline shifts that are common in real world dynamical systems, offering a more accurate and flexible representation of the system's dynamics.

In addition, the matrix $\mathbf{B}$ aids in capturing intricate dynamics that may have been overlooked by only depending on $\mathbf{A}$. Thus, $\mathbf{B}$ acts as a correction to the strictly linear model, improving the model's ability to describe more complex system dynamics that a purely linear approximation might overlook. The proposed linear approximation $\mathbf{X2} = \mathbf{AX1} + \mathbf{B}$ has shown an improved linear fit to the data with reduced residual errors in the linear approximation of the dynamical systems tested in Chapter 6.

We have computed the finite dimensional approximation of the Koopman operator based on the proposed linear dynamic model: $\mathbf{X2} = \mathbf{AX1} + \mathbf{B}$. The proposed linear dynamic model is compared to the conventional $\mathbf{X2} = \mathbf{AX1}$ within the DMD framework. Our novel approach presented in this chapter, which we have called modified DMD (mDMD), is formulated from this perspective. To distinguish the new computed finite dimensional Koopman operator from the conventional DMD operator $\mathbf{A}$, we denote the new linear operator $\mathbf{A}_m$ so that the proposed model is given as $\mathbf{X2} = \mathbf{A}_m\mathbf{X1} + \mathbf{B}$. This distinction emphasizes the enhanced flexibility and accuracy introduced by mDMD compared to the conventional DMD approach.

We have appreciated the flexibility and improved approximations from the proposed linear approximation, based on the test cases. However, our further studies explore the computational costs arising from the added bias matrix $\mathbf{B}$. Furthermore, careful considerations of the dynamic system and data type are explored to validate the addition of $\mathbf{B}$ for the DMD modeling framework.

5.4.1 Computation of A and B

We have considered the data matrices $\mathbf{X1}$ from (5.7) and $\mathbf{X2}$ from (5.8). The following linear relation attempts to provide a best fit linear approximation:

$$\mathbf{A}_m\mathbf{X1} + \mathbf{B} = \mathbf{X2}, \tag{5.29}$$

where $\mathbf{A}_m \in \mathbb{R}^{d\times d}$ and $\mathbf{B} \in \mathbb{R}^{d\times m}$ are unknown, $\mathbf{X1} \in \mathbb{R}^{d\times m}$, $\mathbf{X2} \in \mathbb{R}^{d\times m}$. The proposed modified DMD aims to minimize the Frobenius norm $\|\mathbf{A}_m\mathbf{X1} + \mathbf{B} - \mathbf{X2}\|_F$, providing optimal L^2 norm solutions for both $\mathbf{A}_m$ and $\mathbf{B}$. The Frobenius norm evaluates the overall data matrices, considering all elements in the data matrix. Thus, the solutions computed do not promote sparsity in the matrices.
We have presented the constrained matrix optimization (minimization) problem as follows:

$$\min_{\mathbf{A}_m,\mathbf{B}} \|\mathbf{A}_m\|_F^2 + \|\mathbf{B}\|_F^2 \quad \text{subject to} \quad \mathbf{X2} - \mathbf{A}_m\mathbf{X1} - \mathbf{B} = 0, \tag{5.30}$$

where the objective is minimizing the Frobenius norms $\|\mathbf{A}_m\|_F^2 = Tr(\mathbf{A}_m\mathbf{A}_m^*)$ and $\|\mathbf{B}\|_F^2 = Tr(\mathbf{B}\mathbf{B}^*)$. Based on the constraint $\mathbf{X2} - \mathbf{A}_m\mathbf{X1} - \mathbf{B} = 0$, the matrices $\mathbf{A}_m$ and $\mathbf{B}$ should satisfy $\mathbf{X2} = \mathbf{A}_m\mathbf{X1} + \mathbf{B}$.
To solve for $\mathbf{A}_m$ and $\mathbf{B}$, the optimization problem (5.30) is represented using the Lagrangian function, denoted $l(\mathbf{A}_m, \mathbf{B}, \mathcal{L})$:

$$l(\mathbf{A}_m, \mathbf{B}, \mathcal{L}) = Tr(\mathbf{A}_m\mathbf{A}_m^*) + Tr(\mathbf{B}\mathbf{B}^*) + Tr(\mathcal{L}(\mathbf{X2} - \mathbf{A}_m\mathbf{X1} - \mathbf{B})^*), \tag{5.31}$$

where the Lagrangian formulation incorporates both the objective and constraint function which are given in terms the Frobenius norm, providing a pathway to finding optimal values for the matrices $\mathbf{A}_m$ and $\mathbf{B}$. Using (4.11), then (5.31) becomes:

$$\begin{aligned} l &= \langle \mathbf{A}_m, \mathbf{A}_m\rangle + \langle \mathbf{B}, \mathbf{B}\rangle + \langle \mathcal{L}, \mathbf{X2}\rangle - \langle \mathcal{L}, \mathbf{A_m X1}\rangle - \langle \mathcal{L}, \mathbf{B}\rangle, \\ &= Tr(\mathbf{A}_m\mathbf{A}_m^*) + Tr(\mathbf{B}\mathbf{B}^*) + Tr(\mathcal{L}\mathbf{X2}^*) - Tr(\mathcal{L}(\mathbf{A_m X1})^*) - Tr(\mathcal{L}\mathbf{B}^*), \\ &= Tr(\mathbf{A}_m\mathbf{A}_m^*) + Tr(\mathbf{B}\mathbf{B}^*) + Tr(\mathcal{L}\mathbf{X2}^*) - Tr(\mathcal{L}(\mathbf{X1}^*\mathbf{A_m^*})) - Tr(\mathcal{L}\mathbf{B}^*). \end{aligned}$$

To solve a minimization problem (5.31), we have applied first order optimality conditions. The first order optimality conditions are derived by taking the derivative of $l(\mathbf{A}_m, \mathbf{B}, \mathcal{L})$ with respect to the matrices $\mathbf{A}_m$, $\mathbf{B}$ and the Lagrangian multiplier $\mathcal{L}$. The derivative with respect to a matrix follows the principles of matrix calculus. In this optimization problem, we have incorporated the Frobenius norm gradient to minimize the objective function. In particular, the goal is to find the stationary points that minimize the Lagrangian function utilizing its representation in terms

of the Frobenius norm, often represented as the trace, $Tr(\cdot)$:

$$\frac{\partial l}{\partial \mathbf{A}_m} = 2\mathbf{A}_m - \mathcal{L}\mathbf{X1}^* = 0, \tag{5.32}$$

$$\frac{\partial l}{\partial \mathbf{B}} = 2\mathbf{B} - \mathcal{L} = 0, \tag{5.33}$$

$$\frac{\partial l}{\partial \mathcal{L}} = \mathbf{X2} - \mathbf{A}_m\mathbf{X1} - \mathbf{B} = 0. \tag{5.34}$$

Considering (5.33), we have that $2\mathbf{B} = \mathcal{L}$ which can be substituted for $\mathcal{L}$ in (5.32). This results in $\mathbf{A}_m = \mathbf{BX1}^*$. Substituting the results from manipulating (5.32) and (5.33) into (5.34), we have obtained:

$$\mathbf{A}_m = \mathbf{X2}(\mathbf{X1}^*\mathbf{X1} + \mathbb{I})^{-1}\mathbf{X1}^*, \tag{5.35}$$

$$\mathbf{B} = \mathbf{X2}(\mathbf{X1}^*\mathbf{X1} + \mathbb{I})^{-1}. \tag{5.36}$$

where $\mathbb{I} \in \mathbb{R}^{m\times m}$ denotes the identity matrix. Combining (5.35) and (5.36), we obtain:

$$\mathbf{A}_m = \mathbf{X2}(\mathbf{X1}^*\mathbf{X1} + \mathbb{I})^{-1}\mathbf{X1}^* = \mathbf{BX1}^*. \tag{5.37}$$

Given that $\mathbf{X1} \in \mathbb{R}^{d\times m}$, with $d > m$, the matrix $(\mathbf{X1}^*\mathbf{X1} + \mathbb{I}) \in \mathbb{R}^{m\times m}$. The resulting identity matrix $\mathbb{I}$ in the formulation ensures that all eigenvalues are shifted by at least 1, making the matrix product strictly positive definite and invertible. In contrast, $(\mathbf{X1}^*\mathbf{X1})$ on its own is positive semi-definite and may not be invertible, depending on the rank of $\mathbf{X1}$ as assumed in Section 5.3. However, assuming $\mathbf{X1}$ is not full rank, $(\mathbf{X1}^*\mathbf{X1})$ will have non negative eigenvalues, with some eigenvalues equal to zero due to its positive semi-definiteness. So, the presence of the the identity matrix to the matrix product ensures that eigenvalues of $(\mathbf{X1}^*\mathbf{X1} + \mathbb{I})$ are strictly positive (nonzero). Thus, the result $(\mathbf{X1}^*\mathbf{X1} + \mathbb{I})$ is invertible since it is always positive definite.

To compute the inverse $(\mathbf{X1}^*\mathbf{X1} + \mathbb{I})^{-1}$, we have considered the full SVD of $\mathbf{X1}$. The SVD approach is numerically stable and robust, especially when $\mathbf{X1}$ is not well conditioned or without full rank. A more accurate reconstruction of the system is guaranteed since all the singular values and corresponding modes of the system are returned in the full SVD. By accounting for all the modes, even those associated with small or zero singular values. Thus, the SVD captures the complete structure of the data matrix, leading to better numerical performance and a more reliable

solution. We have let $\mathbf{Z} = (\mathbf{X1}^*\mathbf{X1} + \mathbb{I})$, so that:

$$\begin{aligned}\mathbf{Z} = ((SVD(\mathbf{X1}))^* SVD(\mathbf{X1}) + \mathbb{I}) &= ((\mathbf{U}\Sigma\mathbf{V}^*)^*(\mathbf{U}\Sigma\mathbf{V}^*) + \mathbb{I}),\\ &= (\mathbf{V}\Sigma^*\mathbf{U}^*\mathbf{U}\Sigma\mathbf{V}^* + \mathbb{I}).\end{aligned}$$

The columns of $\mathbf{U}$ are orthonomal, thus the SVD gives the result:

$$\mathbf{Z} = (\mathbf{V}\Sigma^*\Sigma\mathbf{V}^* + \mathbb{I}). \tag{5.38}$$

By substituting (5.38) into (5.37), we obtain a SVD factorization of (5.37):

$$\begin{aligned}\mathbf{A}_m &= \mathbf{X2}(\mathbf{V}\Sigma^*\Sigma\mathbf{V}^* + \mathbb{I})^{-1}(SVD(\mathbf{X1}))^*,\\ &= \mathbf{X2}(\mathbf{V}\Sigma^*\Sigma\mathbf{V}^* + \mathbb{I})^{-1}(\mathbf{U}\Sigma\mathbf{V}^*)^*.\end{aligned}$$

With further simplifications, we get:

$$\begin{aligned}\mathbf{A}_m &= \mathbf{X2}(\mathbf{V}\Sigma^*\Sigma\mathbf{V}^* + \mathbb{I})^{-1}(\mathbf{U}\Sigma\mathbf{V}^*)^*,\\ &= \mathbf{X2}(\mathbf{V}\Sigma^*\Sigma\mathbf{V}^* + \mathbf{V}\mathbf{V}^*)^{-1}(\mathbf{U}\Sigma\mathbf{V}^*)^* \quad \text{orthogonal square matrix} \quad \mathbf{V}\mathbf{V}^* = \mathbf{V}^*\mathbf{V} = \mathbb{I},\\ &= \mathbf{X2}(\mathbf{V}(\Sigma^*\Sigma + \mathbf{I})\mathbf{V}^*)^{-1}(\mathbf{U}\Sigma\mathbf{V}^*)^*\\ &= \mathbf{X2}((\mathbf{V}^*)^{-1}(\Sigma^*\Sigma + \mathbf{I})^{-1}(\mathbf{V})^{-1})(\mathbf{U}\Sigma\mathbf{V}^*)^*,\\ &= \mathbf{X2}((\mathbf{V}^*)^*(\Sigma^*\Sigma + \mathbf{I})^{-1})\mathbf{V}^*\mathbf{V}\Sigma^*\mathbf{U}^* \quad \text{orthogonal square matrix} \quad \mathbf{V}^* = \mathbf{V}^{-1},\\ &= \mathbf{X2}(\mathbf{V}(\Sigma^*\Sigma + \mathbf{I})^{-1})\Sigma^*\mathbf{U}^*,\\ &= \mathbf{X2}\mathbf{V}(\Sigma^*\Sigma + \mathbb{I})^{-1}\Sigma^*\mathbf{U}^*.\end{aligned}$$

Therefore, we have:

$$\mathbf{A}_m = \mathbf{X2}\mathbf{V}(\Sigma^*\Sigma + \mathbb{I})^{-1}\Sigma^*\mathbf{U}^*, \tag{5.39}$$

$$\approx \mathbf{X2}\hat{\mathbf{V}}_r(\hat{\Sigma}_r^*\hat{\Sigma}_r + \mathbb{I}_r)^{-1}\hat{\Sigma}_r^*\hat{\mathbf{U}}_r^*, \tag{5.40}$$

where $\mathbf{A}_m \in \mathbb{R}^{d\times d}$ and $\mathbf{U}_r$, Σ_r and $\mathbf{V}_r$ represent the truncated SVD components, keeping only the first r singular values and vectors. The approximation sign in (5.40) follows from the usage of the truncated SVD. The matrix (5.40) is projected onto the leading r POD modes in $\mathbf{U}_r^*$, giving a reduced dimension approximation

of $\mathbf{A}_m$. This projection guarantees that only the most dynamic modes, which capture the dominant dynamics, are used in the approximation of $\mathbf{A}_m$, significantly reducing computational complexity while preserving key dynamic features of the system:

$$\begin{aligned}\tilde{\mathbf{A}}_m = \hat{\mathbf{U}}_r^* \mathbf{A}_m \hat{\mathbf{U}}_r &= \hat{\mathbf{U}}_r^* \mathbf{X2} \hat{\mathbf{V}}_r (\hat{\Sigma}_r^* \hat{\Sigma}_r + \mathbb{I})^{-1} \hat{\Sigma}_r^* \hat{\mathbf{U}}_r^* \hat{\mathbf{U}}_r, && (5.41)\\ &= \hat{\mathbf{U}}_r^* \mathbf{X2} \hat{\mathbf{V}}_r (\hat{\Sigma}_r^* \hat{\Sigma}_r + \mathbb{I})^{-1} \hat{\Sigma}_r^*, && (5.42)\end{aligned}$$

were $\tilde{\mathbf{A}}_m$ is the reduced dimension approximation of $\mathbf{A}_m$. The eigenvalue problem for the reduced dimension matrix is formulated as follows:

$$\tilde{\mathbf{A}}_m \mathbf{W} = \mathbf{W} \mathbf{\Lambda}, \tag{5.43}$$

where the columns of the matrix $\mathbf{W}$ are the eigenvectors while the diagonal entries of the matrix $\mathbf{\Lambda}$ are the eigenvalues of $\tilde{\mathbf{A}}_m$. Similar to the process outlined for DMD modes in Section 5.3, the mDMD modes, denoted as $\mathbf{\Phi}_m$, are computed using $\mathbf{X2}$ and the matrix $\mathbf{W}$ from (5.43). In particular, $\mathbf{\Phi}_m$ are given as:

$$\mathbf{\Phi}_m = \mathbf{X2} \hat{\mathbf{V}}_r (\hat{\Sigma}_r^* \hat{\Sigma}_r + \mathbb{I})^{-1} \hat{\Sigma}_r^* \mathbf{W}. \tag{5.44}$$

This formulation allows us to extract the dominant mDMD modes while incorporating the regularized offset dynamics captured by $\mathbf{B}$, resulting in a more flexible characterization of the system's behavior compared to traditional DMD as shown in our test cases. As in Section 5.3 for DMD modes, we now extend the process to modified DMD (mDMD). Specifically, the mDMD modes, denoted as Φ_m are eigenvectors of the matrix $\mathbf{A}_m$ as demonstrated by the following derivation:

$$\begin{aligned}\mathbf{A}_m \mathbf{\Phi}_m &= \mathbf{X2} \hat{\mathbf{V}}_r (\hat{\Sigma}_r^* \hat{\Sigma}_r + \mathbb{I})^{-1} \hat{\Sigma}_r^* \hat{\mathbf{U}}_r^* \mathbf{X2} \hat{\mathbf{V}}_r (\hat{\Sigma}_r^* \hat{\Sigma}_r + \mathbb{I})^{-1} \hat{\Sigma}_r^* \mathbf{W} \quad (5.40),\\ &= \mathbf{X2} \hat{\mathbf{V}}_r (\hat{\Sigma}_r^* \hat{\Sigma}_r + \mathbb{I})^{-1} \hat{\Sigma}_r^* \tilde{\mathbf{A}}_m \mathbf{W} \quad (5.42),\\ &= \mathbf{X2} \hat{\mathbf{V}}_r (\hat{\Sigma}_r^* \hat{\Sigma}_r + \mathbb{I})^{-1} \hat{\Sigma}_r^* \mathbf{W} L \mathbf{\Lambda} \quad (5.43),\\ &= \mathbf{\Phi}_m \mathbf{\Lambda} \quad (5.44).\end{aligned}$$

The proposed mDMD algorithm is similar to the DMD algorithm [18]. State estimation and prediction within the mDMD framework proceed similarly to DMD, with the spectral decomposition of the snapshot matrices in mDMD paralleling the

presentation in Section 5.3.1 for DMD. Specifically, mDMD's low-rank matrix (5.42) is analogous to DMD's low-rank matrix (5.18). The matrix (5.42) decomposes into the mDMD modes (5.44) similar to the DMD modes (5.20). Recalling that state estimation and prediction in DMD is given by (5.26), for mDMD we have used (5.44) as opposed to (5.20), so that state estimation and prediction in mDMD is given by :

$$\mathbf{x}(t) = \sum_{j=1}^{r} \boldsymbol{\phi}_{m_j} exp(\omega_j t) b_j = \boldsymbol{\Phi}_m exp(\boldsymbol{\Omega} t) \mathbf{b}_m \tag{5.45}$$

where each mode $\boldsymbol{\phi}_j$ has a dynamic term connected to the initial conditions, denoted as $exp(\omega_j)b_j$. The term $\mathbf{b}_m = \boldsymbol{\Phi}_m^{-1}\mathbf{x}_1$ is the matrix of mDMD mode amplitudes representing initial conditions. The matrix $exp(\boldsymbol{\Omega}_m t)$ is the diagonal matrix of mDMD continuous eigenvalues.

The Lagrangian formulation presented in this thesis for both the DMD and the proposed mDMD provides a more tractable and structured mathematical procedure for finding optimal solutions compared to the classical least squares optimization problem classically used in DMD, (5.10). The regularization term $\|\mathbf{A}_m\|_F^2 + \|\mathbf{B}\|_F^2$, promotes small norm solutions in high dimensional systems, which reduces sensitivity to noise in **X2** and **X1**. The small norm solutions are advantageous for computational efficiency and interpretability when modeling complex systems.

Furthermore, the Lagrangian formulation allows explicit control over balancing between minimizing the constraint violation and the regularization terms. In contrast, the Least squares optimization focuses only on minimizing the magnitude of the constraint violation without necessarily enforcing the exact equality condition intended in the linear dynamic model. Thus, the Least squares optimization typically leads to small constraint violations in the solutions.

Unlike in Least squares optimization where closed form solutions are not obtained if constraints are added as penalty terms, the Lagrangian formulation analytically provides closed form solutions by equating the partial derivatives with respect to the each variable to zero, (5.32), (5.33) and (5.34). Moreover, the Lagrangian multiplier provides a physical interpretation, the cost of involving the constraint for each entry in $\mathbf{X2} - \mathbf{A}_m\mathbf{X1} - \mathbf{B} = 0$, the Least squares sense generalizes deviations resulting from the constraint, individual constraint contributions are not captured.

The mDMD methodology is given in summary as follows:

Algorithm 1: The mDMD Algorithm

1. **Construct data matrices**

    - **X1**: Input data matrix (5.7)
    - **X2**: Output data matrix (5.8)

2. **Compute a high dimensional operator $\mathbf{A}_m$**

    - use truncated SVD of (5.7) as equation (5.40)

3. **Compute a reduced dimensional operator $\tilde{\mathbf{A}}_m$**

    - using leading POD modes as equation (5.42)

4. **Perform a spectral decomposition of $\tilde{\mathbf{A}}_m$**

    - Perform the eigenvalue decomposition of $\tilde{\mathbf{A}}_m$ as equation (5.43)

5. **Compute the mDMD modes $\mathbf{\Phi}_m$, eigenvalues $exp(\mathbf{\Omega}_m t)$ and mode amplitudes $\mathbf{b}_m$**

    - Calculate the mDMD modes from the spectral decomposition results as equation (5.44).

6. **Compute state estimation and prediction $\mathbf{x}(t)$**

    - Use the mDMD modes from step 5 to estimate the system's state and perform future predictions (equation 5.26).

Table 1: Summary of the mDMD Algorithm

Step	Description	Equation Reference
1	Construct data matrices **X1**, **X2**	(5.7), (5.8)
2	Compute $\mathbf{A}_m$ using truncated SVD	(5.40)
3	Compute $\tilde{\mathbf{A}}_m$ using leading POD modes	(5.42)
4	Spectral decomposition of $\tilde{\mathbf{A}}_m$	(5.43)
5	Compute mDMD eigenquantities $\mathbf{\Phi}_m$, $exp(\mathbf{\Omega}_m t)$, $\mathbf{b}_m$	(5.44)
6	State estimation and prediction $\mathbf{x}(t)$ using mDMD modes	(5.26)

Summary of the mDMD concepts and results

Data matrix – A matrix made up of column vectors of snapshots of the system's state. Each column vector represents a specific instant in time [4].
Dynamical system – A mathematical representation of a dynamical system. Typically, formulated in terms of ordinary differential (linear or nonlinear) equations on a state-space. The resulting equations may also include the effect of actuation inputs and represent outputs as sensor measurements of the state [4].
Hilbert space – A generalized vector space with an inner product. In this thesis, a Hilbert space typically refers to an infinite-dimensional function space. These spaces are also complete metric spaces, providing a sufficient mathematical framework to enable calculus on functions [4].
Koopman operator – An infinite-dimensional linear operator that maps measurement functions from an infinite-dimensional Hilbert space through a dynamical system [4].
Koopman eigenfunction – An eigenfunction of the Koopman operator.In our context, these eigenfunctions correspond to the direct measurements on the state-space of a dynamical system that form intrinsic coordinates. In particular, these intrinsic measurements will evolve linearly in time even when the underlying system is nonlinear [4].
Low rank – A matrix is said to be low rank when the number of linearly independent columns and rows is small compared with the size of that matrix. Generally,

when dealing with high dimensional data matrices, low-rank approximations are required [4].
Observable function – A function that measures or projects some property of the system's state. Observable functions are typically members of a Hilbert space [4].
Proper orthogonal decomposition (POD) – The simplification or breakdown of a high dimensional data matrix from a dynamical system into a hierarchical set of orthogonal modes, oftentimes, using SVD. For example, a data consisting of velocity measurements from an incompressible fluid flow system, then POD orders modes according to the amount of energy each mode contains of the given data [4].
Pseudoinverse – The pseudoinverse generalizes the regular matrix inverse for non-square matrices that are not full rank. The SVD is a common method to compute the pseudoinverse [4].
System identification – The process by which we have constructed a model for a dynamical system using measurement data [4].
mDMD modes - These are the spatial patterns or structures associated with specific dynamic behavior (linear) in the system. These modes are derived from the decomposition of data, providing insight into how different parts of the system evolve over time [15].

- Oscillatory Nature:

  Each mDMD mode is essentially a pattern or structure in space within the system, oscillating at a specific frequency. Depending on the data, the mDMD mode oscillations can exhibit either growth (increasing amplitude) or decay (decreasing amplitude) over time.

- Eigenvector Interpretation:

  The mDMD modes are similar to the eigenvectors of the system. They represent the fundamental building blocks or "modes" of the system's dynamics. In particular, the mDMD algorithm finds the best linear approximation to model the underlying complex dynamics using these modes.

- Mode Amplitude:

  Each mDMD mode corresponds to its own mode amplitude, which indicates its significance (dominance) in the overall dynamic behavior of the system. A larger amplitude implies that the mode plays a more dominant role in the

system's behavior. Conversely, a smaller amplitude means that the mode has a lesser impact on the dynamics. The dominance of the mDMD modes is reflected in how much energy or signal the mode carries in relation to the entire system. Higher-amplitude modes will be more noticeable in the dynamic evolution, while low-amplitude modes may be more subtle or even negligible.

Discrete time eigenvalues

The temporal evolution of each mDMD mode is captured by its corresponding discrete-time eigenvalue [17]. These eigenvalues are associated with the best-fit linear operator and describe the frequency and growth or decay behavior of each mode. The discrete-time eigenvalues directly relate to how the modes oscillate, grow, or decay in the system over time. The role of the discrete time eigenvalues is more clearer with the continuous time eigenvalues, given below.

Continuous time eigenvalues

In practice, discrete-time eigenvalues λ are often converted to their continuous-time equivalents ω, providing a more natural representation of the system's dynamics over time. The continuous-time eigenvalues ω are complex valued, allowing us to decompose the dynamics into real and imaginary parts, $\omega = i\alpha + \beta$. This is given by:

$$\omega = \frac{\log(\lambda)}{\Delta t}, \tag{5.46}$$

where Δt denotes the discrete time interval between two consecutive measurements (snapshots) of the system in a discrete-time setting. Furthermore,

- the α denotes the imaginary part, determines the specific frequencies of the oscillating mDMD modes. The mDMD mode oscillates at a frequency given by α
- the β denotes the real part, determines the behaviour of the oscillating mDMD modes. The mDMD mode is growing $\beta > 0$, decaying $\beta < 0$ or neutral $\beta = 0$

We have considered plotting the eigenvalues on the complex plane and interpreted the stability of a dynamical system in the discrete time and (continuous time) sense:

- Stable modes are represented by eigenvalues on the unit circle(eigenvalues on the imaginary axis $\beta = 0$ for continuous time) in the left half of the complex plane
- Decaying modes are represented by eigenvalues that lie inside the circle ($\beta < 0$ for continuous time)
- Growing modes are represented by eigenvalues that lie outside the circle ($\beta > 0$ for continuous time)

The mDMD algorithm provides both spatial modes, which describe the system's structural patterns and temporal eigenvalues, which capture the time evolution of these modes. By examining the eigenvalue spectrum, we assess the stability and long-term behavior of the system, identifying the modes that are growing, decaying, or oscillating. The combination of spatial modes and temporal dynamics forms the foundation of how the mDMD method captures and predicts the evolution of complex systems, as we will demonstrate through the test cases presented next.

Chapter 6

Numerical results

In this chapter, we have presented numerical results. We have investigated the efficacy of the proposed method, mDMD, using five different test cases. For each case, we have conducted a comparative analysis between DMD and mDMD. The test cases exemplify complex dynamics spanning both temporal and spatial domains. Frequently, such problems lack explicit governing equations, yet high-dimensional dynamic data can be extracted through simulations, measurements or experimentation. Hence, it becomes imperative to understand how DMD and mDMD analyse high dimensional data related to dynamical systems. MATLAB R2023b programming language and environment is used for all the test cases in this thesis.
The test cases are as follows:

1. The first test case involves a simulated dataset of simple imaginary oscillations[4]. We present a system estimation application of mDMD.

2. The second test case involves a simulated dataset of a mixture of evolving signals[11]. We present a system's state estimation and future prediction.

3. The third test case delves into the realm of neuroscience [11], applying DMD and mDMD to real data collected from neuroscience experiments.

4. The fourth test case involves a fluid flow [4], we have analyzed a fluid flow around a wake using a simulated dataset.

5. The fifth test case investigates video processing using simulated data derived from a one-dimensional dynamic system [11], designed to replicate real-world video datasets.

6.1 Test case: Imaginary oscillations

Purely imaginary oscillations are oscillations in a system in which the imaginary part dominates, giving rise to sinusoidal behavior. Purely imaginary oscillations play a vital role in understanding the behavior and properties of linear and nonlinear systems, resonance phenomena, and frequency analysis in various scientific applications. In particular, purely imaginary oscillations are fundamental in signal processing applications, particularly in Fourier analysis. In engineering and physics, purely imaginary oscillations can be encountered in various contexts, such as in the analysis of electrical circuits, vibrations, or quantum mechanics. In this test case, we have implemented mDMD and DMD to analyse purely imaginary oscillatory data. The aim is to identify, reconstruct and predict the system states based on data. In this case, the data are given by the function values and our aim is to estimate and predict the values within and outside the simulated time frame. A simulated data set is generated using the following equation, representing a signal:

$$f(y,t) = e^{(a+bi)t} sin(ky). \tag{6.1}$$

Here, y is representing the space, t is representing time. The constants a and b represent growths or decays and oscillations while k is the wave number. We have considered a random case where $a = 0$, $b = 0.5$ and $k = 2$. To simulate (6.1), we have defined a state domain ranging from -10 to 10, subdivided into 400 equally spaced intervals. Similarly, we have defined a time domain with range $[0, 4\pi]$, subdivided into 200 equally spaced intervals. The equation (6.1) has been discretized, where $y \in \mathbb{R}^{1\times 400}$ and $t \in \mathbb{R}^{200\times 1}$. Then we create a system called $\mathbf{SYS} \in \mathbb{R}^{200\times 400}$ using the discrete points in space y_j and time t_i where an entry of **SYS**, denoted $X_i(t_j)$, is obtained using (t_i, y_j) such that $X_i(t_j) = f(t_i, y_j)$ for $i = 1, \ldots, 200$ and $j = 1, \ldots, 400$.

The mDMD linear approximation of (6.1) is formulated as:

- By taking the transpose of **SYS**, we defined a data matrix of the discretized states in rows for each discrete time in columns called $\mathbf{DAT} \in \mathbb{R}^{400\times 200}$. The matrix **DAT** is tall and skinny.

- To construct a linear operator, we have defined an input data matrix $\mathbf{X1} \in \mathbb{R}^{400\times 199}$ (the first 199 columns of **SYS**) and an output data matrix $\mathbf{X2} \in \mathbb{R}^{400\times 199}$

(the last 199 columns of **SYS**).

- We have applied **Algorithm 1** found on page 43: Step 2 to Step 6.

Results are reported below:
Firstly, we present the singular values, denoted as red circles in Figure 6.1. The y-axis represents the magnitude of the singular values, showing how much each singular value contributes to the total variability of the dataset. The first (initial) singular values contributing more significantly than the tail of the plot, this typically indicates that the singular values decrease in significance gradually. The ordering of the singular values fundamentally indicates that the dataset can be well approximated with fewer corresponding dominant modes:

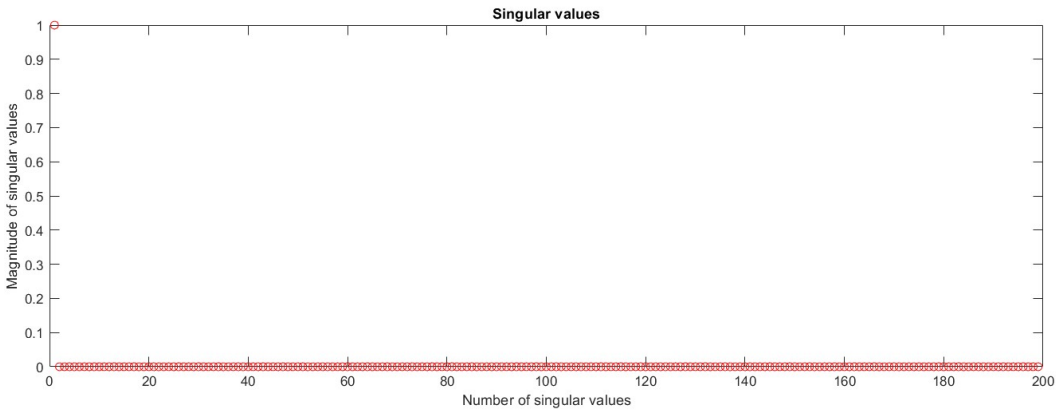


FIGURE 6.1: Singular values distribution

In Figure 6.1, we have used SVD on the input dataset **X1** to compute the singular values (199 of them), shown as red circles. The figure shows one leading singular value situated at the highest magnitude. With the rest seeming flat down on the x-axis indicating minimal contributions to the dynamics, not that they are equal to zero. This indicates that we can r-truncate the the input data matrix **X1**. According to (4.2), the input data matrix $\mathbf{X1} \approx \hat{\mathbf{U}}_1\hat{\Sigma}_1\hat{\mathbf{V}}_1^*$. This result imply, for this computed $r = 1$, there is no better approximation, in the L^2 sense, than $\mathbf{X1} \approx \hat{\mathbf{U}}_1\hat{\Sigma}_1\hat{\mathbf{V}}_1^*$. Thus, we are only using the first column of $\mathbf{U}$, the first entry of Σ and the first column of $\mathbf{V}$. Consequently, the high dimensional data matrix **X1** is approximated by a dominant pattern (mode) given by the column of $\hat{\mathbf{U}}_1 \in \mathbb{R}^{400\times 1}$ and $\hat{\mathbf{V}}_1 \in \mathbb{R}^{199\times 1}$. Here, $\hat{\mathbf{U}}_1$ serves as a SVD basis enabling a transformation, where one moves from

high-dimensional function values into a low-dimensional mode space. In so doing, we are reducing the dimension of the data set, obtaining a tractable basis.
With respect to (5.26) and an arbitrary time t, the result of SVD imply:

$$\mathbf{x}(t) = \sum_{j=1}^{1} \boldsymbol{\phi}_j exp(\omega_j t) b_j \tag{6.2}$$

In the following outcome, we show the reconstruction of (6.1) using (6.2):

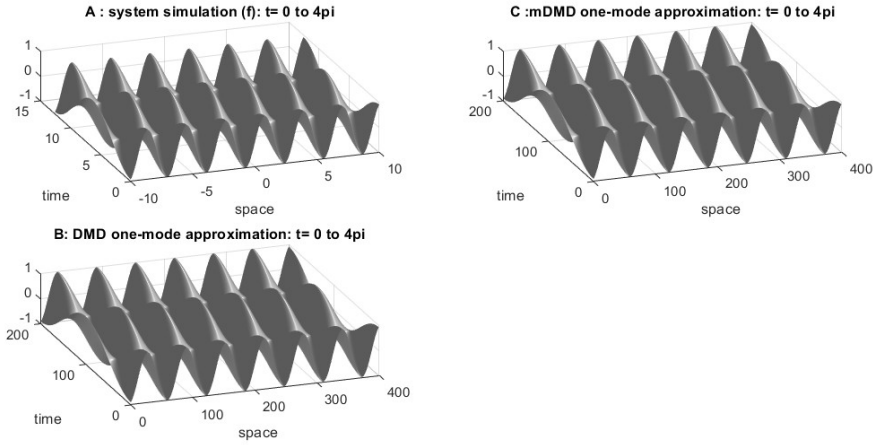


FIGURE 6.2: A: Exact evolution of the system $f(x,t)$. B: DMD approximation of the evolution of $f(x,t)$. C: mDMD approximation of the evolution of $f(x,t)$. Time is denoted by t in all the cases.

In Figure 6.2, we have demonstrated the reconstruction of the full system (6.1). Firstly, we have presented the exact solution labeled on Figure 6.2 as **A**, sampled within the time frame $t \in [0, 4\pi]$ and plotted using (6.1). A linear model is constructed using DMD, giving the reconstruction labeled on Figure 6.2 as **B**, and similarly using a linear model computed by mDMD we have obtained the reconstruction labeled on Figure 6.2 as **C**. The linear approximations by DMD or mDMD are constructed using **Algorithm 1**.
We consider (5.26) which gives DMD approximation taking note that $r = 1$ for this test case. We compute the vector $\mathbf{X}(t_j)$ using (6.2), the components of which are function values in space direction at each t_j as presented in Figure 6.2. The same computational procedure is implemented for mDMD.
Our findings have indicated that mDMD effectively estimates the system's state, exhibiting results comparable to DMD. In the following results of Table 6.1, we have

considered error analysis of the methods used. The error is computed within different time domains, extending from the initial time domain $[0, 4\pi]$. The new time domains are similarly subdivided into 200 equally spaced intervals. Since the models constructed are linear, we are comparing their future predictions to the current exact solution. This analysis helps in understanding the reliability of the model for future state predictions. An error vector has been computed by taking the norm of the difference between the exact solution and the approximated solution. The exact solution in this case is given by (6.1).

Now we present the computation of error. The DMD or mDMD approximates the matrix **SYS** using (5.26). The linear approximations of (6.1) is given by (5.26), denoted as $\tilde{\mathbf{X}}(t_j)$. The error vector is computed by: $\mathbf{ERR} = \|\mathbf{X}(t_j) - \tilde{\mathbf{X}}(t_j)\|^2$, where $\mathbf{X}(t_j) = [X_1(t_j) \quad X_2(t_j) \dots \quad X_{400}(t_j)]^T$ and $\tilde{\mathbf{X}}(t_j) = [\tilde{X}_1(t_j) \quad \tilde{X}_2(t_j) \dots \quad \tilde{X}_{400}(t_j)]^T$ at t_j, where T denotes the transpose in this case.

We have compared the **ERR** obtained from using DMD and mDMD linear approximations. The mean squared error (MSE) is the average of the squares of the errors between corresponding elements of the exact solution vector and the approximated solution vector.

Time	**DMD**	**mDMD**
t	MSE (rMSE)	MSE (rMSE)
$[0,4\pi]$	1.9896 (1.4105)	1.9845 (1.4087)
$[0,8\pi]$	1.9896 (1.4105)	1.9795 (1.4070)
$[0,12\pi]$	1.9896 (1.4105)	1.9746 (1.4052)
$[0,20\pi]$	1.9896 (1.4105)	1.9647 (1.4017)

TABLE 6.1: Mean squared error (MSE) and root MSE (rMSE).

We have computed the MSE and comparisons are made between DMD and mDMD in Table 6.1. Table 6.1 gives the square root of the MSE as well, indicated as rMSE. We have illustrated that as the time horizon increases, the DMD approximation generally gives a non changing error, as shown by the non decreasing pattern in MSE(rMSE) column of Table 6.1. Conversely, mDMD exhibits a decreasing pattern in the MSE(rMSE) column of Table 6.1. Thus, mDMD shows a lower discrepancy even as time progresses. A decreasing mean squared error with increasing prediction time indicates an improved and consistent predictive performance, observed in the proposed mDMD.

6.2 Test case: Signal processing

Signal processing plays a pivotal role in various fields, including engineering, medicine and telecommunications. Its importance lies in the extraction, manipulation, and analysis of information carried by signals, which are representations of physical phenomena. In recent years, methods like DMD have gained significant attention in signal processing tasks. The tasks become very challenging in scenarios where equations are either unknown or too complex to model accurately. We have extended the data driven benefits of DMD to our proposed mDMD in signal processing. In this case, we have shown the application of mDMD on a simulated data set of a mixture of two time dependent signals. The aim is to separate the resulting signal into its signal components. The data in this case are signal measurements given by the function values. Our aim is to estimate and predict the values within and outside the simulated time frame using a linear model.

$$s(t,y) = exp(i2.3t)sech(3+y) + 2exp(i2.8t)sech(y)tanh(y) \tag{6.3}$$

where we want to decompose the signal as $s(t,y) = s_1(t,y) + s_2(t,y)$. To simulate (6.3), we have defined a state domain ranging from -10 to 10, subdivided into 400 equally spaced intervals of measurements. In a similar manner, we have defined a time domain ranging from 0 to 4π, subdivided into 200 equally spaced intervals. The equation (6.3) has been discretized, where $y \in \mathbb{R}^{1\times 400}$ and $t \in \mathbb{R}^{200\times 1}$. Then we create a system called **SYS** $\in \mathbb{R}^{200\times 400}$ using the discrete points in space y_j and time t_i where an entry of **SYS**, denoted $X_i(t_j)$, is obtained using (t_i, y_j) such that $X_i(t_j) = f(t_i, y_j)$ for $i = 1,\ldots,200$ and $j = 1,\ldots,400$. The mDMD linear approximation of (6.3) is then formulated as:

- By taking the transpose of **SYS**, we defined a data matrix of the discretized states in rows for each discrete time in columns called **DAT** $\in \mathbb{R}^{400\times 200}$. The matrix **DAT** is tall and skinny.
- To construct a linear operator, we have defined an input data matrix **X1** $\in \mathbb{R}^{400\times 199}$ and an output data matrix **X2** $\in \mathbb{R}^{400\times 199}$.
- We have applied **Algorithm 1** found on page 43: Step 2 to Step 6.

Results are presented below:

In the first results, we present the recovery of the signal components by the proposed mDMD in Figure 6.3. Next we present the recovery of the mixed signal by both mDMD and DMD including the future prediction of the mixed signal. This is shown in Figure 6.4.

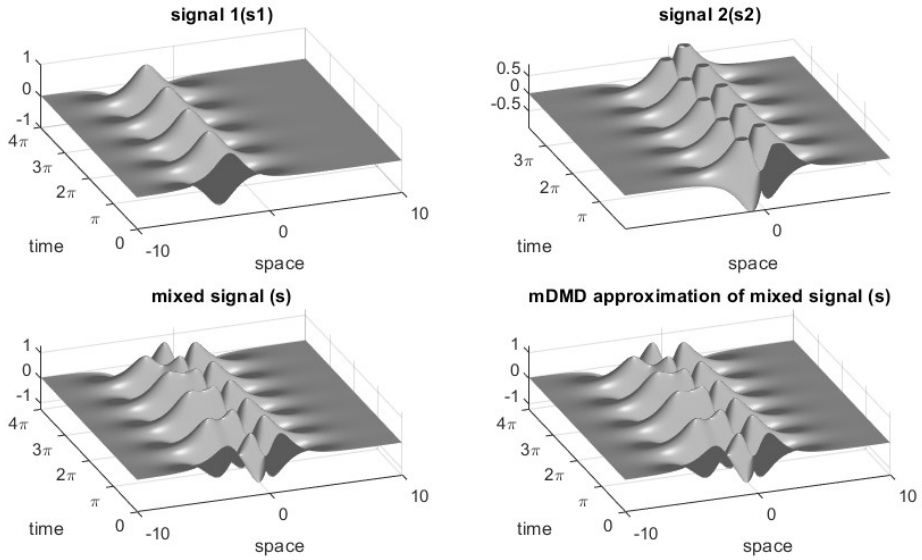


FIGURE 6.3: Top left: Signal given by s_1. Top right: signal given by s_2. Bottom left: mixed signal given by s. Bottom right: mDMD approximation of the mixed signal.

In Figure 6.3, we show how the mixed signal breaks down into its component signals (top panel). The bottom panel demonstrates the reconstructed system using mDMD (bottom right) compared to the original mixed signal (bottom left). With a two-mode approximation derived from the input data matrix's SVD, mDMD accurately captures the mixed signal.

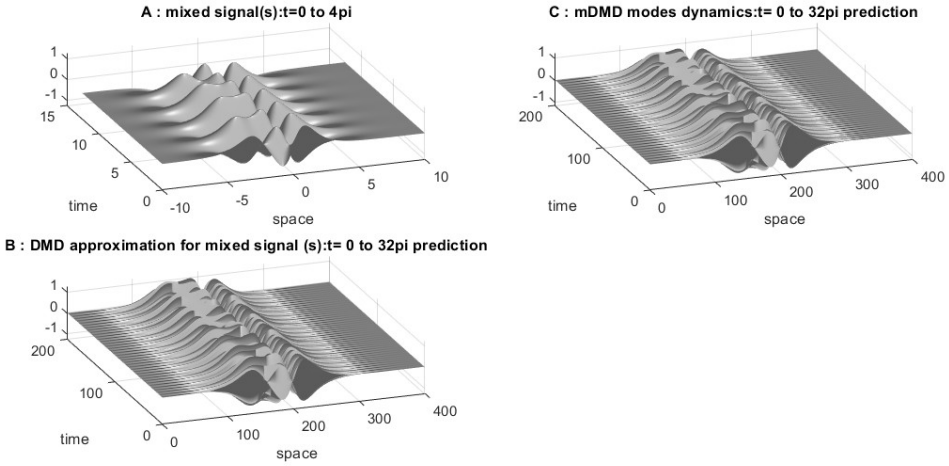


FIGURE 6.4: A: mixed signal, by $s(y,t)$. B: mixed signal, by DMD. C: mixed signal, by mDMD.

Figure 6.4 contrasts the signal reconstructions by DMD solution shown in Figure 6.4 B and mDMD solution shown in Figure 6.4 C. The reconstructed systems appear similar upon visual inspection, with no discernible differences observed. The DMD solution and mDMD solution approximated the system further, within $t \in [0, 32\pi]$. This provides a framework for future state estimation of the system outside the sampled time, $t \in [0, 4\pi]$ as shown in Figure 6.4 A. The future state is approximated using (5.26). Despite the similarities presented, further analysis reveals distinctions between the two methods.

In connection to previous findings, we have presented a convergence behavior plot of iterative errors. The result typically illustrates how the error changes over successive iterations when longer time prediction is considered. In many cases, a rapid decrease in error suggests efficient convergence towards the desired solution, while a slow or fluctuating decrease may indicate convergence issues or suboptimal performance of the algorithm.
In Figures 6.5 and 6.6 below, we have shown the error behaviour for different future time predictions by both methods. The error convergence behaviour for $t \in [0, 32\pi]$ is shown in Figure 6.5 while the error convergence behaviour for $t \in [0, 100\pi]$ is shown in Figure 6.6. For both methods.

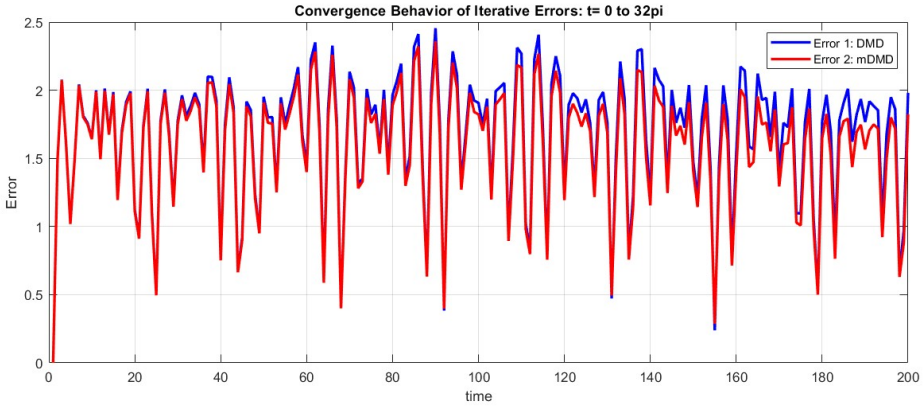


FIGURE 6.5: Convergence behaviour of error. mDMD shown in red. DMD shown in blue. At time $t \in [0, 32\pi]$

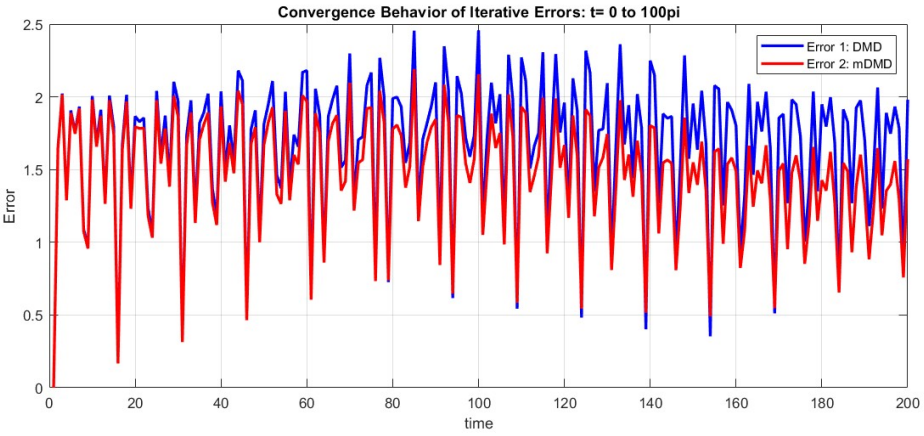


FIGURE 6.6: Convergence behaviour of error. mDMD shown in red. DMD shown in blue. At time $t \in [0, 100\pi]$

Following the results of Figures 6.5 and 6.6, we have illustrated the differences in the methods by computing error vectors. Specifically, Figure 6.5 corresponds to the interval $t \in [0, 32\pi]$, reveals improved error behavior in mDMD with future state prediction. This observation is similar with results in Figure 6.6, where the interval $t \in [0, 100\pi]$, a decrease in error fluctuations. The norm of the error vector is found to be 2.4553 for DMD and 2.3592 for mDMD within $t \in [0, 32\pi]$. The norm of the error vector is found to be 2.4582 for DMD and 2.1916 for mDMD within $t \in [0, 100\pi]$. We have concluded that there is a decrease in the error with longer time prediction for mDMD while an increase in the magnitude of the error with longer time prediction for DMD.
Finally, we have presented error results, given in Table 6.2. Error results are computed as in test case one.

Time	**DMD**	**mDMD**
t	MSE (rMSE)	MSE (rMSE)
[0,4π]	3.1237 (1.7674)	3.0864 (1.7568)
[0,8π]	3.0583 (1.7488)	2.9859 (1.7280)
[0,12π]	3.0237 (1.7389)	2.9163 (1.7077)
[0,20π]	3.0445 (1.7448)	2.8703 (1.6942)

TABLE 6.2: Mean squared error (MSE) and root MSE (rMSE).

At $t \in [0, 4\pi]$, the state estimation is within the defined time domain. At $t \in [0, 8\pi]$, we compute the state prediction further in time. Future state prediction, by doubling time, shows a decrease in all variables of Table 6.2 for mDMD but not entirely the same can be said for DMD. we can conclude that mDMD has a lower discrepancy compared to DMD, for any time prediction in this test case. After $t \in [0, 8\pi]$, mDMD shows a decreasing discrepancy as opposed to DMD.

6.3 Test case: Neuroscience

The human brain plays a vital function in mankind, spanning functions such as actions, emotions, sensations and decisions. Modern biology experiences a challenge in estimating and predicting how brain cells compute and trigger these functions, thereby affecting related treatment. Recent technological and infrastructural advancements have increasingly constructed ways of recording brain signals originating from brain cells. Other related findings indicate that brain or neural system related studies are moving from the challenge of acquiring data to the challenge of analysing complex dynamic data. Despite the variations in current measurement methods, the analysis of neural activity requires dealing with complex high-dimensional, dynamic, and noisy data.
This test case explores the feasibility of employing our proposed mDMD to analyze real neural data, expanding on ideas from [3]. To collect the real data, electrodes were positioned on the outer layer of a person's brain to record neural activity every second. In each electrode, voltage is measured. Further details of this are provided in [4]. The recordings of neural activity have been transformed to its matrix representation: NEU $\in \mathbb{R}^{484\times1003}$ where each row is a fixed time.

- We have defined a data matrix, from NEU, called **DAT** $\in \mathbb{R}^{1003\times484}$, so that time is the column space and measurements are the row space. The matrix **DAT** is tall and skinny.
- To construct a linear operator, we have defined an input data matrix **X1** $\in \mathbb{R}^{1003\times483}$ and an output data matrix **X2** $\in \mathbb{R}^{1003\times483}$.
- We have applied **Algorithm 1** found on page 43: Step 2 to Step 6.

The following results are reported:

We have firstly presented spectrum plots illustrating the distribution of eigenvalues in Figure 6.7, offering insight to the time dynamics of the mDMD modes. Comparisons with results obtained using DMD are provided in Figure 6.8, shedding further light on the positioning and behavior of the eigenvalues within each framework. These visualizations serve to deepen our understanding of the temporal characteristics inherent in mDMD and DMD modes.

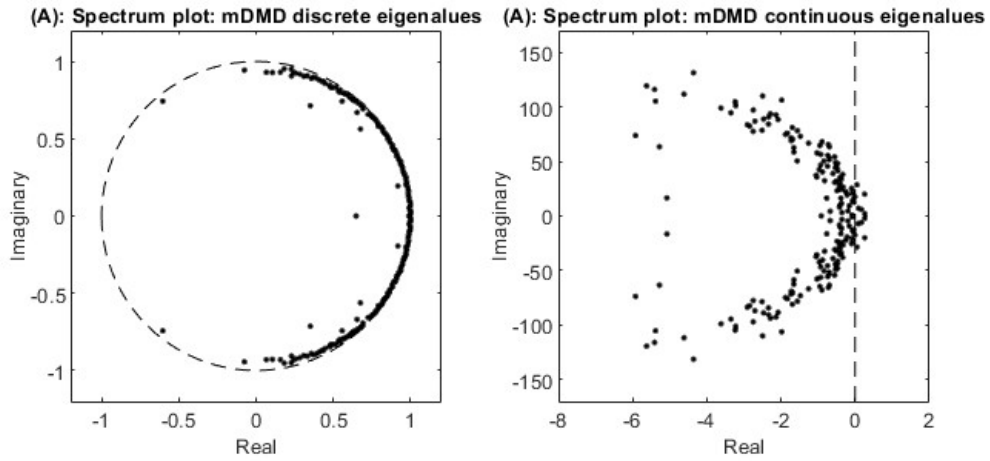


FIGURE 6.7: The figure presents plots of the mDMD eigenvalues. For the discrete time dynamics, λ (left). For the continuous time dynamics, ω (right).

Figure 6.7 presents a spectrum plot of the eigenvalues of the linear operator $\mathbf{A}$. The discrete-time plot (left) and continuous-time plot (right) reveal eigenvalues inside the unit circle and on the left-hand side of the imaginary axis, respectively. These findings indicate a decaying behavior of the associated modes. Additionally, most eigenvalues lie on the unit circle in the discrete time case or on the imaginary axis in the continuous time case, suggesting oscillatory behavior without growth or decay. Notably, no eigenvalue is captured outside the unit circle in the discrete case, and only a few are on the right-hand side of the imaginary axis in the continuous case, implying negligible growth behavior in the modes.

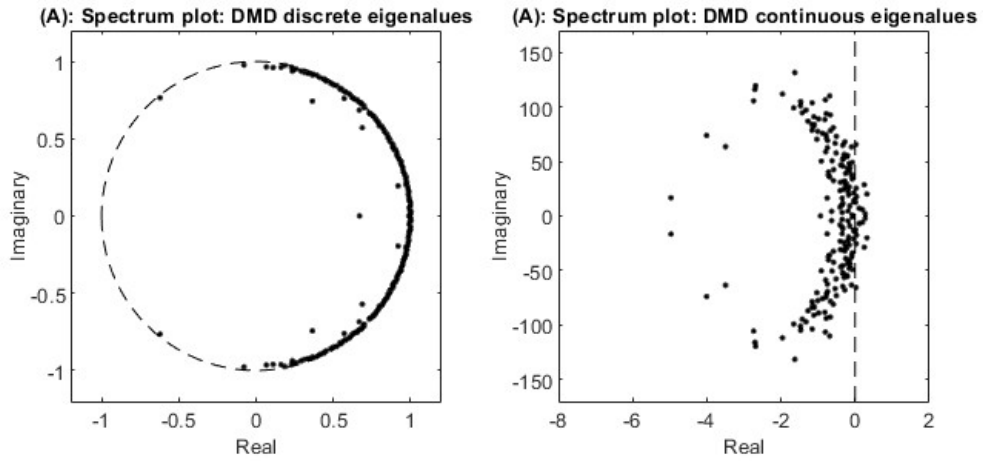


FIGURE 6.8: The figure presents plots of the DMD eigenvalues. For the discrete time dynamics, λ (left). For the continuous time dynamics, ω (right).

Figure 6.8 compares the results of mDMD, from Figure 6.7, to those of DMD method, revealing fewer eigenvalues inside the unit circle or on the left-hand side of the imaginary axis, indicating less decaying behavior in the associated modes. Furthermore, DMD suggests more oscillating modes without growth or decay compared to our proposed mDMD. Lastly, DMD's spectral plot for the continuous case suggests more growth behavior in the modes.

Our next results are shown in Figures 6.9 and 6.10. These results present a comparison between the mDMD mode spectrum and the energy spectrum computed by FFT, in Figure 6.9, similarly in Figure 6.10 for DMD. For the FFT analysis, the energy distribution is computed separately for each individual electrode, depicted in gray, and then averaged, shown in black. The spectrum of the mDMD modes or DMD modes are used to indicate energy distributions based on their frequencies, the results are similar to what FFT results of the power spectrum.

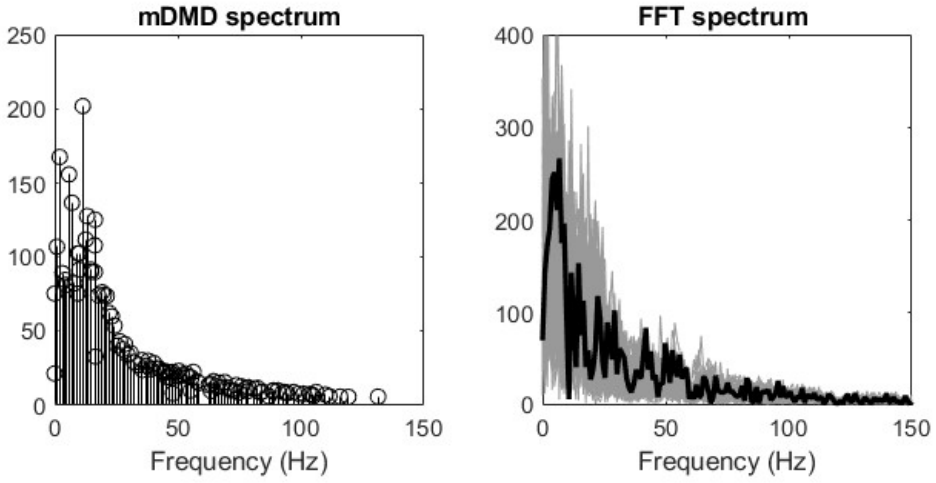


FIGURE 6.9: The mDMD mode-amplitude spectrum (left) and fast Fourier transform (FFT)-energy spectrum (right).

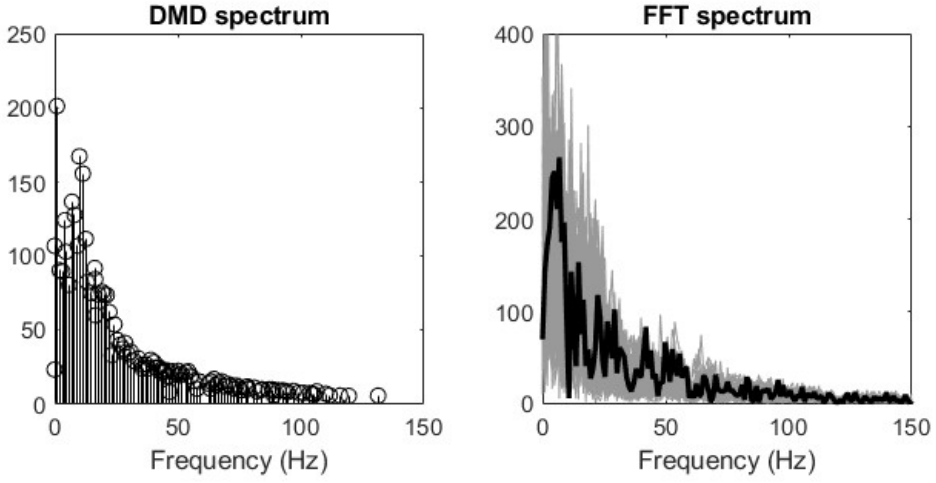


FIGURE 6.10: The DMD mode-amplitude spectrum (left) and fast Fourier transform (FFT)-energy spectrum (right).

In Figure 6.9, we observe that the proposed mDMD method demonstrates consistent decay in mode amplitude as the frequency increases. Unlike the continuous

FFT spectrum, the mDMD mode spectrum is not continuous, allowing for advantageous grouping of modes with similar amplitudes when regular sampling is desired. Figure 6.10 displays the results obtained with DMD, revealing slight differences in modes with high frequencies compared to mDMD. Specifically, mDMD captured 200 modes with a frequency around 10Hz, while DMD captured 200 modes with a frequency of 0Hz.

6.4 Test case: Fluid flow around a cylinder wake

The wake is the region of disturbed flow downstream of an obstacle or body moving through a fluid. A wake is typically characterized by vorticity. In the context of a wake behind an obstacle or a body moving in a fluid, vorticity arises due to the shear and deformation of fluid elements. In the wake region, vorticity is often concentrated within structures called vortices, which are formed as a result of the interaction between the flow and the obstacle. The presence of vorticity in the wake affects various flow characteristics, including turbulence levels, flow separation, and the overall flow pattern. Vorticity can be visualized using numerical simulations, or vorticity measurements. Thus, measuring vorticity helps to estimate the complex flow structures and dynamics within the wake region. The fluid flow is laminar, at Reynolds number Re = 100, see [4]. The fluid flow is governed by the two-dimensional Navier-Stokes equation. A simulation of the Navier-Stokes is utilised to generate the data.
When the simulated fluid flow has reached a stable condition where vortices are consistently formed and shed from the object or obstacle in the flow, 150 snapshots are collected at the same intervals in time of $10\Delta t$. A vortex shedding is approximately $300\Delta t$ where $\Delta t = 0.02$. These are close to experimental results [4]. In this test case, we have considered applying mDMD on a vorticity field of a fluid flow. Comparisons with DMD are made. The recordings of all vorticity field data have been transformed to its matrix representation: VORTALL $\in \mathbb{R}^{151\times 89351}$, each row representing a fixed time for the different states in the vorticity field.

- We have defined a data matrix, from VORTALL, called $\mathbf{DAT} \in \mathbb{R}^{98351\times 151}$, so that time is the column space and measurements are the row space. The matrix **DAT** is tall and skinny.
- To construct a linear operator, we have defined an input data matrix $\mathbf{X1} \in \mathbb{R}^{98351\times 150}$ and an output data matrix $\mathbf{X2} \in \mathbb{R}^{98351\times 150}$.
- We have applied **Algorithm 1** found on page 43: Step 2 to Step 6.

The following results are obtained:

For this test case, we have firstly presented the mDMD modes in Figures 6.11 to 6.14 and DMD modes in Figures 6.16 to 6.19. The SVD analysis of the input data matrix suggested 21 leading singular values. We have plotted the mode pairs that show a periodic vortex shedding motion. In the context of fluid flows, the periodic shedding motion is often given by harmonic sine and cosine functions because of their repetitive nature at regular intervals.
Next, we have presented the spectral plot of the discrete eigenvalues in Figure 6.15 and Figure 6.20 for mDMD and DMD, respectively.

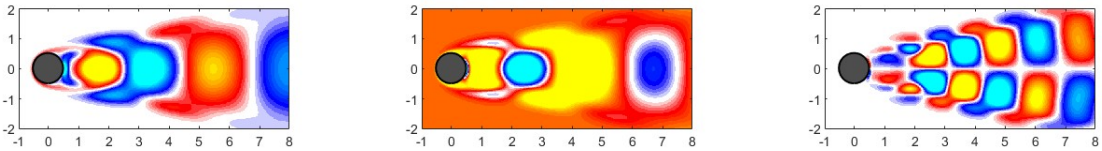


FIGURE 6.11: Mean flow(first picture), dominant mode(second picture), and the first mDMD mode pair (third picture)

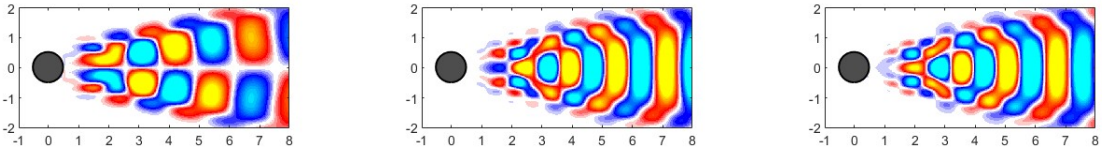


FIGURE 6.12: Second, third and fourth mDMD mode pairs

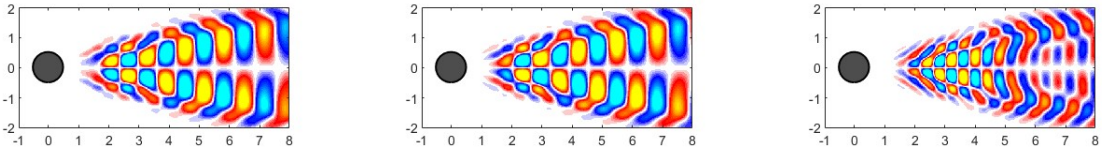


FIGURE 6.13: Fifth, sixth, seventh mDMD mode pairs

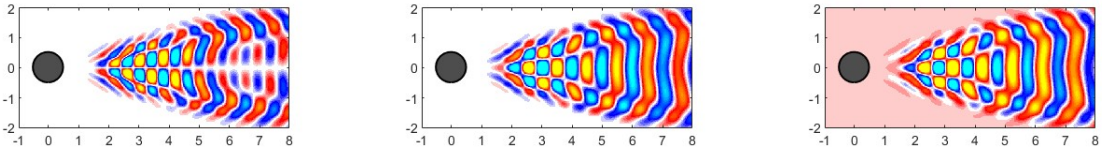


FIGURE 6.14: Eighth, ninth, tenth mDMD mode pairs

The spectrum plots, Figures 6.15 and 6.20 indicate the time dynamics of the associated mode pairs. The eigenvalues are computed from the finite dimensional linear operator, computed by the mDMD for Figure 6.15 and DMD for Figure 6.20.

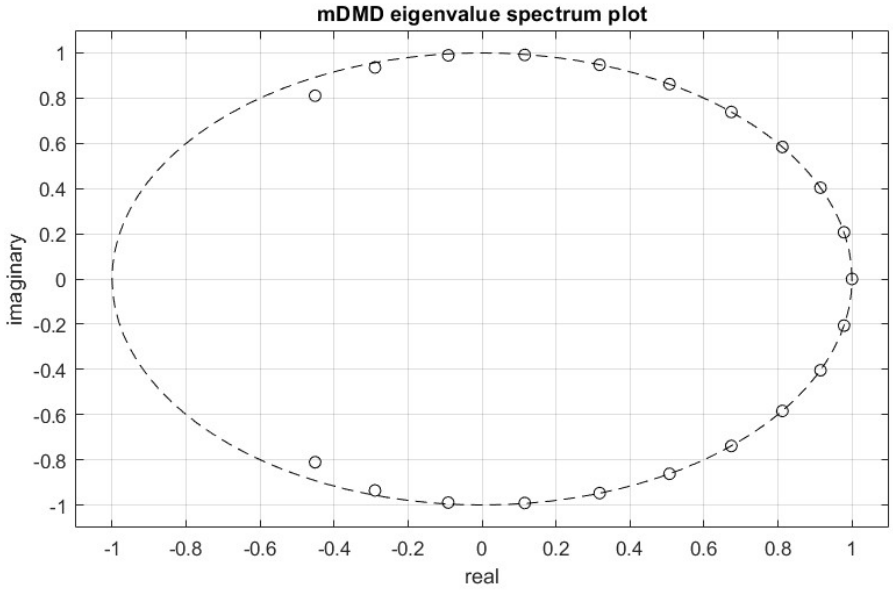


FIGURE 6.15: The spectrum plot of the eigenvalues corresponding to mDMD modes

Figures 6.11 to 6.14 depict the mode pairs, Figure 6.15 illustrates their corresponding eigenvalues. Each mDMD mode exhibits temporal behavior dictated by its associated eigenvalue, excluding for the second mDMD mode (Figure 6.11), characterized by a zero eigenvalue as shown Figure 6.15. The first plot on Figure 6.11 is the mean flow. We have plotted the 10 mode pairs (third mode to twelfth mode of Figures 6.11 to 6.14) with temporal dynamics, each symmetric pair denotes the temporal dynamics of each mode pair.

The mDMD framework effectively identifies both coherent structures and their temporal dynamics, thereby defining a reduced linear dynamic system that approximates the entire dynamical system. Notably, modes associated with eigenvalues inside the unit circle, as observed in Figure 6.15, tend to demonstrate a decaying behavior of the associated mode pairs in this case.

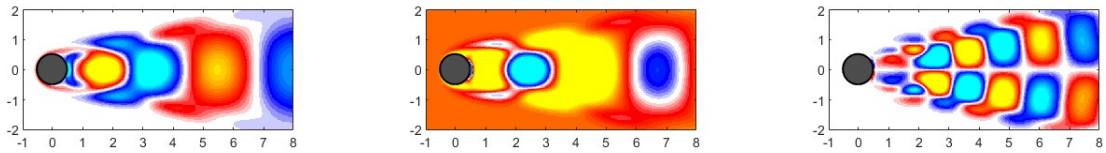


FIGURE 6.16: Mean flow(first picture), dominant mode(second picture), and the first DMD mode pair (third picture)

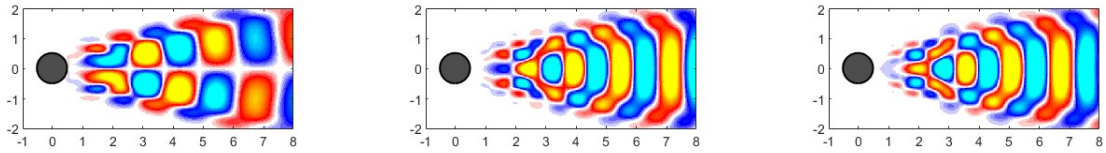


FIGURE 6.17: Second, third and fourth DMD mode pairs

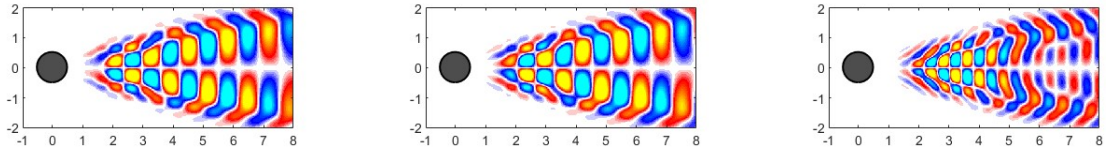


FIGURE 6.18: Fifth, sixth and seventh DMD mode pairs

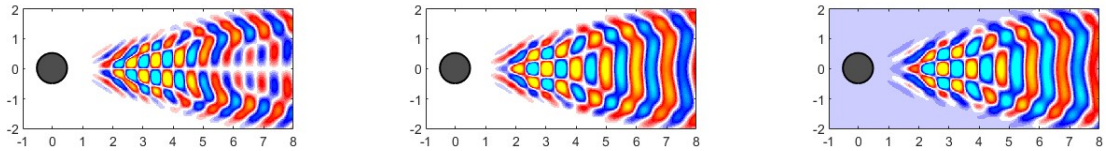


FIGURE 6.19: Eighth, ninth and tenth DMD mode pairs

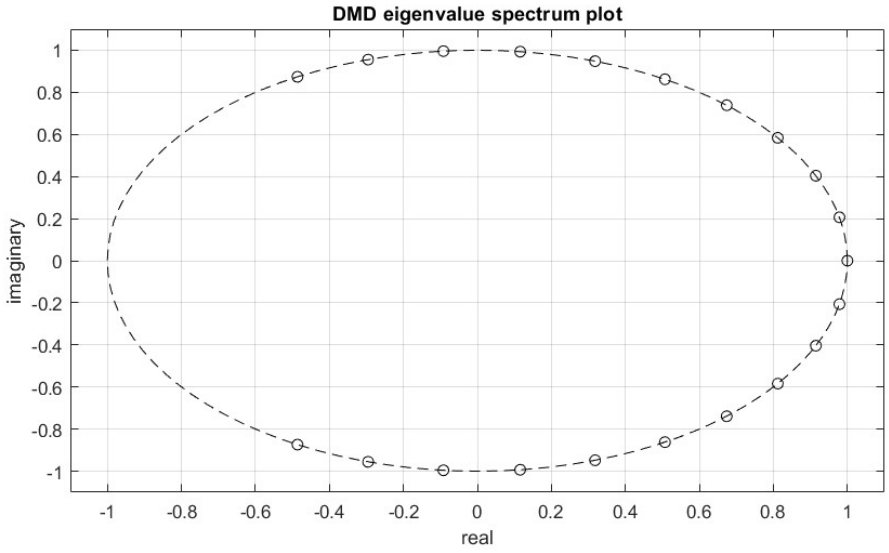


FIGURE 6.20: The spectrum plot of the eigenvalues corresponding to DMD modes

The mDMD outcomes on this dataset are contrasted with DMD findings. While both methods exhibit similar results in capturing dominant modes, slight disparities in the distribution of eigenvalues on the unit circle are noticeable. The mDMD results imply decaying behavior among modes, as evidenced by certain eigenvalues positioned inside the unit circle. Conversely, eigenvalues yielded by DMD, Figure 6.20, suggest that these modes consistently oscillate with growth and not decaying behaviour, with corresponding eigenvalues precisely on the unit circle.
This example illustrates that the proposed mDMD offers outcomes that may align more closely with reality, where we observe a decay of vortex shedding. The decay of vortex shedding refers to the reduction or damping of this oscillatory behavior over time. This can manifest as a blurring or merging of the vortices in the flow field images, indicating a decrease in shedding intensity or coherence, tenth mode pair of Figure 6.14 and Figure 6.19 for mDMD and DMD respectively. Our proposed mDMD captures the decaying behaviour as opposed to DMD, shown in Figure 6.15 and Figure 6.20.

6.5 Test case: Video Processing

Video processing has applications in video surveillance and recognition. There is a growing demand for competitive and real-time video surveillance techniques. For a wide range of computer vision applications and related algorithms, the focus is on the ability to separate background variations in a video stream from objects of interest, namely foreground objects. Thus, background or foreground separations is challenging in the context of detecting, identifying, tracking, and object recognition from the high dimensional video data. We have considered the video frames as snapshots of a complex, nonlinear dynamical system. The modes associated with nearly zero eigenvalues correspond to stationary or low-rank background components, while modes with nonzero eigenvalues represent non-stationary or sparse foreground motions. The proposed mDMD is tested to extract time-dependent and time independent components from video frames, mixture of signals. We have considered a simple example to mimic the actual video. The data in this case are the function values, we aim to estimate and predict these values within and outside the simulated time frame using simulated data. We have considered the following function:

$$f(t,y) = 2sech(y)tanh(y)exp(i2.8t) + 0.5cos(y), \tag{6.4}$$

where $f(t,y) = f_2(t,y) + f_1(y)$ such that $f_1(y)$ does not depend on time while $f_2(t,y)$ depends on time. To simulate (6.4), we have defined a state domain x from -15 to 15, subdivided into 200 equally spaced intervals of measurements. We have defined a time domain t from 0 to 4π, subdivided into 400 equally spaced intervals. Equation (6.4) has been discretized similar to the discretization in test case one, such that the system is given as SYG $\in \mathbb{R}^{400\times200}$. The mDMD linear approximation of (6.4) is formulated as:

- By taking the transpose of **SYG**, we defined a data matrix of the discretized states in rows for each discrete time in columns called **DAT** $\in \mathbb{R}^{200\times400}$. The matrix **DAT** is short and fat.
- To construct a linear operator, we have defined an input data matrix **X1** $\in \mathbb{R}^{200\times399}$ and an output data matrix **X2** $\in \mathbb{R}^{200\times399}$.
- We have applied **Algorithm 1** found on page 43: Step 2 to Step 6.

The results are reported next:
We have firstly presented the reconstruction of the video example. The reconstruction begins with the SVD of the input data matrix **X1**, which displayed two leading singular values, however, the results are not included here. In Figure 6.21, the exact solution is given, labelled Figure 6.21 A, for time $t \in [0, 4\pi]$. The figure shows the 2-mode approximation by DMD, labelled Figure 6.21 B, and the 2-mode approximation by mDMD, labelled Figure 6.21 C. The mDMD and DMD solutions give approximations extended to $t \in [0, 8\pi]$, showing future state predictions, by doubling the time.

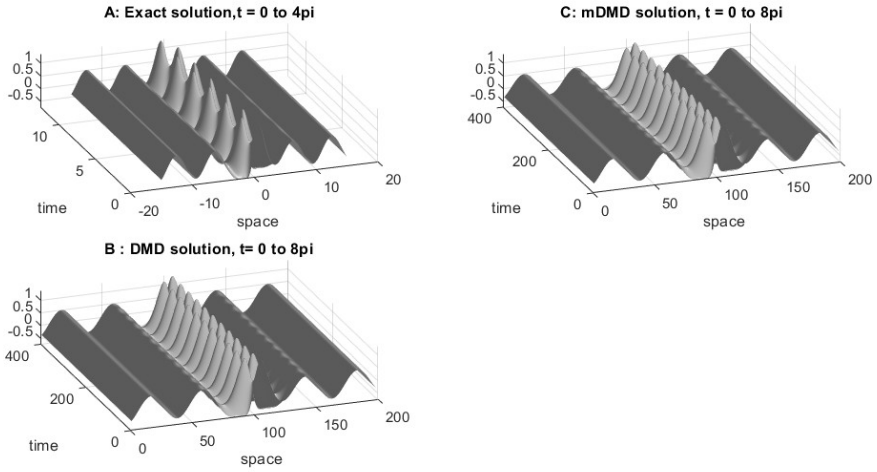


FIGURE 6.21: A: Original video, $f(y,t)$. B: DMD estimation and prediction with two dominant modes. C: mDMD estimation and prediction with two dominant modes.

In Figure 6.21, we have reconstructed the video signal within the sampled time $t \in [0, 4\pi]$ frame and shown predictions outside the sampled time frame $t \in [0, 8\pi]$ through the linear models given by mDMD and DMD. Remarkably, both techniques yield highly accurate reconstructions of the original mixture of signals. This demonstrates the robustness and efficacy of both approaches for future state prediction.
Secondly in Table 6.3, we have presented the error associated to the DMD and mDMD approximations of the exact system, the error vectors are calculated in the same way as described previously in test case one. The error is computed within the sampled time domain $[0, 4\pi]$ and extended for future time predictions.

Time	**DMD**	**mDMD**
t	MSE (rMSE)	MSE (rMSE)
$[0,4\pi]$	1.9587 (1.3807)	1.9065 (1.3807)
$[0,8\pi]$	2.0032 (1.4154)	1.9198 (1.3856)
$[0,12\pi]$	1.9800 (1.4071)	1.8630 (1.3649)
$[0,20\pi]$	1.9936 (1.4120)	1.8078 (1.3446)

TABLE 6.3: Mean squared error (MSE) and root MSE (rMSE).

We have explored the observed approximations from Figure 6.21, comparing MSE calculations as shown in Table 6.5. For $t \in [0, 8\pi]$ DMD gives a mean squared error of 2.0032 while mDMD gives a mean squared error of 1.9198, there after, we see a consistent drop of the MSE for the mDMD as opposed to the DMD as longer time prediction is taken. We have concluded that mDMD consistently produce better linear models for estimation and prediction for this class of problems .

Next, we have presented the decomposition of the signal into its component signals by mDMD, top panel of Figure 6.22. Further we show, bottom panel of Figure 6.22, the spectrum plot of the continuous eigenvalues. The position on the spectrum plot indicating temporal behaviour of the corresponding mDMD modes.

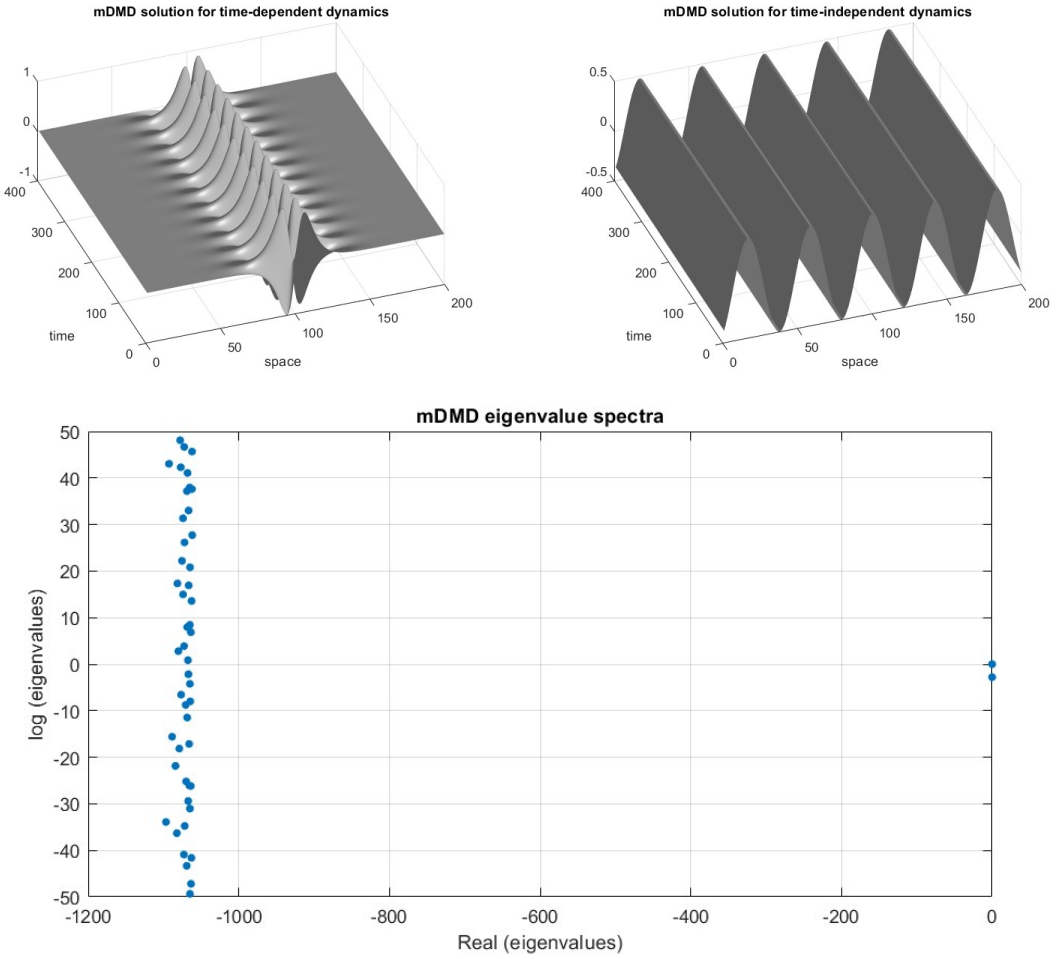


FIGURE 6.22: Top left: Time dependent dynamics $f_2(y,t)$, foreground. Top right: Time independent modes $f_1(y)$, background. Bottom panel: The mDMD distribution of spectral quantities, two eigenvalues approximately equal to zero are seen by the edge, on the origin.

Figure 6.22 shows the capability of the proposed mDMD to disentangle the mixed dynamics of the full system $f(y,t)$ into its constituent parts, top panel: where the time-dependent signal $f_1(y,t)$, is on the left hand side, and the time-independent signal $f_2(y,t)$, is on the right hand side. This signal separation achieved by mDMD

underscores its efficacy in investigating these type of dynamic systems. Furthermore, the Figure 6.22 provides valuable insights into the temporal dynamics of the individual signals, represented by the dominant modes, by the associated eigenvalues depicted on the spectrum plot (bottom). The eigenvalues are computed from the finite dimensional linear operator. The eigenvalues are indicated as blue dots with the background mode evolution described by the two eigenvalues on the edge and the foreground mode evolution described by the aligned eigenvalues.

Next, we have presented the convergence behaviour of the error, as described previously in Section 6.2. In Figure 6.23, we have plotted for DMD, in blue, and mDMD, in red. The plot is taken for $t \in [0, 8\pi]$. In Figure 6.24, we have presented similar results for $t \in [0, 80\pi]$.

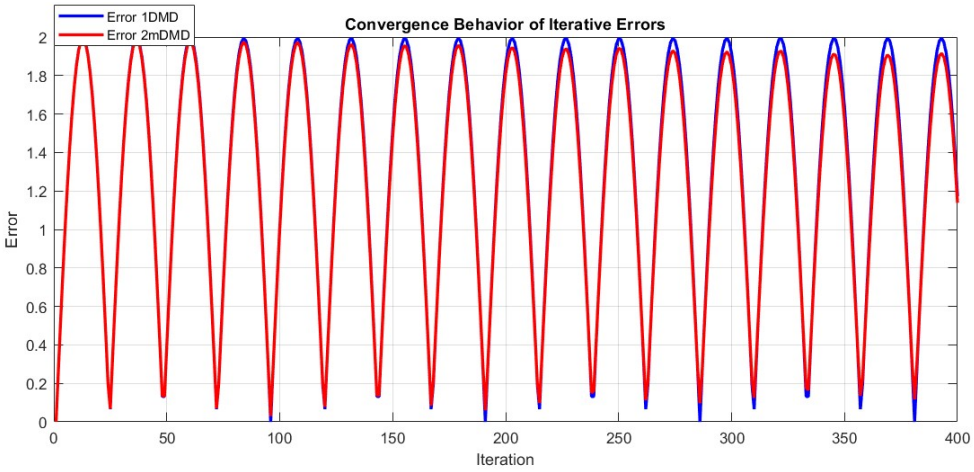


FIGURE 6.23: The convergence behaviour of the error iteratively for: $t = 0$ to $t = 8\pi$ (short time future system state approximation).

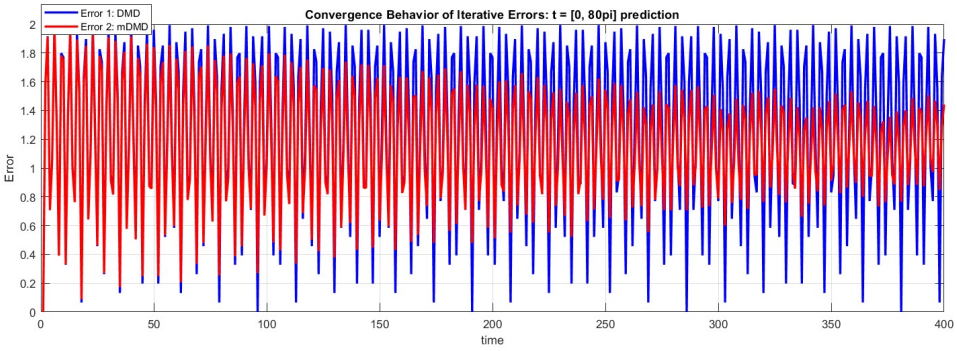


FIGURE 6.24: The convergence behaviour of the error iteratively for: $t = 0$ to $t = 80\pi$ (longer time future system state approximation).

Figure 6.23 shows a comparison between the error convergence patterns of DMD (blue) and the proposed mDMD approach (red). Initially, for short-time system state predictions, both methods exhibit similar fluctuating error trends with DMD having a MSE of 1.9847 while mDMD having a MSE of 1.7218 . However, as depicted in Figure 6.24, as the time horizon for state prediction increases, the error

convergence behaviour of mDMD shows notable improvement, marked by diminishing fluctuations, with DMD having a MSE of 1.9889 while mDMD having a MSE of 1.5161 . This finding underscores the enhanced predictive capabilities of mDMD over extended time intervals.

Chapter 7

Conclusions

In this thesis, we have explored the foundational concept of Dynamic Mode Decomposition (DMD), which seeks to reduce (linearize) the complexity (nonlinearity) of high-dimensional dynamic data of a dynamical system by solving an optimization problem. DMD operates on the assumption that within a high-dimensional nonlinear system, a lower-dimensional linear dynamic model can approximate the system's behavior. By solving an optimization problem, DMD aims to discover a matrix that captures these underlying dynamics. Central to DMD is the linear Koopman operator, which governs the linear evolution of all measurement functions in an infinite-dimensional space. We have dealt with direct measurements of the state as opposed to measurement functions of the state. Since this operator functions in an infinite-dimensional context, DMD focuses on computing a finite-dimensional approximation to it, providing insight into the system's dynamics. This direct connection to Koopman operator theory distinguishes DMD from other decomposition methods, as it not only identifies dominant spatial modes but also tracks their temporal evolution through the associated eigenvalues of the linear operator [14] .
In our study, we not only explored the foundational principles of DMD but also sought to enhance its core framework by introducing a matrix $\mathbf{B}$, which we have generally perceived as a regularization component to the optimization problem. This regularizer modifies the linear dynamic model governed by the linear operator (Koopman operator), aiming to improve the robustness and accuracy of DMD. Regularization, common in optimization, necessarily prevents overfitting and ensures that the model can generalize effectively by penalizing overly complex or extreme solutions. By incorporating this regularization term, we aim to refine the DMD method to better capture the true underlying dynamics of complex systems, offering more reliable and interpretable results, particularly in noisy or high-dimensional

data settings.

Further, the modified DMD (mDMD) demonstrates enhanced performance in our numerical tests, reconstructing the system with greater accuracy by leveraging dominant modes and their temporal dynamics. Test case one, for instance, validates the effectiveness of mDMD in identifying coherent spatial patterns from high-dimensional data. These patterns, or modes, mirror the capabilities of traditional DMD while providing a more refined structure due to regularization. As a result, mDMD not only captures the key dynamics of the system but also facilitates the construction of accurate, lower-dimensional linear dynamic models directly from the data. This reinforces the power of mDMD in handling complex systems with improved stability and interpretability, as demonstrated in other case studies.

Our research contributes significantly to advancing the field of dynamic system analysis through mDMD. One key finding is the enhanced predictive accuracy of mDMD, as demonstrated by reduced error norms, especially highlighted in test case two. This improvement showcases the effectiveness of mDMD in capturing dynamic behavior with greater precision. Moreover, mDMD proves its applicability to real-world data, notably demonstrated in test case three, where the resulting mode spectra closely resemble power spectra derived from Fast Fourier Transform (FFT) methods. A particularly valuable feature of mDMD is its ability to compute periodic mode pairs, as exemplified in test case four, further emphasizing its strength in uncovering periodic dynamics within complex datasets. Additionally, in test case five, mDMD successfully disentangles mixed signals from diverse components of dynamic systems, highlighting its versatility and robustness in signal separation. These findings demonstrate mDMD's capacity for handling a wide range of applications, marking a significant leap forward in the realms of data-driven analysis and model construction.

Regularization in optimization offers significant advantages, particularly by reducing the variance in the computed finite-dimensional Koopman operator, which enhances the stability and robustness of the model. This results in more reliable and consistent predictions. In our study, we demonstrate the effectiveness of applying regularization to the modified optimization problem. We computed the finite-dimensional Koopman operator and the regularizer using the Frobenius norm, which minimizes overall model error without encouraging sparsity, as compared to the L_1 norm, which is often employed for promoting sparse solutions.

Looking ahead, our future research aims to incorporate optimal control variables into the mDMD framework, thereby extending its applicability to broader dynamic systems. The numerical test cases reveal an overall improvement with mDMD, underscoring the necessity for further assessments to confirm its robustness across a broader range of scenarios. However, it is essential to recognize the limitations inherent in this study, including the limited number of test cases, which may influence the generalizability of the results. Future research should aim to classify the types of systems that benefit most from each method, thereby pinpointing their respective strengths in relation to specific study goals. Additionally, investigating how the proposed modifications perform across various DMD variants will validate their efficacy and potentially expand the scope of linear dynamic models within the DMD framework.

Bibliography

[1] Ralph Abraham, Jerrold E Marsden, and Tudor Ratiu. *Manifolds, tensor analysis, and applications*. Vol. 75. Springer Science & Business Media, 2012.

[2] Peter J Baddoo et al. "Physics-informed dynamic mode decomposition (piDMD)". In: *arXiv preprint arXiv:2112.04307* (2021).

[3] Bingni W Brunton et al. "Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition". In: *Journal of neuroscience methods* 258 (2016), pp. 1–15.

[4] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[5] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the national academy of sciences* 113.15 (2016), pp. 3932–3937.

[6] Steven L Brunton et al. "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control". In: *PloS one* 11.2 (2016), e0150171.

[7] Steven L Brunton et al. "Modern Koopman theory for dynamical systems". In: *arXiv preprint arXiv:2102.12086* (2021).

[8] Matthew J Colbrook. "The mpEDMD algorithm for data-driven computations of measure-preserving dynamical systems". In: *arXiv preprint arXiv:2209.02244* (2022).

[9] Zlatko Drmač. "Dynamic Mode Decomposition—A Numerical Linear Algebra Perspective". In: *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications* (2020), pp. 161–194.

[10] Bernard O Koopman and J v Neumann. "Dynamical systems of continuous spectra". In: *Proceedings of the National Academy of Sciences* 18.3 (1932), pp. 255–263.

[11] J Nathan Kutz et al. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[12] Alexandre Mauroy and Igor Mezić. "Global stability analysis using the eigenfunctions of the Koopman operator". In: *IEEE Transactions on Automatic Control* 61.11 (2016), pp. 3356–3369.

[13] Alexandre Mauroy, Y Susuki, and I Mezić. *Koopman operator in systems and control*. Springer, 2020.

[14] Igor Mezić. "Spectral properties of dynamical systems, model reduction and decompositions". In: *Nonlinear Dynamics* 41 (2005), pp. 309–325.

[15] Archana Muniraju. "Analysis of Dynamic Mode Decomposition". PhD thesis. The University of Wisconsin-Milwaukee, 2018.

[16] Clarence W Rowley et al. "Spectral analysis of nonlinear flows". In: *Journal of fluid mechanics* 641 (2009), pp. 115–127.

[17] Peter J Schmid. "Dynamic mode decomposition of numerical and experimental data". In: *Journal of fluid mechanics* 656 (2010), pp. 5–28.

[18] JH Tu et al. "37064: On dynamic mode decomposition: theory and applications. vol. 1, issue 2". In: *J. Comput. Dyn* (2014), pp. 391–421.

[19] Jonathan H Tu. "Dynamic mode decomposition: Theory and applications". PhD thesis. Princeton University, 2013.

[20] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition". In: *Journal of Nonlinear Science* 25 (2015), pp. 1307–1346.