FLOW CONTROL AND SENSING USING DATA-DRIVEN REDUCED-ORDER MODELING

BY

PALASH SASHITTAL

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Aerospace Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Daniel J. Bodony, Chair
Associate Professor Randy H. Ewoldt
Professor Theodore J. Heindel
Assistant Professor Andres Goza
Assistant Professor Laura Villafañe Roca

ABSTRACT

Transfer operators, such as the Koopman operator, are driving a paradigm shift in how we perform data-driven modeling of fluid flows. Approximations of the Koopman operator provide linear representations even for strongly nonlinear flows, which enables the application of standard linear methods for estimation and control under realistic flow conditions. In the past decade, we have witnessed several breakthroughs in obtaining low-dimensional approximations of the Koopman operators, providing a tractable reduced-order model for complex fluid flows using data from numerical simulations or experiments. In this thesis, we leverage these recent developments in operator-theoretic modeling of fluid flows and provide data-driven solutions to the flow control and sensing problems. The contributions of this thesis can be divided into three parts.

In the first part, we introduce a novel method, low-rank Dynamic Mode Decomposition (lrDMD), for data-driven reduced-order modeling of fluid flows. While existing data-driven modeling methods fit an endomorphic linear function on a low-dimensional subspace, lrDMD approximates flow dynamics using a linear map between different subspaces. We show that this approach leads to the design of better reduced-order feedback controllers. We formulate a rank-constrained matrix optimization problem and propose two complementary methods to solve the problem. lrDMD outperforms existing methods in feedback control and optimal actuator placement.

In the second part, we present a completely data-driven framework for sensor placement in fluid flows. This framework can be applied in conjunction with any reduced-order modeling technique that constructs a linear model for the flow dynamics. We formulate an optimization problem that minimizes the trace of a data-driven approximation of the estimation error covariance matrix, where the estimates are provided by a Kalman filter. We propose an efficient adjoint-based gradient descent method to solve the optimization problem. We show that sensors placed using our method lead to better performance in important applications, such as flow estimation and control, compared to existing data-driven sensor placement methods.

In the third and final part, we propose a new method of interface tracking and reconstruction in multiphase flows using shadowgraphs or back-lit imaging data. First, we show that while traditional modeling methods provide valuable information about the spatio-temporal structure of flow instabilities, they are not able to resolve spatial or temporal discontinuities, such as the liquid-gas interface, in the data. To remedy this, we propose a two-step approach, using data-driven modeling techniques in conjunction with optical flow methods, that preserves sharp interfaces and provides reliable reconstruction and short-time prediction of the flow. We apply our method to an experimental liquid jet with a co-axial air-blast atomizer using back-lit imaging. Our method is able to accurately reconstruct and predict the flow while preserving the sharpness of the liquid-gas interface.

ACKNOWLEDGMENTS

I would like to begin by thanking my advisor, Prof. Daniel J. Bodony, who guided and supported me through the ups and downs of research. Joining his lab marked a turning point in my graduate studies. He gave me the freedom to pick my topics of research and was always available to meet when I needed advice. He not only has a commitment to the professional development of his students, but also cares about their personal goals and mental health. I really could not have asked for a better mentor.

I would also like to thank my thesis committee, Prof. Theodore Heindel, Prof. Randy Ewoldt, Prof. Andres Goza and Prof. Laura Villafañe Roca, for their thoughtful feedback and guidance. I am additionally grateful to all the wonderful professors I have had the fortune of learning from and interacting with during my time at UIUC. Special thanks goes to the administrative staff of the Aerospace Engineering Department, particularly Staci McDannel, for helping me out when I was running late against deadlines, which happened more often than I would like to admit.

I would next like to acknowledge the support of my fellow research group members – Fabian Dettenrieder, Cory Mikida, Sandeep Murthy, Bryson Sullivan and David Fellows. Particularly, I would like to thank Alexandru Fikl for numerous stimulating conversations and listening to my annoying rants. My graduate school experience would have been very boring without the interactions with everyone in the group. I feel lucky to be part of a group with so many talented and friendly people.

The work presented in this thesis was sponsored, in part, by the Office of Naval Research (ONR) as part of the Multidisciplinary University Research Initiatives (MURI) Program, under Grant Number N00014-16-1-2617.

Lastly, I would like to thank my family for everything they have done for me. I am grateful to my parents for all the sacrifices they have made for me. Thank you Aai, for always being there to talk to me about my concerns and multiplying my happiness with your positive attitude. Thank you Dad, for always picking up my call no matter how busy you were at the time. Thank you Shakku, for tolerating me as we were growing up together. Thank you Maitreyee, for being my support away from home.

To Aai, for her love and support

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Fluid flows that are relevant for engineering applications, such as flow over an aircraft and combustion chamber fuel injection, are complex due to flow features spanning varied length-scales and coupled processes involving multiple physical phenomena. Due to this, existing analytical methods need to make simplifying assumptions and often do not provide sufficiently accurate predictions for practical applications. Even numerical simulations require solutions of very high-dimensional system of equations that challenge our fastest computers.

One way to resolve this problem is to develop approximate models of the flow, while accounting for the most relevant flow features and their interactions. A low-dimensional model that faithfully approximates the dynamics of the underlying complex flow enables computationally-efficient methods for flow analysis, control and optimization. This is the goal of reduced-order modeling, which is an active field of research in fluid mechanics. With the increasing size of flow datasets, driven by advancement in computational capabilities of computers, efficient numerical methods for simulations and experimental techniques for flow measurement, data-driven model reduction methods have recently gained a lot of traction. The reader is referred to [125] for a comprehensive review of model reduction techniques for fluid flows.

Model reduction methods rely on the fact that several high-dimensional fluid flows exhibit dominant flow features that evolve on a low-dimensional manifold. Several methods involve decomposition of the flow field into linear combination of *modes*, where each mode has a distinct spatial structure and temporal behavior. For a given set of modes, Petrov-Galerkin projection [27, 124, 151] is used to project the dynamics of the high-dimensional system onto the linear space spanned by the modes. Typically, these modes are obtained from data, one of the most popular method being Proper Orthogonal Decomposition (POD) [99]. This approach of reduced-order modeling has been successfully applied to control several fluid flows [13, 74, 137, 138]. However, these methods require complete knowledge of the equations governing the high-dimensional system and require computation of the projection coefficients, which is inconvenient for complex nonlinear systems.

An alternative is provided by system-identification or model learning methods. Early system identification methods, such as the Eigensystem Realization Algorithm (ERA) [79], autoregressive modeling [5] and Observer/Kalman filter Identification (OKID) [78], were developed only for linear systems. Fortunately, linear models can closely approximate the dynamics of fluids in certain regimes. Even in more complex cases, linear models are useful for short-term predictions. As such, traditional system-identification methods have been used extensively for flow control in previous studies [47, 65, 73–75]. For strongly nonlinear systems, where traditional methods fail, the last decade has witnessed tremendous progress in data-driven model reduction by using the Koopman operator to represent the nonlinear system as an infinite-dimensional linear system [109, 116, 126, 132, 144].

While the Koopman operator is infinite-dimensional, finite-dimensional approximations provide valuable information about the system behaviour. Moreover, since the operator is linear, it allows us to use standard linear estimation and control theory tools even for complex flows. This thesis introduces novel methods that leverage recent developments in Koopman-based reduced-order modeling techniques and provide solutions to long-standing problems in the fluids community, such as efficient feedback control and optimal sensor placement.

1.1. Thesis Structure

The thesis is composed of the following chapters.

- Chapter 2 contains the prerequisite background material for reading the thesis. We introduce the Koopman representation of dynamical systems with and without incorporating the effect of control. Data-driven methods that provide finite-dimensional approximations of the Koopman operator are presented. Then, we describe important results from linear control and estimation theory. Specifically, we present the Kalman filter for state estimation and the Linear Quadratic Regulator for feedback control of linear time-invariant systems. Finally, we provide background information about Riemannian optimization by drawing parallels with Euclidean optimization. In particular, we describe the geometry of the Grassmanian manifold, which is useful for optimization problems with orthogonality constraints, and introduce operations that would facilitate optimization on the Grassmanian manifold.

- Chapter 3 introduces a novel control-oriented data-driven reduced-order modeling method called low-rank Dynamic Mode Decomposition (lrDMD). This method generalizes the approach of Dynamic Mode Decomposition (DMD) by posing the reduced-order data-driven modeling as a rank-constrained optimization problem. We present a subspace projection-based method and a Riemannian optimization method to solve this problem. We show that the proposed reduced-order model outperforms existing methods for control applications, such as suppressing the vortex shedding in the wake of an inclined flat plate in uniform flow.

- Chapter 4 presents a data-driven sensor placement framework, which can be used in conjunction with any reduced-order modeling method that provides a linear description of the system dynamics. We formulate the optimization problem, with an objective function that uses a data-driven approximation of the error covariance matrix and controls based on the Kalman filter equations. A gradient-based optimization method is used to find optimal sensor locations where the gradient is computed using adjoint equations. Finally, results are shown for flow reconstruction, prediction and control applications.

- In Chapter 5 we show that naive implementation of existing data-driven reduced-order modeling methods is not suitable for interface tracking and reconstruction in multiphase flows. We propose a new two-step method that uses optical flow methods in conjunction with DMD to reconstruct and predict interface locations in multiphase flows using imaging data. We demonstrate our method on a real experiment of liquid jet surrounded by a coaxial airblast atomizer and visualized using back-lit imaging.

- Chapter 6 summarizes the contributions of the thesis and explores two interesting future directions for research. First, we extend the low-rank Dynamic Mode Decomposition framework to incorporate the effect of control input. Second, we propose a data-driven stochastic flow control framework using input-output hidden Markov models. The proposed feedback controller, obtained by solving a mixed integer linear program, can be used to drive the asymptotic behaviour of the system.

1.2. Published Work

The work presented in this thesis has resulted in the following proceedings and peer-reviewed publications. Several chapters of the thesis are reproduced with permission from the following papers.

Palash Sashittal, Daniel J. Bodony, ″Data driven sensor placement for fluid flows″, *Theoretical and Computational Fluid Dynamics* Volume 35 (2021): (pp. 709–729).

Palash Sashittal, Daniel J. Bodony, ″Reduced-order control using low-rank dynamic mode decomposition″, *Theoretical and Computational Fluid Dynamics* Volume 33.6 (2019): (pp. 603-623).

Palash Sashittal, Daniel J. Bodony, ″Low-rank dynamic mode decomposition using Riemannian manifold optimization″, In *IEEE Conference on Decision and Control (CDC) 2018* (pp. 2265-2270).

Daniel J. Bodony, Palash Sashittal, Aaron Towne, ″Low-rank modeling of primary atomization″, In *proceedings of Center of Turbulence Research Summer Program (CTRSP) 2018*.

This research has also produced the following conference publications.

Palash Sashittal, Daniel J. Bodony, ″Data-driven sensor placement for fluid flows″, Invited talk at *AIAA Aviation 2021*.

Palash Sashittal, Daniel J. Bodony, ″Data-driven reduced order control for partially observed fluid systems″, In *AIAA Scitech 2020*.

Palash Sashittal, Daniel J. Bodony, ″Model reduction of primary atomization using optical flow″, In *ILASS-Americas 2019*.

Michael Banks, Palash Sashittal, Daniel J. Bodony, ”Towards data-driven control of multiphase flows”, In *14th International Conference on Liquid Atomization and Sprays 2018*.

Data and Code Availability

The code and data generated for the published work presenting in this thesis is open-source and available online. A Matlab code that performs low-rank Dynamic Mode Decomposition (Chapter 3) is provided at https://bitbucket.org/sashitt2/lrdmd/src/master. A Matlab code for data-driven sensor placement in fluid flows (Chapter 4) is made available at https://github.com/sashitt2/optimal_sensing.

CHAPTER 2

BACKGROUND

This chapter contains background material for the rest of the thesis. In Section 2.1 we introduce the Koopman representation of dynamical systems and data-driven methods that approximate the Koopman operator. Section 2.2 describes important results from linear estimation and control theory that are used for flow estimation and control in the subsequent chapters. Finally, Section 2.3 describes Reimannian optimization as a generalization of Euclidean optimization. As an example, we provide details about optimization of Grassmanian manifolds. This optimization paradigm is employed in the later chapters to solve optimization problems on matrix manifolds.

2.1. Koopman Theory

Dynamical systems are typically modeled as the evolution of a system in state space. Specifically, the state lies in a vector space or on a nonlinear manifold within a vector space, and evolves in time. This is described either by differential equations or discrete-time maps. An alternative approach was proposed by Koopman [87] in 1931 which involved describing the evolution of functions on the state space. This description had many advantages, since the function space is a vector space and the evolution is described by a linear operator, now known as a the Koopman operator. This approach was first applied to fluid flows in the work of Mezič [108] to decompose the evolution of the flow field onto the eigenfunctions of the Koopman operator.

Over the last decade, following the work of Schmid [132] and Rowley *et al.* [126], several methods have been developed to construct data-driven finite-dimensional approximations of the Koopman operator. In this section we will provide an introduction to Koopman theory for systems with and without control, and some of the most commonly used data-driven methods to approximate the Koopman operator. For a thorough review of this topic the reader is referred to [109, 116].

2.1.1. Koopman Representation of Dynamical Systems

Consider a discrete autonomous dynamical system evolving on the set $\mathcal{Q}$ as follows.

$$\mathbf{q}_{k+1} = f(\mathbf{q}_k), \tag{2.1}$$

where k is the index for time and $\mathbf{q}_k \in \mathcal{Q}$. The dynamics on $\mathcal{Q}$ is defined by the (possibly nonlinear) function $f : \mathcal{Q} \to \mathcal{Q}$. The Koopman operator $\mathcal{K}$ is a linear operator that completely characterizes the dynamics defined on $\mathcal{Q}$. While the linearity of the Koopman operator is favorable for analyzing the dynamical system, it comes at a cost. The Koopman operator can be infinite dimensional even for a finite dimensional dynamical system. Nevertheless, finite dimensional approximations of the Koopman operator provide valuable information about the dynamical system.

The Koopman operator $\mathcal{K}$ acts on scalar-valued functions $g : \mathcal{Q} \to \mathbb{R}$ of $\mathbf{q}$ as follows.

$$\mathcal{K}g(\mathbf{q}_k) = g(\mathbf{q}_{k+1}) = g(f(\mathbf{q})).$$

As such, the Koopman operator is also known as the composition operator since the action of $\mathcal{K}$ on $g(\cdot)$ produces the function $g \circ f(\cdot)$. The action of the Koopman operator can be easily generalized to vector-valued functions $g = [g_1, \ldots, g_N]$, where $g_i : \mathcal{Q} \to \mathbb{R}$.

Example 2.1.1. Consider the nonlinear dynamical system,

$$y_{k+1} = y_k^2. \tag{2.2}$$

The dynamical system above is nonlinear due to the quadratic term y_k^2. We use Carleman Linearization [28] to find a linear embedding of the dynamical system. We introduce an extended state vector $\mathbf{x}$ with the i-th component $x^{(i)}$ is a function g_i of y defined as

$$g_i(y) = \exp(2^{(i-1)} \ln y).$$

The dynamical system in Eq. (2.2) can be represented as,

$$\begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \end{bmatrix}_{k},$$

which is an infinite dimensional linear system. The spectral properties of the infinite dimensional state transition matrix provides insights into the stability behaviour of the dynamical system [109].

While in general the Koopman operator is infinite dimensional, for some systems, it is possible to identify a set of independent functions $\{g_1, \dots, g_N\}$ that form an invariant subspace under the action of the Koopman operator. Specifically, if $\mathcal{T}$ is the invariant subspace spanned for the function $\{g_1, \dots, g_N\}$, then $\mathcal{K}g \in \mathcal{T}$ whenever $g \in \mathcal{T}$. The action of the Koopman operator can be represented exactly by a finite-dimensional linear mapping when restricted to the invariant subspace. Let us look at an example of such a system.

Example 2.1.2. Consider the following 2-dimensional dynamical system.

$$\begin{aligned} y^{(1)}_{k+1} &= \mu_1 y^{(1)}_k, \\ y^{(2)}_{k+1} &= \mu_2 y^{(2)}_k + (\mu_1^2 - \mu_2) c (y^{(1)}_k)^2, \end{aligned}$$

where y is the state vector, the superscripts in parenthesis denote the component of the state vector, subscript denotes the time index and μ_1, μ_2 and c are scalar parameters. This system has an equivalent linear Koopman representation as follows,

$$\begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{bmatrix}_{k+1} = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & (\mu_1^2 - \mu_2)c \\ 0 & 0 & \mu_1^2 \end{bmatrix} \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{bmatrix}_{k},$$

where $x^{(1)} = g_1(\mathbf{y}) = y^{(1)}$, $x^{(2)} = g_2(\mathbf{y}) = y^{(2)}$ and $x^{(3)} = g_3(\mathbf{y}) = (y^{(1)})^2$. The nonlinear dynamics of $\mathbf{y}$ is captured by the linear dynamics of the nonlinear functions $\{g_1, g_2, g_3\}$ of $\mathbf{y}$.

Identification of such invariant subspaces is very useful, as they provide coordinates in which the system evolves linearly. As such, these coordinates can be viewed as global linearization.

Unfortunately such coordinate transformations are difficult to find and do not even exist for certain systems [86].

While a Koopman representation allows us to use linear control theory and estimation tools for nonlinear dynamical systems, traditionally, it has been used only on uncontrolled dynamical systems. Recently there have been developments in extending the Koopman framework to controlled dynamical systems which we will describe in the following subsection.

2.1.2. Incorporating the Effect of Control

There are several ways in which the Koopman framework can be extended to incorporate the effect of control on the dynamical system [80, 85, 88, 120]. Here, we show the approach described in [88] to define the Koopman operator for controlled systems. Consider a discrete non-autonomous dynamical system forced by control input from the set $\mathcal{U}$ as,

$$\mathbf{q}_{k+1} = h(\mathbf{q}_k, \mathbf{u}_k), \tag{2.3}$$

where k is the index of time, $\mathbf{q}_k \in \mathcal{Q}$, $\mathbf{u}_k \in \mathcal{U}$ and $h(\cdot)$ describes the dynamics of the system in terms of both the state $\mathbf{q}$ and control input $\mathbf{u}$. Let $\ell(\mathcal{U})$ denote the set of all sequences $\{\mathbf{u}\} := (\mathbf{u}_k)_{k=0}^{\infty}$ with $\mathbf{u}_k \in \mathcal{U}$ and $\{\mathbf{u}\}_i$ denote the i-th element in the sequence $\{\mathbf{u}\}$.

We introduce the function $H : \mathcal{Q} \times \ell(\mathcal{U}) \rightarrow \mathcal{Q} \times \ell(\mathcal{U})$ which describes the dynamics of the extended state $\boldsymbol{\chi} = (\mathbf{q}, \{\mathbf{u}\}) \in \mathcal{Q} \times \ell(\mathcal{U})$ as follows.

$$\boldsymbol{\chi}_{k+1} = H(\boldsymbol{\chi}_k) = (h(\mathbf{q}_k, \{\mathbf{u}\}_0), \mathcal{S}\{\mathbf{u}\}),$$

where $\mathcal{S}$ is the left shift operator, such that $\mathcal{S}\{\mathbf{u}\}_i = \{\mathbf{u}\}_{i+1}$ for any positive integer i. The Koopman operator $\mathcal{K}$ acts on function $g : \mathcal{Q} \times \ell(\mathcal{U}) \rightarrow \mathbb{R}$ such that,

$$\mathcal{K}g(\boldsymbol{\chi}_k) = g(\boldsymbol{\chi}_{k+1}) = g(H(\boldsymbol{\chi}_k)).$$

Note that the extended state vector $\boldsymbol{\chi}$ is infinite dimensional. This is an added complexity compared to the uncontrolled case in which the Koopman operator $\mathcal{K}$ was infinite dimensional, but the state $\mathbf{q}$ was finite dimensional. The dimensionality of $\boldsymbol{\chi}$ has implications in tractability of data-driven approximation methods which will be discussed in the next section.

2.1.3. Data-driven Approximation of the Koopman Operator

In this section, we delve into data-driven methods for approximating the Koopman operator. There are several methods that provide a data-driven approximation of the Koopman operator. Most methods rely on Dynamic Mode Decomposition (DMD) [34, 132] or one of its variants [34, 37, 48, 57, 63, 64, 129, 154] to approximate the Koopman operator. A common theme in these methods is to project the infinite-dimensional Koopman operator onto a finite-dimensional subspace that is learned from data. As such, the development of data-driven methods for Koopman operator approximation are closely tied with data-driven reduced-order modeling of fluid flows. Typically, due to the multiphysics and multiscale nature of fluid flows, accurate prediction of flow features requires numerical simulation of high-dimensional systems. Apart from finding a finite-dimensional approximation of the Koopman operator, several data-driven flow modeling methods aim to find a low-dimensional representation of the flow dynamics. All the methods that we will discuss in this section provide a low-dimensional subspace and a linear description of the flow dynamics restricted to that subspace from data.

Dynamic Mode Decomposition [132]

DMD is a data-driven method that provides a linear representation A of the dynamics of an autonomous system projected onto the space spanned by the proper orthogonal decomposition (POD) modes L of the data snapshots. Let $\{\mathbf{q}_0, \dots, \mathbf{q}_n\}$ be a set of $n+1$ snapshots of the dynamical system. By Eq. (2.1), the snapshots are related as

$$\mathbf{q}_{k+1} = f(\mathbf{q}_k),$$

The DMD algorithm is as follows.

1. Construct the data matrices $X, Y \in \mathbb{C}^{m \times n}$ as follows.

$$X := [\mathbf{q}_0 \mid \cdots \mid \mathbf{q}_{n-1}], \quad Y := [\mathbf{q}_1 \mid \cdots \mid \mathbf{q}_n].$$

2. Perform a rank r reduced singular value decomposition of the matrix X such that,

$$X \approx U\Sigma V^H,$$

where $U \in \mathbb{C}^{m\times r}$, $\Sigma \in \mathbb{C}^{r\times r}$ and $V \in \mathbb{C}^{n\times r}$.

3. The DMD operator $A \in \mathbb{C}^{r\times r}$ and the orthonormal basis of the r-dimensional subspace L is given by

$$\begin{aligned} A &:= U^H Y V \Sigma^{-1}, \\ L &:= U. \end{aligned} \tag{2.4}$$

Dynamic Mode Decomposition with Control [119]

This section describes the extension of DMD to controlled systems. In this case, we are given a set of full-state observations $\{\mathbf{q}_0, \dots, \mathbf{q}_n\}$ and control inputs $\{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\}$. DMDc generates a linear model for the system dynamics,

$$\mathbf{q}_{k+1} = M\mathbf{q}_k + B\mathbf{u}_k,$$

with state transition matrix $M = LAL^T$ of rank r, where A describes the dynamics of the system on the low dimensional subspace spanned by the columns of orthogonal matrix L and B is the actuation matrix. The algorithm is as follows.

1. Construct the data matrices $X, Y \in \mathbb{C}^{m\times n}$ as

$$X := [\mathbf{q}_0 \mid \cdots \mid \mathbf{q}_{n-1}], \quad Y := [\mathbf{q}_1 \mid \cdots \mid \mathbf{q}_n].$$

Similarly, construct matrix $\Gamma \in \mathbb{C}^{p\times n}$ with the control inputs,

$$\Gamma := [\mathbf{u}_0 \mid \cdots \mid \mathbf{u}_{n-1}].$$

Also, construct the following stacked data matrix $\Omega \in \mathbb{C}^{(m+p)\times n}$

$$\Omega := \begin{bmatrix} X \\ \Gamma \end{bmatrix}.$$

2. Perform a rank $r' > r$ singular value decomposition of the matrix Ω such that

$$\Omega \approx \widehat{U}\widehat{\Sigma}\widehat{V}^H.$$

3. Perform a rank r singular value decomposition of the matrix Ω such that

$$X \approx U\Sigma V^H.$$

4. The reduced-order model A, the orthonormal basis of the low-dimensional subspace L and the actuation matrix B are given by

$$\begin{aligned} &A := U^H Y \widehat{V}\widehat{\Sigma}^{-1}\widehat{U}_1^H U, \quad B := U U^H Y \widehat{V}\widehat{\Sigma}^{-1}\widehat{U}_2^H, \\ &L := U, \end{aligned}$$

where $\widehat{U}_1 \in \mathbb{C}^{m\times r'}$ and $\widehat{U}_2 \in \mathbb{C}^{p\times r'}$ such that $\widehat{U}^H = [\widehat{U}_1^H \mid \widehat{U}_2^H]$.

Extended Dynamic Mode Decomposition

In this subsection, we describe an extension of DMD called extended dynamic mode decomposition (EDMD) [152]. Apart from the set of data snapshots $\{\mathbf{q}_0, \ldots, \mathbf{q}_n\}$, EDMD requires a *dictionary of observables*, $\mathcal{D} := \{\psi_1, \ldots, \psi_p\}$ where, for each $i \in \{1, \ldots, p\}$, $\psi_i \in L^2(\mathbb{R}^n)$, i.e. ψ_i is a square integrable function of the state variable $\mathbf{q}$. While there are many pragmatic choices that have been suggested, the optimal choice of dictionary elements remains an open question [48, 125, 152].

DMD is a special case of Extended DMD in the sense that, EDMD is equivalent to DMD when the dictionary of observables is chosen such that $\psi_i(\mathbf{q}) = q_i$. Algorithmically we can apply the same method as DMD by constructing the following data matrices,

$$X := [\boldsymbol{\psi}(\mathbf{q}_0) \mid \cdots \mid \boldsymbol{\psi}(\mathbf{q}_{n-1})], \quad Y := [\boldsymbol{\psi}(\mathbf{q}_1) \mid \cdots \mid \boldsymbol{\psi}(\mathbf{q}_n)],$$

where $\boldsymbol{\psi}(\mathbf{q}) = [\psi_1(\mathbf{q}), \cdots, \psi_p(\mathbf{q})]^T$. The authors in [152] showed that the EDMD approach can be viewed as a spectral collocation method for approximating the Koopman operator with respect to the basis determined by the set of observables $\{\psi_1, \cdots, \psi_p\}$. Like any spectral method, the accuracy of the approximation relies on the choice and size of the basis. However, the number of basis functions can be prohibitively large for high-dimensional dynamical systems. For instance, if quadratic functions of the state $\mathbf{q}$ are considered, the number of observables will be $O(m^2)$. An alternative is to use the kernel method proposed in [152], which is efficient for cases where the number of degrees of freedom m is much larger than the number of snapshots n.

2.2. Linear Estimation and Control Theory

Although most fluid flows of interest exhibit non-linear dynamics, several studies have shown the effective implementation of linear estimation and control theory for fluid flows [12, 56, 67, 82, 83]. This is attributed to faithful representation of input, output and the important dynamic processes of the fluid flows by linearization of the underlying dynamics for many applications [83]. For example, nonlinear instability leading to laminar-to-turbulence transition in fluid flows is often preceded by linear amplification disturbances in the system, which can be mitigated by linear control techniques. This section describes optimal estimation and control methods that are applicable to linear time invariant (LTI) systems under certain conditions.

Consider the following LTI system.

$$\begin{aligned} \mathbf{q}_{k+1} &= M\mathbf{q}_k + B\mathbf{u}_k + \mathbf{w}_k, \qquad (2.5) \\ \mathbf{z}_k &= C\mathbf{q}_k + \mathbf{v}_k, \end{aligned}$$

with time indexed by k and initial condition $\mathbf{q}_0$. At each iteration k, $\mathbf{q}_k \in \mathbb{C}^m$ is the state variable, $\mathbf{u}_k \in \mathbb{C}^d$ is the control input and $\mathbf{z}_k \in \mathbb{C}^p$ is the observation of the system. The matrices $B \in \mathbb{C}^{m\times d}$ and $C \in \mathbb{C}^{p\times m}$ are called the control and the observation matrix, respectively. $\mathbf{w}_k$ and $\mathbf{v}_k$

are random variables that denote the noise in the dynamics of the system and the process of observation. As is standard for linear control theory [93], we assume that $\mathbf{w}_k$ and $\mathbf{v}_k$ are zero mean Gaussian variables with covariance matrices $W \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{p\times p}$ respectively. Note that the observers and controllers described in this section are optimal only for systems with zero-mean additive Gaussian noise.

In this study, we are particularly interested in the application of linear estimation and control theory for reduced-order models of fluid flows. As such, we assume that the dynamics of the system can be accurately approximated on a low-dimensional linear subspace. Under this assumption, we can focus on the dynamics of the system described in Eq. (2.5) restricted to the r-dimensional space spanned by the columns of an orthonormal matrix $L \in \mathbb{C}^{m\times r}$ (i.e. $L^H L = I_r$, where I_r is the $r \times r$ identity matrix). The dynamics of the system restricted to the r-dimensional subspace is

$$\begin{aligned} \mathbf{a}_{k+1} &= A\mathbf{a}_k + L^H B\mathbf{u}_k + L^H \mathbf{w}_k, \\ \mathbf{z}_k &= CL\mathbf{a}_k + \mathbf{v}_k, \end{aligned} \tag{2.6}$$

where $\mathbf{a} \in \mathbb{C}^r$ is the projection of the state variable $\mathbf{q}$ onto the space spanned by the columns of L, $A = L^H ML$ and the initial condition is $\mathbf{a}_0 = L^H\mathbf{q}_0$. Note that $L^H\mathbf{w}_k$ is a Gaussian random variable with zero mean and covariance matrix given by $L^H WL$.

Remark 2.2.1. When the full-order model (Eq. (2.5)) is not known and only state trajectories of the system are available, reduced-order model (Eq. (2.6)) can be learned using data-driven methods, such as those described in Section 2.1.3.

2.2.1. Kalman Filter

Let the estimate of $\mathbf{a}_k$ given observations $\mathbf{z}^\ell := \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_\ell\}$ be denoted by $\mathbf{a}_{k|\ell}$. The estimate for the full-state variable $\mathbf{q}_k$ will be similarly denoted as $\mathbf{q}_{k|\ell} = L\mathbf{a}_{k|\ell}$. The standard estimation problem for linear state space models is determining the estimate of $\mathbf{a}_k$ given $\mathbf{z}^k$, denoted by $\mathbf{a}_{k|k}$, such that

$$E[(\mathbf{a}_k - \mathbf{a}_{k|k})^2] \leq E[(\mathbf{a}_k - \widetilde{\mathbf{a}}_{k|k})^2],$$

for any causal estimate $\widetilde{\mathbf{a}}_{k|k}$. Using $\mathbf{a}_{k|\ell} = E[\mathbf{a}_k|\mathbf{z}^\ell]$ and $\Sigma_{k|\ell} = \text{var}\,(\mathbf{a}_k|\mathbf{z}^\ell)$, we get the following Kalman filter [148] forward recursion

$$\begin{aligned}
\mathbf{a}_{0|0} &= E[\mathbf{a}_0], \\
\mathbf{a}_{k+1|k} &= A\mathbf{a}_{k|k}, \\
\mathbf{a}_{k+1|k+1} &= \mathbf{a}_{k+1|k} + K_{k+1}(\mathbf{z}_{k+1} - CL\mathbf{a}_{k+1|k}),
\end{aligned}$$

with the following recursive *offline* computations

$$\begin{aligned}
\Sigma_{0|0} &= \text{var}(\mathbf{a}_0), \\
\Sigma_{k+1|k} &= A\Sigma_{k|k}A^H + L^H W L, \\
\Sigma_{k+1|k+1} &= \Sigma_{k+1|k} - \Sigma_{k+1|k}L^H C^H (CL\Sigma_{k+1|k}L^H C^H + V)^{-1} C\Sigma_{k+1|k}, \\
K_{k+1} &= \Sigma_{k+1|k+1}L^H C^H V^{-1}.
\end{aligned}$$

Here $K_k \in \mathbb{C}^{r\times p}$ is the time varying Kalman gain matrix and $\Sigma_k \in \mathbb{C}^{r\times r}$ is the covariance matrix of $\mathbf{a}_k$ and var$(\cdot)$ denotes the variance. The recursion for $\Sigma_{k|k-1}$ and $\Sigma_{k|k}$ has the form of the Riccati equation. This recursion is referred to as *offline* because $\Sigma_{k|k-1}$, $\Sigma_{k|k}$ and K_k only depend on the reduced-order model A, observation matrix C and the basis L, and not on the observations $\mathbf{z}^k$. The predicted observation is obtained by the predicted state estimate by

$$\mathbf{z}_{k|k-1} = CL\mathbf{a}_{k|k-1}$$

and the estimated error covariance matrix $S_k \in \mathbb{C}^{p\times p}$ is given by

$$\begin{aligned}
S_k &= E[(\mathbf{z}_k - \mathbf{z}_{k|k-1})(\mathbf{z}_k - \mathbf{z}_{k|k-1})^H] \\
&= C\Sigma_{k|k-1}C^H + V.
\end{aligned}$$

For large values of n, the recursion for $\Sigma_{k|k-1}$ converges to a steady-state value Σ, provided that (C, A) is observable and (A, W) is controllable [25]. In that case, Σ satisfies the following Algebraic Riccati Equation (ARE),

$$\Sigma = A\Sigma A^H + L^H W L - A\Sigma L^H C^H (CL\Sigma L^H C^H + V)^{-1} CL\Sigma A^H. \tag{2.7}$$

The resulting steady-state Kalman gain matrix is

$$K = \Sigma L^H C^H (CL\Sigma L^H C^H + V)^{-1}. \tag{2.8}$$

The estimate of the hidden variable using the steady-state Kalman gain matrix is generally used when the time horizon for the control is large relative to the effective convergence time of the Riccati equation.

2.2.2. Observer-based Feedback Controller

The Kalman filter-based observer with the optimal sensor locations provides filtered estimates, $\mathbf{q}_{k|k}$, of the state variable, $\mathbf{q}_k$, for a given reduced-order model at each iteration k. We can use this estimate to perform feedback control of the flow using the Linear Quadratic Regulator (LQR) framework [10].

We use the following updates for the estimate $\mathbf{a}_{k|k}$ to find the control $\mathbf{u}_k$ at iteration k,

$$\begin{aligned} \mathbf{a}_{k+1|k+1} &= \mathbf{a}_{k+1|k} + K(\mathbf{z}_{k+1} - CL\mathbf{a}_{k+1|k}), \\ \mathbf{u}_k &= -G\mathbf{a}_{k|k}, \end{aligned}$$

where $G \in \mathbb{C}^{d \times r}$ is feedback gain matrix and the Kalman gain matrix K is computed using Eq. (2.8). We compute the feedback gain matrix G to minimize the following objective function,

$$J_c = E\left[\sum_{k=0}^{\infty} (\mathbf{a}_k^H \mathbf{a}_k + \beta \mathbf{u}_k^H \mathbf{u}_k)\right], \tag{2.9}$$

where $E[\cdot]$ denotes the expectation and $\beta \in \mathbb{R}^+$ determines the trade-off between minimizing the perturbation and the control input in the reduced-order system. Similar to the steady-state

Kalman equations (Eq. (2.7)), we solve a discrete algebraic Riccati equation,

$$P = A^H P A - (A^H P L^H B)(\beta I_d + B^H L P L^H B)^{-1}(B^H L P A) + I_r, \tag{2.10}$$

for $P \in \mathbb{C}^{r \times r}$ and set $G = (\beta I_d + B^H L P L^H B)^{-1} B^H L P A$. Although most fluid flow systems are nonlinear, the LQR is a powerful tool used for stabilization of the flow when a feedback controller can be employed [12, 56, 67, 82, 83].

Remark 2.2.2. As described above, for an observer-based reduced-order feedback controller, the control input at iteration k is given by $\mathbf{u}_k = -G\mathbf{a}_{k|k}$. However, if the full-state $\mathbf{q}_k$ is available, then the same control gain matrix G can be used to describe the control $\mathbf{u}_k = -GL^T\mathbf{q}_k$ where L is the reduced-order basis.

2.3. Riemannian Optimization

Optimization on smooth manifolds, specifically on matrix manifolds, arise in several engineering applications. In the last decade, there have been several theoretical advances in generalization of standard methods for unconstrained optimization to solve the broader class of optimization problems on smooth manifolds. Several software packages for optimization on manifolds [17, 24, 143] have also become available. Here we describe the procedure for Riemannian optimization while drawing parallels to Euclidean optimization. A comprehensive review on Riemannian optimization, with particular focus on matrix manifold optimization is given in [2]. In this section we will concisely show that Riemannian optimization is a generalization of Euclidean optimization on curved manifolds.

Consider optimization in Euclidean space $\mathcal{E}$, where at iteration k we are at point x_k and need to update our position in some direction in the tangent space, $dx_k \in \mathcal{T}_{x_k}\mathcal{E}$. This direction is called the search direction. Typically, in gradient-based algorithms, the search direction dx_k is taken along the derivative of the objective function at the point x_k. However, in Riemannian space, the derivative of the objective function, in general, might not lie in the tangent space of manifold at the point x_k. Therefore, we need to consider the projection of the derivative on the tangent plane to find the search direction dx_k. Figure 2.1 shows a schematic to describe this operation.

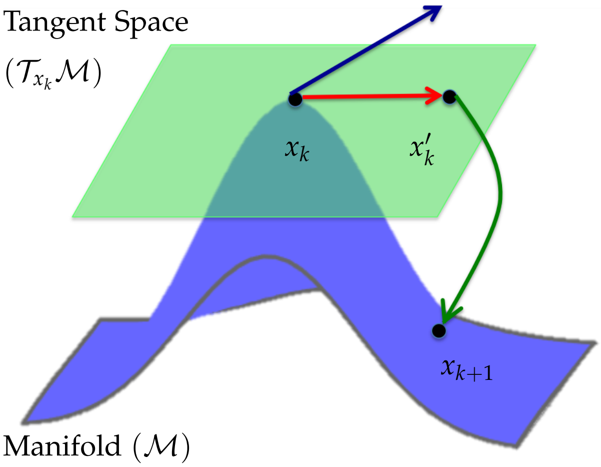


Figure 2.1.: Schematic of the Riemannian optimization method on a curved manifold.

The tangent space is shown in green while the curved surface manifold is shown in blue. The Euclidean gradient (blue arrow) is outside the tangent space and needs to be projected onto the tangent space depicted by the red arrow in the schematic.

The update rule, from x_k to x_{k+1}, for the optimization in the Euclidean and the Riemannian spaces is also different. In Euclidean optimization, the tangent space at some point x_k is same as the ambient space with the origin translated to that point. Therefore, the update rule is simply $x_{k+1} = x_k + \eta dx_k$ with the new point at iteration $k+1$ as $x_{k+1} \in \mathcal{E}$, as discussed before. Now consider the same case with $x_k \in \mathcal{M}$ where $\mathcal{M}$ is some Riemannian manifold and search direction $dx_k \in \mathcal{T}_{x_k}\mathcal{M}$. With the same update rule, $x_k + dx_k$ might not reside in $\mathcal{M}$ due to the curvature of the manifold. Therefore, we need a way to move onto the manifold from the tangent space. This is shown in Figure 2.1 by the green arrow from $x'_k \in \mathcal{T}_{x_k}\mathcal{M}$ to $x_{k+1} \in \mathcal{M}$. This operation is known as a *Retraction*. For a review on retractions the reader is referred to [2].

In the following section, we will characterize the metric, tangent space and normal space of the Grassmanian manifold. This will be used in solving optimization problems on the manifold of fixed-rank matrices.

2.3.1. Grassmanian Manifold

Grassmanian manifold $\mathcal{G}_{r,m}$ is a manifold in which each point is a linear subspace of rank r embedded in an m-dimensional Euclidean space. Note that this different than the Stiefel manifold

$\mathcal{S}_{m,r}$, where each point is a $m \times r$ orthonormal matrix. Let $\mathcal{O}_r$ be the set of $r \times r$ orthonormal matrices, also known as the *special orthogonal group*.

We can represent each point on the Grassmanian manifold as an orthonormal $m \times r$ matrix L whose columns span the corresponding subspace. An important observation is that the matrix representation L of any point on the Grassmanian manifold is not unique. Specifically, for any orthonormal matrix L, all matrices LO where $O \in \mathcal{O}_r$, equivalently represent the same point. Each point corresponds to the equivalent class, defined by the equivalence relation $\sim$, as follows.

$$[L] = \{L' \in \mathcal{S}_{m,r} : L' \sim L\} = \{LO : O \in \mathcal{O}_r\}.$$

As such, the Grassmanian manifold can be viewed as a quotient space arising from the Stiefel manifold. In practice we will choose an arbitrary orthonormal matrix from the set $[L]$ to represent the point on the Grassmanian manifold, and construct the algorithm such that all the operators, such as projection, translation and retraction, depend on the equivalent class $[L]$ and not on the representative matrix itself.

2.3.2. Geometry of the Grassmanian Manifold

Let $\mathcal{T}_L\mathcal{G}_{m,r}$ denote the tangent place at the point $[L]$ on the manifold $\mathcal{G}_{m,r}$. We begin by defining the canonical metric $g_c : \mathcal{T}_L\mathcal{G}_{m,r} \times \mathcal{T}_L\mathcal{G}_{m,r} \to \mathbb{R}$, which assigns an inner product on the manifold $\mathcal{G}_{m,r}$ as

$$g_c(dL_1, dL_2) = tr(dL_1^T dL_2),$$

for $dL_1, dL_2 \in \mathcal{T}_L\mathcal{G}_{m,r}$. Consider the objective function $F(L)$ defined on the Grassmanian manifold where $L \in \mathcal{G}_{m,r}$. Let $\nabla F(L)$ and $\nabla^2 F(L)$ be the Euclidean gradient and Hessian,

$$(\nabla F(L))_{ij} = \frac{dF}{dL_{ij}}, \quad \text{and}$$
$$(\nabla^2 F(L))_{ij,kl} = \frac{d^2 F}{dL_{ij} dL_{kl}}.$$

The Riemannian gradient of the function at $[L]$ is defined as the tangent vector, $\operatorname{grad} F(L) \in$

$\mathcal{T}_L\mathcal{G}_{m,r}$ such that,

$$tr((\nabla F)^T dL) = g_c(\text{grad}\, F(L), dL) \;\forall\; dL \in \mathcal{T}_L\mathcal{G}_{m,r}.$$

This gives the following relation between Euclidean and Riemannian gradients,

$$\text{grad}\, F(L) = \nabla F(L) - LL^T \nabla F(L). \tag{2.11}$$

Using the definition of the Hessian in [44], the action of the Riemannian Hessian on any tangent vector dL can be described as follows,

$$\begin{aligned}\text{Hess}\, F(L)[dL] &= (I_m - LL^T)\nabla^2 F(L)[dL] \\ &\quad - dL(L^T \nabla F(L)).\end{aligned} \tag{2.12}$$

The exponential map on the Grassmanian manifold in the direction of dL is given by

$$L(t) = LV\cos(\Sigma t)V^T + U\sin(\Sigma t)V^T \tag{2.13}$$

where $dL = U\Sigma V^T$ is the singular value decomposition of dL and $t \in [0,1]$. There are several ways in which retraction can be performed on the Grassmanian manifold. The most common method is,

$$\text{Retr}_L(dL) = UV^T, \tag{2.14}$$

where $(L + dL) = U\Sigma V^T$ is the singular value decomposition of $(L + dL)$. Another common method of retraction uses QR-factorization instead of SVD [2].

CHAPTER 3

LOW-RANK DYNAMIC MODE DECOMPOSITION

This chapter describes a control-oriented data-driven reduced-order modeling method, lrDMD, that can be used for reduced-order feedback control of fluid flows. Section 3.1 describes the motivation for the proposed method and Section 3.2 outlines the mathematical formulation of the problem. Section 3.3 details the numerical implementation of subspace projection and gradient descent algorithms used to solve the lrDMD optimization problem, including the model reduction computational performance. The control application results are shown in Section 3.4 and Section 3.5 concludes the chapter.

3.1. Motivation

Fluid flows are governed by physics, which can be described by partial differential equations that when discretized, result in very high-dimensional systems for numerical simulations. Solving optimal control and design problems for these high dimensional systems is challenging, even on very large computers [84]. This creates a need for reduced-order models that can faithfully approximate the dynamics of the system. Model reduction and reduced-order control therefore has been an active area of research in fluid mechanics. The reader is referred to [125] for a thorough review of model reduction techniques for flow analysis and control.

Model reduction for fluid flows is often done using data available from high-fidelity numerical simulations or data measured from experiments. One common approach, called Petrov-Galerkin projection [27, 124, 151], extracts a reduced basis from the snapshots and projects the full system dynamics onto the linear subspace formed by the reduced basis. Reduced-order feedback control is then constructed using the resulting reduced-order model. This model-based feedback control strategy has been used in several studies [13, 74, 137, 138] for applications such as control of flat-plate boundary layers and suppression of vortex shedding over a circular cylinder. Although these methods can be very effective (see [83] for a thorough review), they require prior knowledge of the full-state governing equations and require computation of the projection coefficients, which is inconvenient for complex nonlinear systems.

To remedy this, non-intrusive data-driven modeling techniques learn a reduced-order model of the fluid flow directly from data, with little to no prior knowledge of the dynamics of the system. Koopman-based modeling methods, such as DMD and its variants [33, 37, 52, 63, 64], belong to this category and have been used for reduced-order control of dynamical systems with uncertain parameters [90], for suppressing flow separation [38, 62], and for optimal actuator selection for airfoil separation control [20].

Existing Koopman-based data-driven modeling methods approximate the dynamics of the system by fitting an endomorphic linear function on some low-dimensional subspace. As such, the linear approximation of the dynamical system is learned constrained to a particular low dimensional subspace. This is very useful in a model order reduction method, since it provides a low order basis to describe the flow. However, in this study, we find that this constraint is restrictive for the reduced-order control performance of these methods, specifically for non-normal systems in which dominant eigenmodes and adjoint modes do not coincide.

We investigate the possibility of using a linear map between different subspaces to construct reduced-order models and controllers for unsteady fluid flows. To do this, we formulate a rank-constrained matrix optimization problem and propose two methods to solve it. We call this model reduction method low-rank Dynamic Mode Decomposition (lrDMD) [57, 129]. We show that the flexibility of using different input and output spaces for the reduced-order model allows lrDMD to capture both the eigenmodes, which describe the dominant directions of response of the system, and the adjoint modes, which describe the dominant directions of receptivity of the system to forcing. The ability of lrDMD to capture the adjoint modes of the system enables construction of efficient data-driven feedback controllers.

3.2. Mathematical Formulation

Consider a dynamical system with the state vector $\mathbf{q} \in \mathbb{R}^m$ such that

$$\mathbf{q}_{k+1} = f(\mathbf{q}_k) \tag{3.1}$$

where the subscript denotes the time iteration number. We have access to a sequence of time snapshots of this state vector which we represent as a data matrices $X, Y \in \mathbb{R}^{m \times n}$ formed by n pairs of data snapshots as follows,

$$X \coloneqq (\mathbf{q}_0 | \cdots | \mathbf{q}_{n-1}), \quad Y \coloneqq (\mathbf{q}_1 | \cdots | \mathbf{q}_n).$$

The matrix Y is called the time-shifted data matrix associated with X. Our aim is to build a reduced-order model of the dynamical system (3.1) that captures the essential features of the function $f(\cdot)$. To this end, we construct a linear approximation of $f(\cdot)$ from the given data matrices. A natural choice of optimization problem to solve for this purpose would be

$$\hat{A} = \underset{A \in \mathbb{R}^{m \times m}}{\arg\min} \|Y - AX\|_F^2 , \tag{3.2}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\hat{A} \in \mathbb{R}^{m \times m}$ is the inferred state transition matrix. However, for fluid systems m is generally large, frequently in the range of $10^6 - 10^9$, which makes this optimization problem computationally intractable. To remedy this, low-rank versions of this problem are adopted. Two popular methods that employ this approach are Dynamic Mode Decomposition (DMD) [132, 144] (discussed in Section 2.1.3) and Optimal Mode Decomposition (OMD) [52, 154] (described in Section 3.2.1). We will also introduce a generalization of these methods which we call low-rank Dynamic Mode Decomposition (lrDMD) [57, 129].

3.2.1. Optimal Mode Decomposition

In OMD [52, 154], the following constrained version of the optimization problem described in Eq.(3.2).

$$\min_{L,M} \left\| Y - LML^T X \right\|_F^2 \tag{3.3}$$

$$\text{s.t. } L^T L = I \tag{3.4}$$

$$M \in \mathbb{R}^{r \times r}, L \in \mathbb{R}^{m \times r}, \tag{3.5}$$

where r is the rank of the linear approximation of the state transition matrix $\hat{A} = LML^T$. This particular form of low rank approximation has useful implications in reduced-order modeling. The matrix L provides the basis of a low dimensional subspace to approximate the trajectories of the dynamical system while M provides the dynamical evolution of the state vector on this subspace. The solution of M for a fixed L is given by

$$M^*(L) = L^T Y X^T L (L^T X X^T L)^{-1}.$$

The optimal subspace L is found in [52, 154] using a conjugate-gradient-based algorithm. The OMD solution coincides with the DMD approximation when L is chosen as the r leading left singular vectors of the data matrix X. In this way OMD is a generalization of the DMD formulation.

3.2.2. Low-rank Dynamic Mode Decomposition

We now consider a rank-constrained solution of the optimization problem in (3.2). Recently the authors of [57] showed that there exists a closed form solution for this optimization problem. We solve the following equivalent optimization problem,

$$\min_{L,D,R} \left\| Y - LDR^T X \right\|_F^2, \tag{3.6}$$

where $L, R \in \mathbb{R}^{m \times r}$ and $L^T L = R^T R = I_r$ ($r \times r$ identity matrix) and $\hat{A} = LDR^T$ is the r-ranked matrix approximating the dynamics of the underlying system. Observe that (3.6) optimizes over all possible left and right subspaces of the state transition matrix in contrast to standard DMD and OMD. A key step to solving (3.6) is to find the optimal solution for D as a function of L and R. For a fixed L and R there exists a closed form optimal solution for D, namely

$$D^*(L, R) = (L^T Y X^T R)(R^T X X^T R)^{-1}. \tag{3.7}$$

Using (3.7) in the objective function and simplifying further the optimization problem can be reduced to

$$\min_{L,R} \left(- \left\| L^T Y Q_R \right\|_F^2 \right) \tag{3.8}$$

where $Q_R = X^T R (R^T X X^T R)^{-1} R^T X$. We will denote the objective function (3.8) by $G(L, R)$. An important observation is that $G(L, R)$ is only a function of the spaces spanned by the columns of L and R. To see this consider an orthogonal matrix $P \in \mathbb{R}^{r \times r}$ such that $P^T P = I_r$. For this objective function, $G(LP, R) = G(L, R)$ and $G(L, RP) = G(L, R)$. This implies that the objective function does not change values as long as the space spanned by the columns of L and R do not change. Hence the optimization can be equivalently performed over the set of r-dimensional subspaces in $\mathbb{R}^m$ instead of the set of orthogonal matrices in $\mathbb{R}^{m \times r}$.

The set of r-dimensional subspaces in $\mathbb{R}^m$ is known as the Grassmanian manifold $\mathcal{G}_{m,r}$ [2, 44]. In matrix representation an element of $\mathcal{G}_{m,r}$ is specified by an orthogonal basis of the r-dimensional subspace. The manifold on which we perform the optimization (3.8) is a product manifold of two Grassmanian manifolds

$$\mathcal{M} := \mathcal{G}_{m,r} \times \mathcal{G}_{m,r}.$$

An element of $\mathcal{M}$, in matrix representation, will be characterized by $M = (L, R)$.

The L and R basis solutions of the optimization problem (3.8) need not be the best basis for a Petrov-Galerkin reduced-order model. These bases only provide the subspaces for the best low-rank linear description of the dynamical system from data in the least-squares sense.

3.3. Numerical Methods for lrDMD Optimization Problem

In this section we present the numerical methods employed to solve the lrDMD optimization problem and compare the computational performance of the methods to OMD and DMD. Recall that the objective function and the constraints are,

$$\begin{aligned}\min_{L,R} G(L,R) &:= \left(-\left\|L^T Y Q_R\right\|_F^2\right) \\ \text{s.t. } (L,R) &\in \mathcal{M} := \mathcal{G}_{m,r} \times \mathcal{G}_{m,r} \\ \text{with } Q_R &= X^T R (R^T X X^T R)^{-1} R^T X.\end{aligned} \tag{3.9}$$

We propose two methods to solve this optimization problem. The first method is a computationally efficient subspace projection method that provides an approximate solution to the problem. The second method is a gradient-based method that is guaranteed to converge to a local minimum.

Note that there is no guarantee that the gradient based method will converge to the global minimizer of the given optimization problem (3.9). The gradient based method can therefore be thought as a way to improve the initial guess provided by either DMD or the subspace projection method that will be described in the following subsection.

3.3.1. Subspace Projection Method

In this method, we use iterative subspace projection to find a good approximation of the optimal solution. We first make the observation that Q_R is an orthogonal projection matrix in the column space of $X^T R$. This means that there exists an orthogonal matrix C_R such that $C_R C_R^T = Q_R$ and $\text{Im}(C_R) = \text{Im}(Q_R)$. Substituting Q_R with $C_R C_R^T$, we get the following cost function,

$$G(L,R) = -\left\|L^T Y C_R\right\|_F^2,$$

with the constraint that $\text{Im}(C_R) = \text{Im}(Q_R)$. Note that $L^T L = C_R^T C_R = I_r$. We make the following observations,

1. If C_R is fixed, the optimal L is given by an orthogonal basis of the subspace spanned by the r left singular vectors of YC_R with the highest singular values

2. If L is fixed, the optimal C_R is given by an orthogonal basis subspace spanned by the r left singular vectors of $Y^T L$ with the highest singular values

To show the first observation, consider the case when C_R is fixed. Let $YC_R = N$. By using the definition of Frobenius norm we get,

$$\left\| L^T N \right\|_F^2 = tr(L^T N N^T L).$$

The maximum trace is obtained only if L is an orthogonal basis of the space spanned by the eigenvectors of NN^T corresponding to the largest eigenvalues. This is the same as the left singular vectors corresponding to the largest singular values of N. A similar argument goes for the case when L is fixed.

Observe that since C_R is given by the left singular vectors of $Y^T L$ by the optimality condition, we have $\text{Im}(C_R) = \text{Im}(Y^T L)$. But the constraint is that $\text{Im}(C_R) = \text{Im}(X^T R)$. If R is found such that $X^T R \approx Y^T L$, both the optimality condition and the constraint are satisfied. Therefore, we find R by solving the following optimization problem

$$\min_R \left\| Y^T L - X^T R \right\|_F^2.$$

This is known as the 'Orthogonal Procrustes problem' [135]. The solution to this problem is given by $R = UV^T$ where U, V are the left and the right singular vectors of $XY^T L$. Therefore,

$$[U, V] = \text{SVD}(XY^T L)$$
$$R = UV^T.$$

Remark 1: Even though R is solution of an optimization problem, C_R may not satisfy its optimality condition exactly. However, the constraint for the optimization is always exactly satisfied.

Remark 2: If the optimal L does not differ excessively from the left singular vectors of A and B, then the optimal solution is given by $R = L$ and we get back the Optimal Mode Decomposition [52] solution.

The solution provided by this algorithm relies heavily on the initial guess. We only need an initial guess for R or, effectively, C_R. In our study, we choose the r leading left singular vectors of X as the initial guess for R. Note that this initial guess is the same as the projection subspace used in DMD. Due to the optimal choice of L for a fixed C_R, the chosen initial guess ensures that the algorithm provides a solution with a reconstruction error at most as high as the error in DMD reconstruction of the same rank.

Algorithm 1 Subspace projection method

```
Require: $Y \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{m \times n}$
Guess initial $L_0$,$R_0$ and compute $C_{R_0} \leftarrow \Pi(X^T R_0)$
$k = 0$
repeat
    $L_{k+1} \leftarrow \Pi(Y C_{R_k})$
    $R_{k+1} \leftarrow \arg\min \left\| Y^T L_k - X^T R \right\|_F^2$
    $C_{R_{k+1}} \leftarrow \Pi(X^T R_{k+1})$
    $\epsilon \leftarrow (G(L_{k+1}, R_{k+1}) - G(L_k, R_k))/G(L_k, R_k)$
    $k \leftarrow k + 1$
until $\epsilon \leq$ threshold
$D = (L_k^T Y X^T R_k)(R_k^T X X^T R_k)^{-1}(R_k^T L_k)$
return $L_k, D, R_k$
```

3.3.2. Gradient Descent Method

In this section, we describe the Riemannian gradient descent method employed to solve the optimization problem. For a thorough review on Riemannian optimization on the Grassmanian manifold the reader is referred to [44] and the presentation in Section 2.3.1.

Geometry of $\mathcal{M}$ with its Canonical Metric

We define the Riemannian metric on the manifold $\mathcal{M}$ as

$$g_c((dL_1, dR_1), (dL_2, dR_2)) = tr(dL_1^T dL_2) + tr(dR_1^T dR_2)$$

for $(dL_1, dR_1), (dL_2, dR_2) \in \mathcal{T}_{L,R}\mathcal{M}$. This metric is similar to the Euclidean metrics because for Grassmanian manifolds the Euclidean metric and the canonical metric are the same (see Section 2.3.1). In the following section, we define the gradients and Hessians of objective function $G(L, R)$ in (3.8).

Euclidean Gradient and Hessian

We first find the Euclidean derivative and Hessian of $G(L, R)$ with respect to the matrices L and R. The objective function G can also be written as

$$G = -tr(Q_R Y^T L L^T Y).$$

The gradient of G with respect to L is given by

$$\frac{\partial G}{\partial L} = -2 Y Q_R Y^T L.$$

The gradient of G with respect to R is

$$\begin{aligned}\frac{\partial G}{\partial R} &= -\frac{\partial tr(Q_R Y^T L L^T Y)}{\partial R} \\ &= -\frac{\partial tr(L^T Y Q_R Y^T L)}{\partial R} \\ &= -\frac{\partial tr((L^T Y X^T R)(R^T X X^T R)^{-1}(R^T X Y^T L))}{\partial R}.\end{aligned}$$

Computing and simplifying the derivative results in the following expression,

$$\frac{\partial G}{\partial R} = -2X(P - Q_R P),$$

where $P = Y^T L L^T Y X^T R (R^T X X^T R)^{-1}$. Therefore, the Euclidean derivative of the objective function G is

$$\nabla G(L, R) = (-2 Y Q_R Y^T L, -2X(P - Q_R P)).$$

For the Hessian we use the identity

$$d(X^{-1}) = -X^{-1} d(X) X^{-1}.$$

The Euclidean Hessian comes out to be

$$\begin{aligned}\nabla^2 G(L,R)[dL,dR] = (&-2YdQ_RA^TL - 2YQ_RY^TdL,\\ &-2X((I_n - Q_R)dP - dQ_RP)).\end{aligned}$$

where dQ_R and dP are defined as follows.

$$\begin{aligned}dQ_R = {}& X^TdRS^{-1}R^TX - X^TRS^{-1}(dR^TXX^TR)S^{-1}R^TX\\ &- X^TRS^{-1}(R^TXX^TdR)S^{-1}R^TX + X^TRS^{-1}dR^TX,\end{aligned}$$

$$\begin{aligned}dP = {}& Y^TdLL^TYX^TRS^{-1} + Y^TLdL^TYX^TRS^{-1} + Y^TLL^TYX^TdRS^{-1}\\ &- Y^TLL^TYX^TRS^{-1}(dR^TXX^TR)S^{-1} - Y^TLL^TYX^TRS^{-1}(R^TXX^TdR)S^{-1},\end{aligned}$$

where $S = R^TXX^TR$.

Riemannian Gradient and Hessian

Using the Riemannian gradient definition in (2.11) we get

$$\begin{aligned}\operatorname{grad} G(L,R) = (&-2(I_m - LL^T)YQ_RY^TL,\\ &-2(I_m - RR^T)X(P - Q_RP)).\end{aligned}$$

The action of Riemannian Hessian on some tangent vector $(dL, dR) \in \mathcal{T}_{L,R}\mathcal{M}$ can be obtained using (2.12) to get,

$$\begin{aligned}\operatorname{Hess} G(L,R)[dL,dR] = &-2(I_m - LL^T)(YdQ_RA^TL + YQ_RY^TdL) + 2dL(L^TYQ_RY^TL),\\ &-2(I_m - RR^T)X((I_n - Q_R)dP - dQ_RP) + 2dR(X(P - Q_RP)).\end{aligned}$$

Using the gradient and the Hessian, we perform a second-order gradient descent method called the Trust-region algorithm on the manifold $(\mathcal{M}, g_c)$.

Trust-Region Algorithm

In the previous section we defined the gradient and Hessian of the objective function $G(L, R)$ on the manifold $\mathcal{M}$. In this section we describe the Riemannian trust-region algorithm [1] used to solve the optimization problem. At every k^{th} iteration we solve the following trust-region subproblem,

$$\min_{(dL,dR)\in\mathcal{T}_{L_k,R_k}\mathcal{M}} m_{L_k,R_k}(dL, dR) \tag{3.10}$$

such that $g_c((dL, dR), (dL, dR)) \leq \Delta_k^2$ where $m_{L_k,R_k}(dL, dR)$ is the local quadratic approximation of the objective function $G(L_k, R_k)$ given by,

$$\begin{aligned} m_{L_k,R_k}(dL, dR) &= G(L_k, R_k) + g_c(\operatorname{grad} G(L_k, R_k), (dL, dR)) \\ &\quad + \frac{1}{2} g_c(\operatorname{Hess} G(L_k, R_k)[dL, dR], (dL, dR)) \end{aligned}$$

The trust region is defined as a ball centered at 0 in $\mathcal{T}_{L_k,R_k}\mathcal{M}$ with a radius $\Delta_k > 0$. Thus, the subproblem finds the tangent vector that minimizes m_{L_k,R_k} within a ball of radius Δ_k. The solution of the trust-region subproblem (dL_k, dR_k) gives the direction of update to the next iteration. This problem is solved iteratively using the truncated conjugated gradient method. A candidate for the next iterate on manifold $\mathcal{M}$ is found using the retraction function,

$$\begin{aligned} (L_{k+1}, R_{k+1}) &= \operatorname{Retr}_{L_k,R_k}(dL_k, dR_k) \\ &= (U_L V_L^T, U_R V_R^T), \end{aligned}$$

where U_L and U_R are the left singular vectors and V_L and V_R are the right singular vectors of $L_k + dL_k$ and $R_k + dR_k$ respectively. The decision of whether to accept this candidate or reject it is based on the following ratio,

$$\rho_k = \frac{G(L_k, R_k) - G(L_{k+1}, R_{k+1})}{m_{L_k,R_k}(0) - m_{L_k,R_k}(dL_k, dR_k)}, \tag{3.11}$$

where ρ_k indicates our confidence in the accuracy of the quadratic model $m_{L,R}(dL, dR)$. If ρ_k is greater than a certain threshold ρ', we accept the candidate; otherwise it is rejected. In this study we set $\rho' = 0.1$. Further, ρ_k also informs the radius of the trust-region. A small ρ_k reduces our confidence in the trust-region and Δ_k is decreased and, likewise, if ρ_k is large and close to 1 we increase Δ_k.

If $\rho_k < 0.1$, then the quadratic model $m_{L,R}(dL, dR)$ is inaccurate, so the candidate is rejected and the trust-region radius is reduced. If $\rho_k > 0.1$, the candidate is always accepted; however, if $\rho_k < 1/4$, the radius is reduced. If $\rho_k > 3/4$ and $g_c((dL, dR), (dL, dR)) = \Delta_k^2$, then the candidate is accepted and the radius is also increased. The maximum ceiling for the trust-region radius is set to $\sqrt{r}$. Algorithm 2 shows the logical sequence of steps.

The stopping criterion for convergence is placed on the norm of the gradient of $G(L, R)$ (defined by the canonical metric on $\mathcal{M}$) with a threshold of 10^{-6}, that is $\|\text{grad}\, G\| \leq 10^{-6}$.

Algorithm 2 Trust-region algorithm

```
Require: $Y \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{m \times n}$
Guess initial $L_0$,$R_0$
$k = 0$
repeat
    Solve trust-region subproblem (3.10) to get $(dL_k, dR_k)$
    Compute $\rho_k$ from (3.11)
    if $\rho_k < 1/4$ then
        $\Delta_{k+1} = \Delta_k/4$
    else if $\rho_k > 3/4$ and $\|(dL_k, dR_k)\|_{g_c} = \Delta_k$ then
        $\Delta_k = \min(2\Delta_k, \sqrt{r})$
    else
        $\Delta_{k+1} = \Delta_k$
    end if
    if $\rho_k > 0.1$ then
        $(L_{k+1}, R_{k+1}) = \text{Retr}_{L_k,R_k}(dL_k, dR_k)$
    else
        $(L_{k+1}, R_{k+1}) = (L_k, R_k)$
    end if
    $k \leftarrow k + 1$
until $\|\text{grad}\, G(L_k, R_k)\|_{g_c} <$ threshold
$D = (L_k^T Y X^T R_k)(R_k^T X X^T R_k)^{-1}(R_k^T L_k)$
return $L_k, D, R_k$
```

This concludes the algorithm to solve the new low-rank DMD problem. The solution to this problem with this algorithm will hereafter be referred to as lrDMD.

3.3.3. Computational Performance

The advantage to using the form of the optimization problem in Eq. (3.6) instead of Eq. (3.2) is that we only need to solve for $\mathcal{O}(m)$ variables for the fully parameterized state transition matrix. For large m, even this can make the problem computationally intractable. An important observation that can help is that we only need to consider the solution of R such that $\mathrm{Im}(R) \subseteq \mathrm{Im}(X)$. Consider M to be the full ranked data matrix that contains all the data snapshots that form the data matrices X and Y and let there be p such snapshots. Consider the singular value decomposition of the data matrix $M \in \mathbb{R}^{m \times p}$

$$M = U \Sigma V^T.$$

We assume that the basis of both L and R is contained within U so that

$$L = U\underline{L}$$
$$R = U\underline{R}$$

where $\underline{L}, \underline{R} \in \mathcal{G}_{r,p}$. The optimization problem reduces to

$$\begin{aligned} \min_{\underline{L},\underline{R}} \; & G(\underline{L}, \underline{R}) := (-\left\| \underline{L}^T \underline{Y} Q_{\underline{R}} \right\|_F^2) \\ \text{s.t. } & (\underline{L}, \underline{R}) \in \mathcal{M} := \mathcal{G}_{r,p} \times \mathcal{G}_{r,p} \\ & Q_{\underline{R}} = \underline{X}^T \underline{R} (\underline{R}^T \underline{X}\underline{X}^T \underline{R})^{-1} \underline{R}^T \underline{X} \end{aligned}$$

where $\underline{Y} = U^T Y$ and $\underline{X} = U^T X$. Since $p \ll m$, this saves significant computational time. In this study, we use this technique for lrDMD and OMD where gradient based methods are used to solve the optimization problem. Therefore, in this study, OMD and the gradient based implementation of lrDMD solve the optimization problem with the reduced data matrices $\underline{Y}$ and $\underline{X}$.

Table 3.1 compares the computational performance of DMD, OMD and the two algorithms to solve the lrDMD problem. It shows the time taken in seconds by the three methods in computing a reduced-order model for the given data matrices. The data snapshots have dimension $m = 62001$ and describe the vorticity field for a flow past flat plate described in Section 3.4.2 with consecutive

Method	$n = 50$			$n = 200$		
	$r = 10$	$r = 20$	$r = 30$	$r = 10$	$r = 20$	$r = 30$
DMD	0.1443	0.1317	0.1096	0.6965	0.6691	0.6464
OMD	3.2687	5.5517	10.5841	1.6759	17.4687	26.5706
lrDMD (subProj)	0.7731	0.4127	1.4995	2.3426	2.8213	1.4139
lrDMD (GradD)	19.9936	2.8781	0.8261	58.0509	17.1708	26.0866

Table 3.1.: Time taken in seconds by DMD, OMD and lrDMD for generating reduced-order models of rank r with data matrices composed of n snapshots. lrDMD (subProj) refers to subspace projection method implemented using Algorithm 1 and lrDMD (GradD) refers to gradient based method described in detail in [129].

snapshots that are 100 timesteps apart. The data matrices are generated with $n = \{50, 200\}$ snapshots and reduced-order models of rank $r = \{10, 20, 30\}$ are computed. lrDMD (subProj) refers to the solution of lrDMD using the subspace projection method described in Algorithm 1 whereas lrDMD (GradD) refers to the gradient based method [129]. The gradient based methods used in OMD and lrDMD are implemented using ManOpt [24] package on MATLAB using the trust-region [1] algorithm. The initial condition for OMD is given by the DMD solution whereas the initial condition for the lrDMD is the solution from Algorithm 1. All computations are performed on a standard desktop PC with a 2.2 GHz quad-core Intel i7 processor and 16GB RAM running on Mac OS X 10.11.

Table 3.1 shows that time taken by DMD is always lower than OMD or either method of solving the lrDMD problem, with the subspace projection method incurring computational times $2 - 3\times$ larger than DMD but is faster than OMD for all cases. An interesting observation is that for certain cases like $(n, r) = (50, 30)$, the initial condition provided to 'lrDMD (GradD)' by the subspace projection method is very close to the optimal and no iterations of the gradient based method are required for convergence. In almost all cases lrDMD with gradient descent takes around the same time as OMD except a few cases when OMD takes less time than lrDMD.

Table 3.2 shows the error norm defined by $\epsilon = \left\|Y - \hat{A}X\right\|_F$ for all the cases considered in Table 3.1. It can be seen that lrDMD with gradient descent and subspace projection shows error norm much less than OMD or DMD for all cases considered. Subspace projection in particular, shows lesser error than OMD in almost all cases while taking lesser time to compute the reduced-order model. For data snapshots that are spaced far apart, lrDMD with gradient descent is recommended for accurate low rank reconstruction of the system dynamics if computation time

Method	$n = 50$			$n = 200$		
	$r = 10$	$r = 20$	$r = 30$	$r = 10$	$r = 20$	$r = 30$
DMD	14.7449	5.7669	3.3224	24.0859	2.2424	0.3296
OMD	4.2816	0.0921	0.0049	12.1151	0.6408	0.0176
lrDMD (subProj)	1.8183	0.0160	0.0015	6.3463	0.5206	0.0295
lrDMD (GradD)	1.2686	0.0031	0.0015	5.0654	0.1638	0.0058

Table 3.2.: Error norm ϵ of reconstruction for reduced-order models of rank r generated by DMD, OMD and lrDMD with data matrices composed of n snapshots. lrDMD (subProj) refers to subspace projection method implemented using Algorithm 1 and lrDMD (GradD) refers to gradient based method described in detail in [129].

is not a factor.

3.4. Results

In this section we compare the performance of the reduced-order LQR controllers built using DMD, OMD and lrDMD on unsteady dynamical systems. The first system we look into is the complex linear Ginzburg-Landau equation. We use the reduced-order models to find optimal feedback control as well as the optimal actuator location for the system in the unstable regime. We also employ LQR controllers on the incompressible flow past inclined flat plate at high angle of attack. The results show that the reduced-order linear feedback controllers can be effective in regions of phase-space with strong nonlinearities and can be used to suppress nonlinear vortex shedding.

3.4.1. Linearized Ginzburg Landau Equation

In this section we apply the control strategies described so far on the linearized complex Ginzburg-Landau (GL) equation, which is a well known model equation for fluid systems. Even though all the analysis in this study has been on the real number field, the results can be easily extended complex numbers. The GL equation is as follows,

$$\frac{\partial q}{\partial t} + \nu \frac{\partial q}{\partial x} = \mu(x) q + \gamma \frac{\partial^2 q}{\partial x^2},$$
$$\text{with} \quad \mu(x) = \mu_0 - c_u^2 + \mu_2 x^2/2,$$

where the real part of $q(x,t)$ represents velocity or stream function perturbation amplitude. The GL equation exhibits a variety of stability behaviors observed in fluid flows for different values of the constants ν, c_u, μ_0 and μ_2 in different regions of the spatial domain. For this study we have taken the value of these constants corresponding to the supercritical, globally unstable regime case in [33], shown in Table 3.3.

variable	description	value
U	advection velocity	2
c_u	most unstable wavenumber	0.2
c_d	dispersion parameter	-1.0
μ_0	overall amplification	0.41
μ_2	degree of non-parallelism	-0.01
μ_t	transitional μ	0.32
μ_c	critical μ	0.4

Table 3.3.: Ginzburg-Landau equation parameter values corresponding to the supercritical, globally unstable regime case in [33].

In discrete space we use the spectral formulation of the derivative operators using Hermite polynomials evaluated at the Hermite nodes [13]. Time discretization is performed using forward Euler time-stepping scheme. In this study we consider $n = 220$ grid points with the computational domain $x \in [-85, 85]$ and a time-step of $dt = 1$. For more details about the system the reader is referred to [13].

We will describe this system as a discrete input-output system as,

$$\mathbf{q}_{k+1} = A\mathbf{q}_k + B\mathbf{u}_k, \tag{3.12}$$

where $\mathbf{q}_k \in \mathbb{R}^m$ is the state variable, $A \in \mathbb{R}^{m\times m}$ is the state transition matrix, $B \in \mathbb{R}^{m\times p}$ is the spatial support of the controller and $\mathbf{u}_k \in \mathbb{R}^p$ is the control input. Our goal is to find $K \in \mathbb{R}^{p\times m}$

such that when $u_k = -Kq_k$, we minimize the following cost,

$$J = \sum_{k=1}^{\infty} (\mathbf{q}_k^* Q \mathbf{q}_k + \mathbf{u}_k^* S \mathbf{u}_k), \tag{3.13}$$

for given positive definite matrices Q and S. We denote the conjugate transpose of $\mathbf{q}$ by $\mathbf{q}^*$. We choose the controller matrix B and the values of the weight matrix Q and S that are consistent with the previous study of Chen *et al.* [33]. The controller we choose is a Gaussian centered in the convectively unstable region of flow,

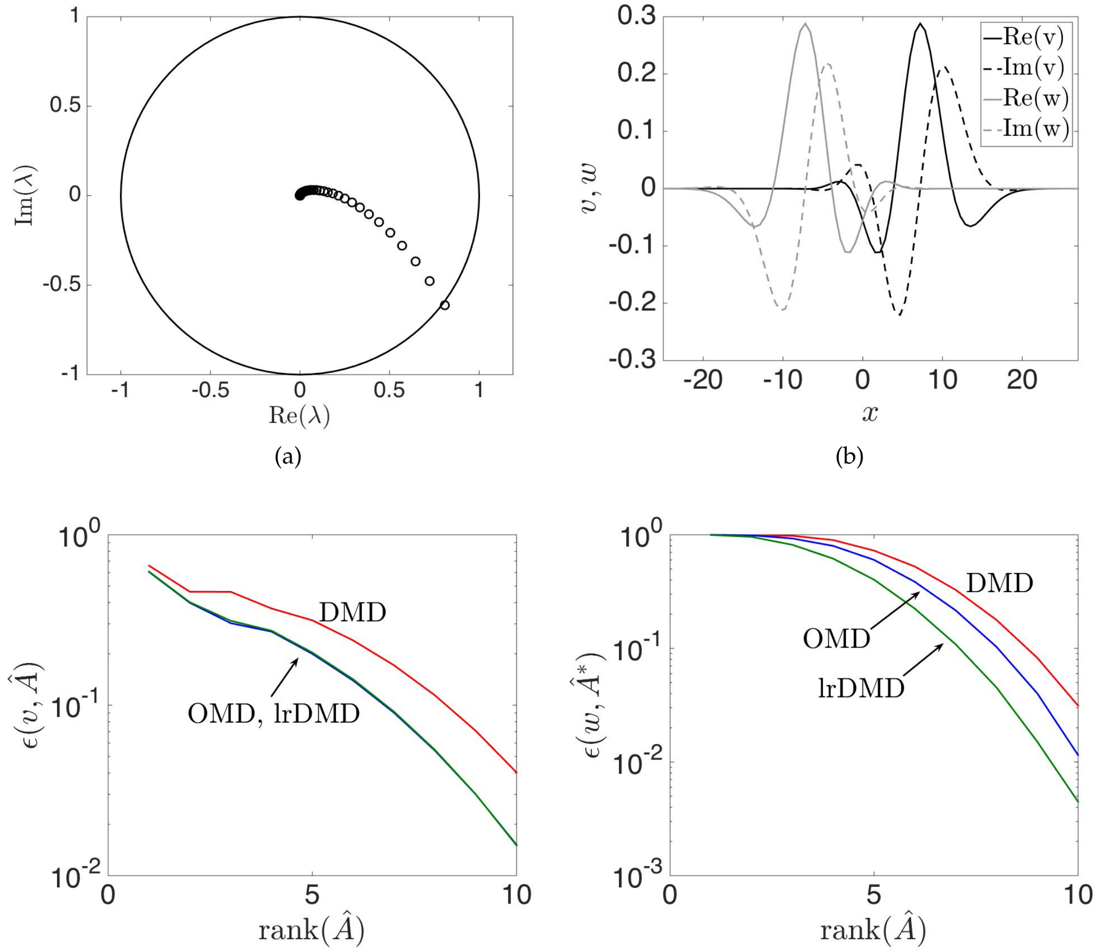
$$B = \exp\left(-\frac{(x - x_a)^2}{2\sigma^2}\right), \tag{3.14}$$

where we choose $x_a = 8$ and $\sigma = 5$ which places the actuator just inside the region of amplification which is $[-8.6, 8.6]$. We use $Q = \beta^2 M$ and $S = I$ where M is the weighting matrix corresponding to the trapezoidal integration operator for our discretized domain. The value of $\beta = 7$ determines the relative reward between minimizing the state perturbation amplitude and applying small input size. To capture the transience in the system, we use 15 snapshots with initial condition with the same profile as in Eq. (3.14) centered at $x_0 = -20$. We generate a low-rank approximation of A which is used to construct reduced-order controllers using the LQR framework and to find the optimal actuator location.

Adjoint Reconstruction

It is known that state transition matrices arising from the discretization of the linearized governing equations for various fluid flow applications are *nonnormal* [134]. Nonnormal systems exhibit non-orthogonal eigenmodes that can differ significantly from the adjoint modes [133] that are known to play a major role in flow control and optimization [98]. Therefore, it is crucial for effective flow control applications that the reduced-order models not only accurately predict the flow field but also extract adjoint information from the data.

Figure 3.1a shows the eigenvalues of the discretized state transition matrix of the GL equation in the supercritical regime. It shows one unstable eigenvalue $0.8073 - 0.6109i$ outside the unit circle. Figure 3.1b shows the eigenmode v and the adjoint mode w corresponding to this unstable



(a)

(b)

(c)

(d)

Figure 3.1.: lrDMD identifies low-order subspaces of containing leading eigenmodes and adjoint modes from data. (a) Eigenvalues of the discrete GL system matrix (black circles). Unit circle is shown with a solid black line for comparison. (b) Eigenmode 'v' and adjoint mode 'w' corresponding to the unstable eigenvalue outside the unit circle. (c) Projection error for the unstable eigenmode for DMD (red), OMD (blue) and lrDMD (green) for different rank approximations. (d) Projection error for the unstable adjoint mode for DMD (red), OMD (blue) and lrDMD (green) for different rank approximations.

eigenvalue. For faithful reconstruction of these unstable modes, the eigenmode must lie in the column space of the reduced-order model $\hat{A}$ and the adjoint mode must lie in the row space. We define the projection error

$$\epsilon(\mathbf{q}, V) = \frac{\|\mathbf{q} - \mathbb{P}_V \mathbf{q}\|_2}{\|\mathbf{q}\|_2} \tag{3.15}$$

where $\mathbb{P}_V$ is the orthogonal projection matrix in the column space of V. We look at the projection error of the unstable eigenmode in the column space of $\hat{A}$ given by $\epsilon(v, \hat{A})$ and the projection error of the unstable adjoint mode in the row space of $\hat{A}$ given by $\epsilon(w, \hat{A}^*)$. For the DMD-based reduced-order model, bases for the row space and column space are both given by the POD modes. For OMD, the solution L of the optimization problem in Eq. (3.3) serves as the basis of both the column and the row space of the reduced-order model. In the case of lrDMD, the basis of the row space is given by R and the basis of the column space is given by L from the solution of the optimization problem in Eq. (3.8).

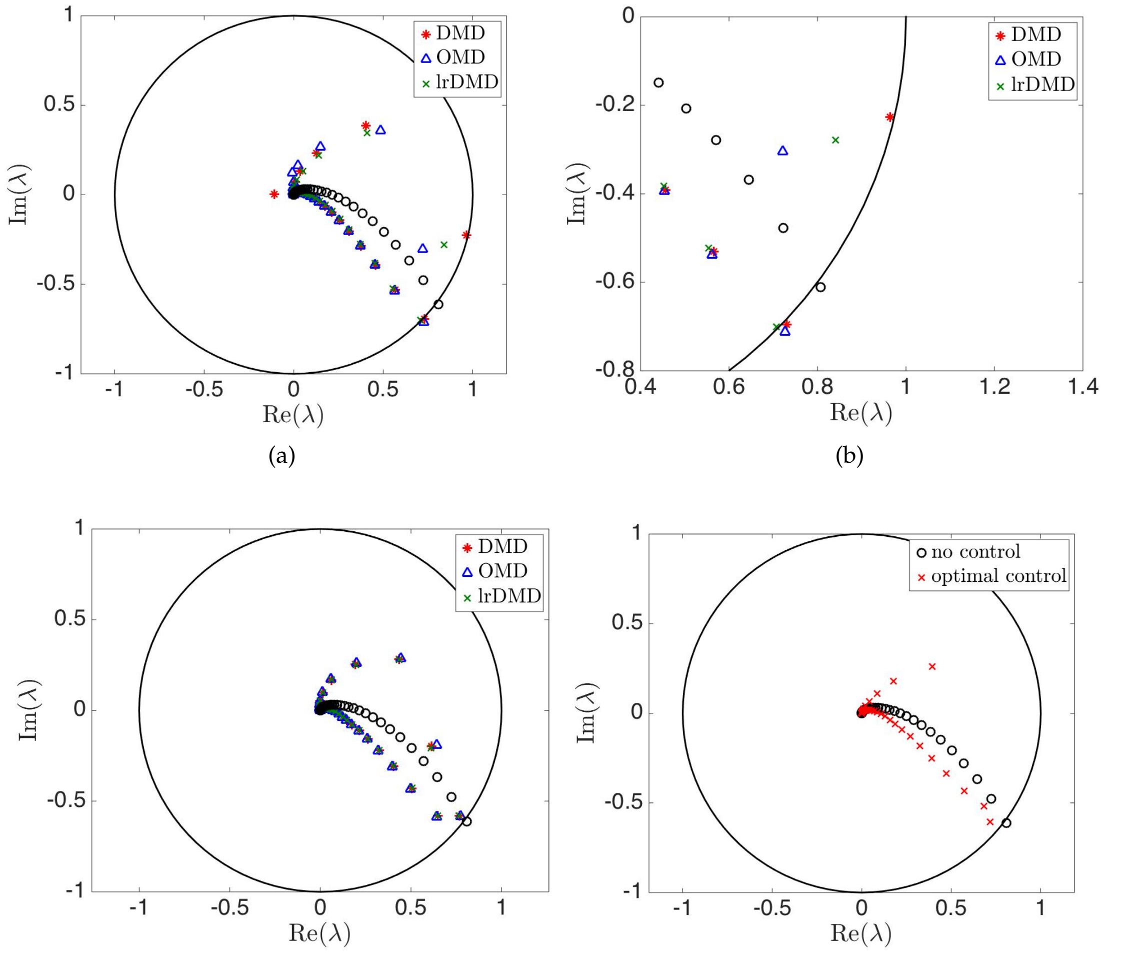
Figure 3.1c shows the projection error of the unstable eigenmode for the reduced-order models of different ranks obtained from DMD, OMD and lrDMD. At all rank approximations the error incurred by OMD and lrDMD are about the same and less than the error incurred by DMD. Similarly, Figure 3.1d shows the projection error of the unstable adjoint mode. In this case we see that lrDMD outperforms both OMD and DMD at all ranks. This shows that the additional degree of freedom of choosing separate input and output spaces in lrDMD allows for lower projection error in the adjoint modes of interest while also keeping the projection error in the eigenmodes as low as in OMD.

Here we have compared the subspace identification of lrDMD, OMD and DMD algorithms by evaluating the projection error of the unstable eigenmode and adjoint mode of the Ginzburg-Landau system in the supercritical regime. We extend this comparison to Eigensystem Realization Algorithm (ERA) in Appendix A.1.

Optimal Control

In the supercritical regime, the GL system is globally unstable, with a single eigenvalue of the state transition matrix that lies outside the unit circle, as shown in Figure 3.1a. The goal of the feedback controllers is to stabilize the system while minimizing the cost in Eq. (3.13).

Figure 3.2 shows the eigenvalues of the controlled Ginzburg-Landau system using reduced-order LQR controllers constructed using DMD, OMD and lrDMD. Figures 3.2a and 3.2b illustrate the rank-5 approximations whereas Fig. 3.2c shows the rank-9 approximations. Figure 3.2d shows the eigenvalues of the controlled GL system with LQR optimal control constructed using the full-order system matrix. It can be clearly seen in Fig. 3.2b that for the rank-5 approximation, DMD is not able to stabilize the system and has one eigenvalue outside the unit circle. OMD also fails



(a)

(b)

(c)

(d)

Figure 3.2.: lrDMD enables construction of effective low-order feedback controllers. The eigenvalues of the uncontrolled (○) and controlled GL system with reduced-order controllers constructed using DMD, OMD and lrDMD with (a,b) rank-5 and (c) rank-5 approximations. (d) Shows the eigenvalues of uncontrolled and optimally controlled GL system.

to stabilize the system with one eigenvalue outside the unit circle. This shows that 5 modes are not sufficient for reduced-order control even with OMD which finds the optimal basis for the low order subspace. However, the freedom of choosing separate input and output subspaces enables lrDMD based reduced-order controller to stabilize the system at the same rank 5 approximation. For the rank 9 approximation in Fig. 3.2c, we see that the DMD, OMD and lrDMD controlled system eigenvalues are very close to each other and also close to the eigenvalues of the optimally controlled system. Thus with sufficient rank approximation all three methods can stabilize the system but lrDMD can stabilize the system at a rank lower than OMD or DMD.

Optimal Actuator Placement

Optimal actuator placement is a challenging problem in stability theory especially when applied to large scale fluid systems. Reduced-order models make the investigation of optimal actuator location computationally feasible. We use the reduced-order feedback controllers to find the optimal actuator location x_a (in Eq. (3.14)) in the computational domain for a fixed variance in the spatial support of the actuator. The authors in [33] find the optimal actuator and sensor location that minimizes the H_2 norm of the controlled system [111]. We find the optimal actuator location that minimizes the supremum of the cost of the controlled system described in Eq. (3.13) for all possible initial conditions. We employ brute force sampling of actuator locations while using the reduced-order model approximations.

Let the control input for the GL system (Eq. (3.12)) be given by the feedback law $\mathbf{u}_k = -K\mathbf{q}_k$. The controlled dynamics and the cost function can be re-written as

$$\begin{aligned} J &= \sum_{k=1}^{\infty} \mathbf{q}_k^*(Q + K^*SK)\mathbf{q}_k \\ \text{s.t.} \quad \mathbf{q}_{k+1} &= (A - BK)\mathbf{q}_k \end{aligned} \tag{3.16}$$

for some given initial condition $\mathbf{q}_0$. Let $\underline{Q} = (Q + K^*SK)$ and $\underline{A} = (A - BK)$. We are interested in the following infinite summation,

$$\begin{aligned} J &= \sum_{k=1}^{\infty} \mathbf{q}_0^*(\underline{A}^*)^k \underline{Q}\,\underline{A}^k \mathbf{q}_0 \\ &= \mathbf{q}_0^* F \mathbf{q}_0. \end{aligned}$$

where $F = \sum_{k=1}^{\infty}(\underline{A}^*)^k \underline{Q}\,\underline{A}^k$. Notice that for a given dynamical system, the cost solely depends on the initial condition. When the controlled system governed by $\underline{A}$ is stable, i.e. all the eigenvalues of $\underline{A}$ lie within the unit circle, F can be efficiently computed by solving the following discrete Lyapunov equation

$$\underline{A}^* F \underline{A} - F + \underline{Q} = 0.$$

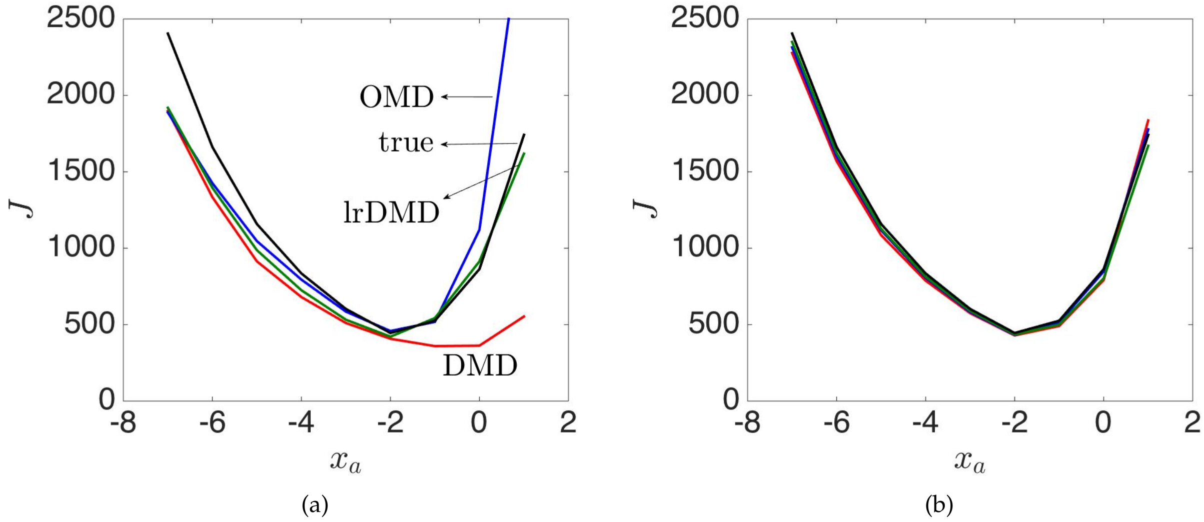


Figure 3.3.: lrDMD provides accurate estimates for optimal actuator placement. Supremum of the cost J over all initial conditions for the controlled GL system with DMD (red), OMD (blue) and lrDMD (green) reduced-order controllers for (a) rank-5 approximation and (b) rank-9 approximation.

An alternative formulation to handle the case of unstable dynamical systems, i.e. when not all eigenvalues of $\underline{A}$ lie within the unit circle, is shown in the paper by Bhattacharjee *et al.*(Section II.B. [20]).

The maximum value of the cost function is given by the largest eigenvalue of F denoted by $\lambda_{\max}(F)$. The initial condition that yields this cost is the eigenvector of X that corresponds to the largest eigenvalue. We use the reduced-order approximation of the state transition matrix A to construct low-order feedback gain K and compute the values of $\lambda_{\max}(F)$. We then conduct an exhaustive search over all possible actuator location in an interval to find the optimal actuator location. The performance of the reduced-order method is measured by how close this location is to the true optimal actuator location.

Figure 3.3 shows the supremum of the cost over all possible initial conditions for different actuator locations $x_a \in [-7, 1]$ using DMD, OMD and lrDMD approximation compared with the true full-order system. The optimal actuator location is given by the minima of this plot. Figure 3.3a shows the results for a rank-5 approximation. lrDMD outperforms both DMD and OMD in matching the cost over the range of actuator locations considered. The DMD approximation attains its minima at $x_a = 0$ whereas the true optimal actuator location is $x_a = -2$. Optimal actuator locations given by OMD and lrDMD agree with the true optimal actuator location. Figure 3.3b shows the same results for a rank-5 approximation using DMD, OMD and lrDMD. At this rank all three methods perform equivalently and provide accurate costs for all actuator locations considered.

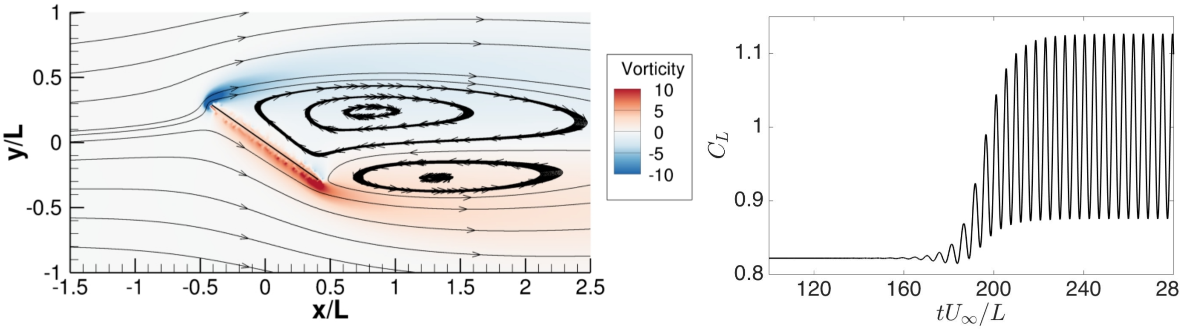
3.4.2. Full-state Feedback Control of Flow over Inclined Flat Plate

In this section we demonstrate the performance of the controllers on stabilizing the two-dimensional uniform flow approaching an inclined flat plate. The freestream flow is at a low Reynolds number of 100 and the flat plate is inclined at an angle of 35°. At these conditions, it has been shown [4] that the steady state of the flow is unstable and the flow exhibits periodic vortex shedding. The goal of the reduced-order controllers is to bring the system back to steady state from different initial conditions in the transition process.

The flow is simulated using the fast immersed boundary method developed in [35]. It is an efficient method to solve incompressible Navier-Stokes equations based on an immersed boundary formulation. In order to achieve uniform flow conditions in the far field, a multi-domain approach is employed. The domain of interest is considered to be embedded in a series of domains, each twice-as-large as the preceding but with the same number of the uniform grid points. The numerical parameters of the flow are taken to follow the work of [4]. The grid size used is 250×250 and the domain of interest is given by $[-2, 3] \times [-2.5, 2.5]$ where the lengths are non-dimensionalized by the chord length of the flat plate, L. The center of the flat plate is located at the origin. Five domains, each with the same number of grid points are used for an effective computational domain that is 2^4 times larger than the domain of interest. The time-step is taken as $dt = 0.01L/U_\infty$ where U_∞ is the freestream velocity.

Steady State

The computation of the steady state of an unstable flow is generally more difficult than a stable flow configuration. We are interested in the initial condition that is a fixed point of the governing equations of the flow [3]. Using a computational Fortran wrapper around the immersed boundary method code, we perform Newton-GMRES iterations on the nonlinear solver. We find



(a)

(b)

Figure 3.4.: Unstable configuration of flow over an inclined flat plate. (a) Vorticity contours and velocity streamlines for steady flow with $Re = 100$ over 35° inclined flat plate (b) C_L vs. time for 35° inclined flat plate with the unstable steady state as the initial condition

$x_{c,1}$	$y_{c,1}$	$x_{c,2}$	$y_{c,2}$	a	c
0	1.3423	0	0.89	20	2

Table 3.4.: Numerical parameters for the actuator location and strength for feedback control of flow past inclined flat plate.

the circulation field γ that satisfies

$$g(\gamma) = \gamma - \psi_T(\gamma)$$

where $\psi_T(\cdot)$ is the nonlinear solver that advances the solution γ_k at timestep k to $\gamma_{k+1} = \psi_T(\gamma_k)$ after T timesteps. For the purpose of this study we choose $T = 50$ and iterate until a convergence tolerance of 10^{-8} in the L_2 norm. Figure 3.4a shows the vorticity contours and velocity streamlines of the steady state while Figure 3.4b shows the coefficient of lift as instabilities grow with the unstable steady state as the initial condition.

Snapshots

The snapshots for the study are generated by the nonlinear impulse response of the flow to an actuator [4]. We consider the evolution of the vorticity field that can be described as

$$\mathbf{q}_{k+1} = \phi_T(\mathbf{q}_k) + B\mathbf{u}_k, \tag{3.17}$$

where $\mathbf{q}_k$ is the vorticity field at iteration k, ϕ_T is the nonlinear solver that advances the vorticity field by T timesteps, B is the actuator and $\mathbf{u}_k$ is the control input at iteration k. The actuator is a simple model of localized body force [3] at the located near the leading edge of the flat plate (see Fig. 3.5). The instantaneous vorticity field generated by impulse control input of the actuator is

$$B(r) = c[(1 - ar_1^2)\exp(-ar_1^2) - (1 - ar_2^2)\exp(-ar_2^2)],$$

where $r_i^2 = (x - x_{c,i})^2 + (y - y_{c,i})^2$ for $i = 1, 2$. The constants a and c determine the shape and strength of the control, respectively. Table 3.4 shows values of the constants used for this study. Figure 3.5 shows the vorticity field generated by the actuator relative to the position of the flat plate in the computational domain.

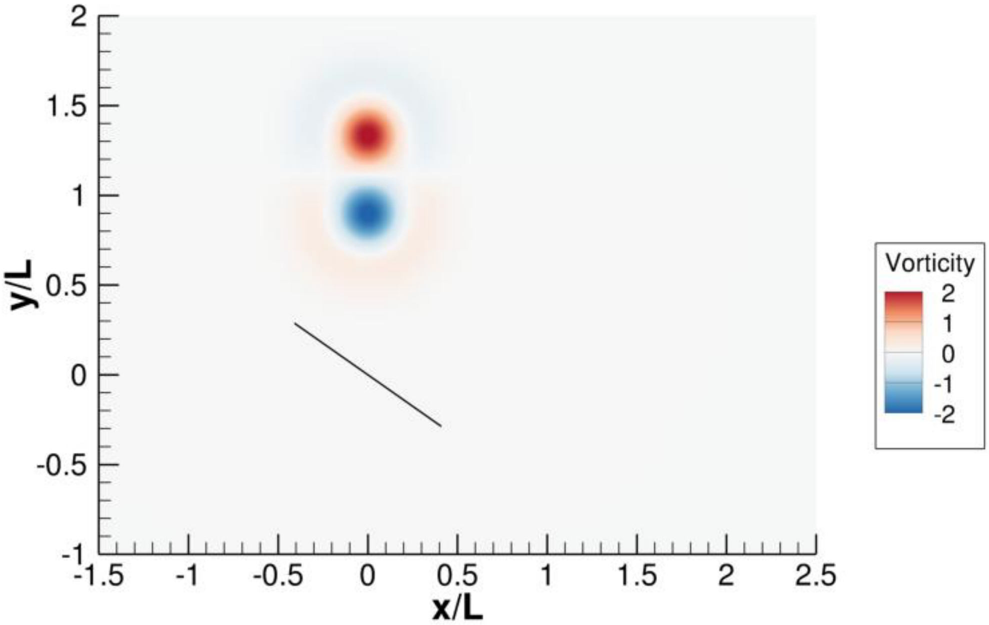


Figure 3.5.: Vorticity field generated by the actuator placed near the leading edge of a flat plate that is inclined at 35° with the freestream.

Getting a set of snapshots that captures the essential physics is crucial for the performance of data-driven methods. Therefore, an important step in building data-driven reduced-order controllers is to identify the nature of instabilities in the flow and select data that best captures the behaviour of interest. The goal is to select a set of snapshots that capture the near-linear behaviour of the flow before the nonlinearities dominate the system dynamics. Figure 3.6 shows the evolution with time of drag coefficient of the impulse response of the flat plate. The two regions marked as exponential growth and limit cycle can be approximated by linear dynamics. We are only interested in the exponential growth region and we choose to sample this region for our study.

Feedback Control

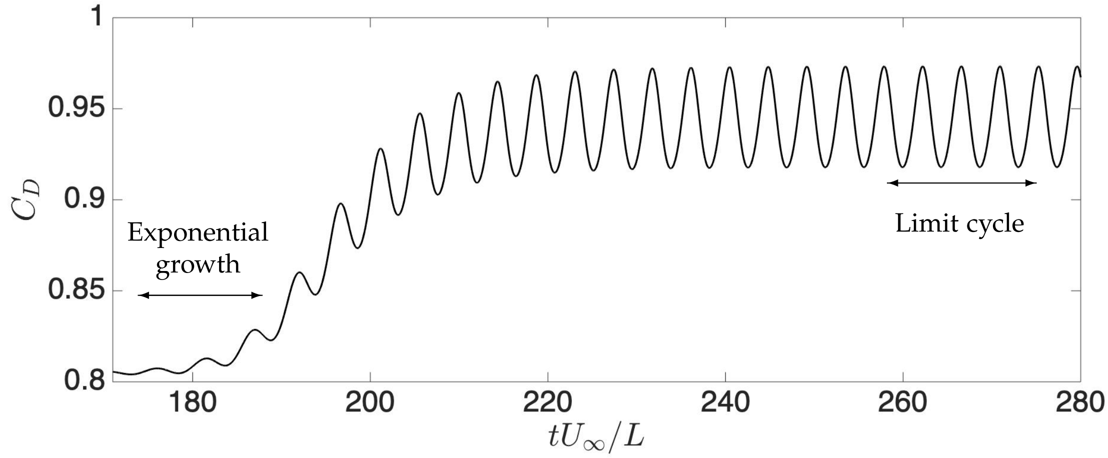


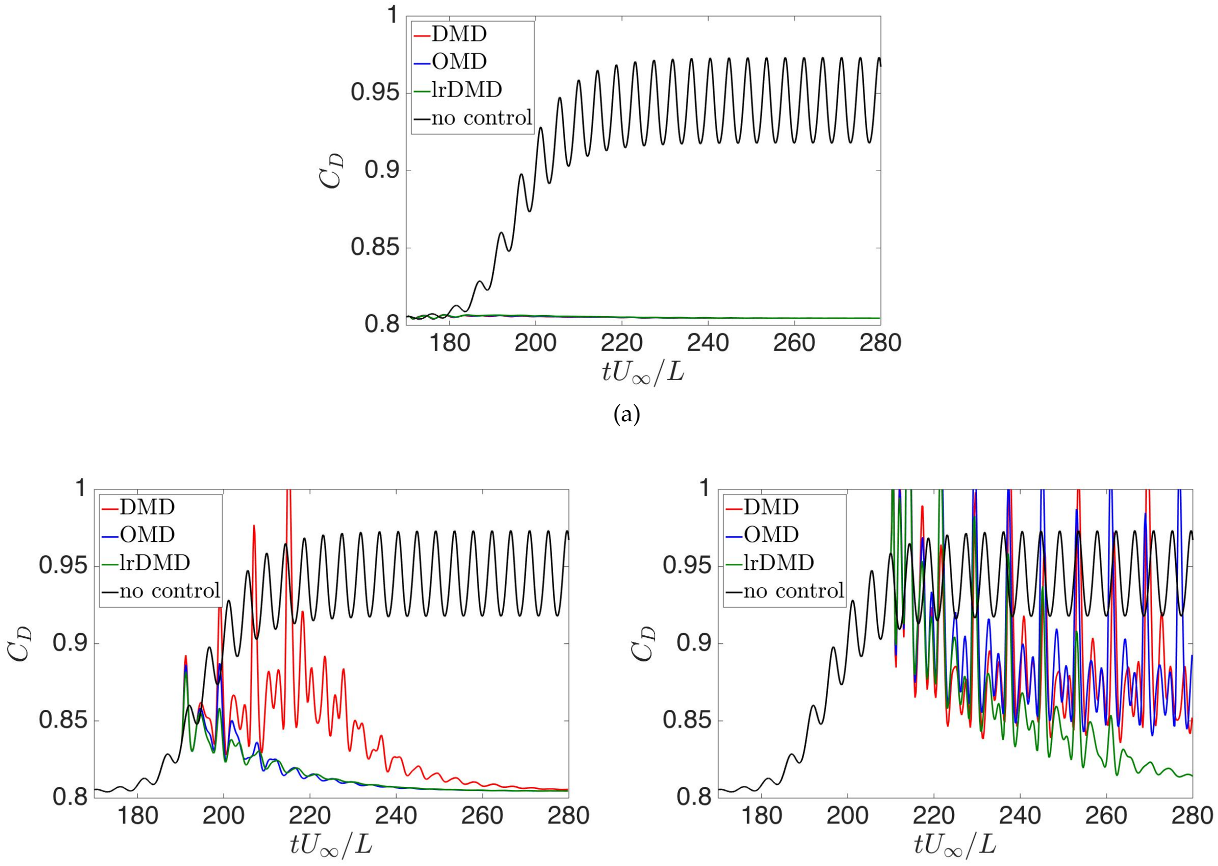
Figure 3.6.: C_D vs. time plot of the impulse response of flow over an 35° inclined flat plate with steady state as the initial condition. Snapshots of this simulation were used to learn the the reduced-order models in the study.

We model the system as linear about the steady state of the flow. To this end, we subtract the computed steady state from the flow snapshots and build a reduced-order model for the perturbations about the steady state. The reduced-order model can be described as

$$\widetilde{\mathbf{q}}_{k+1} = A\widetilde{\mathbf{q}}_k + B\mathbf{u}_k$$

where $\widetilde{\mathbf{q}}_k = \mathbf{q}_k - \overline{\mathbf{q}}$ and $\overline{\mathbf{q}}$ is the steady state vorticity field. This can be considered as a model of the dynamics of perturbations in the flow around the steady state.

The control is performed using a Fortran wrapper around the nonlinear solver. The controller is activated at three different points in the transition process, $t_0 = \{170, 190, 210\}$ (refer to Fig. 3.4b), in separate simulations, to show the effect of the nonlinear dynamics of the flow. The nonlinear effects grow larger as the flow deviates from the steady state with time. At $t_0 = 170$, the flow shows exponential growth in perturbation which is predominantly linear behavior. At $t_0 = 190$, the flow has entered an algebraic growth in perturbation magnitude and $t_0 = 210$ is just before the flow enters a limit cycle behavior. At $t = \{190, 210\}$ the perturbations are too large for the linear assumption to be valid.



(a)

(b)

(c)

Figure 3.7.: lrDMD enables suppression of vortex shedding at multiple operating conditions. C_D vs. time for 35° inclined flat plate with the actuator activated at (a) $t_0 = 170$ (b) $t_0 = 190$ and (c) $t_0 = 210$ using DMD, OMD and lrDMD based reduced-order controllers.

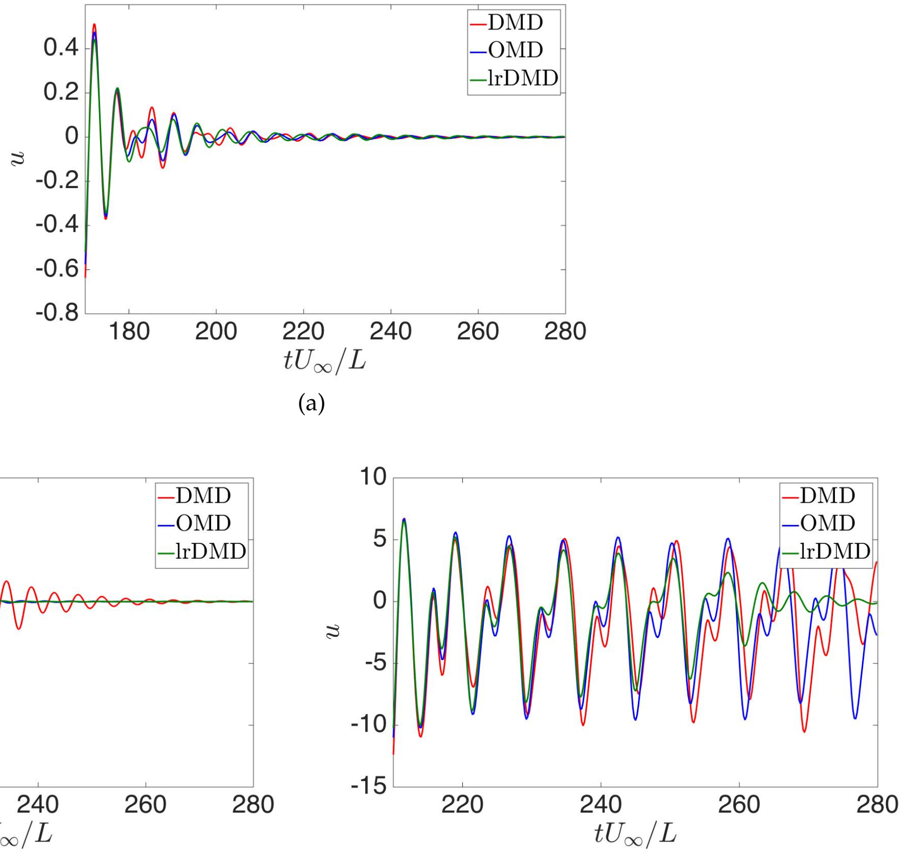
Equation (3.17) is used to generate snapshots to build the data matrix. We choose $T = 20$ and collect 200 snapshots to cover multiple period of oscillations of the exponentially growing perturbations. Linear approximations of rank 20 using DMD, OMD and lrDMD are used to generate the LQR controllers. Since $T = 20$, the feedback control is applied after every 20 timesteps of the nonlinear flow solver. A rank of 20 is high enough to faithfully reconstruct the dynamics but much lower than the dimension of the full order system which is of $O(10^4)$. Figure 3.7 shows the evolution of C_D as the controller is switched on at the three time points. Figure 3.7a shows DMD, OMD and lrDMD are all able to prevent the growth of perturbations in the flow when the controller is activated at $t_0 = 170$. This is expected since the perturbations are small and the flow is in the linear regime. In Fig. 3.7b at $t_0 = 190$, OMD and lrDMD show very similar control performance and are able to suppress the vortex shedding earlier than DMD. Finally in Fig. 3.7c, lrDMD clearly outperforms both OMD and DMD in bringing the flow back to the steady state. This shows that in the linear regime all three methods are equivalent in performance. As the perturbations grow and nonlinear effects in the flow become significant, the additional flexibility of lrDMD that allows a different input and output subspaces results in a better control performance compared to OMD and DMD.

Figure 3.8 shows the control input with time for the three activation times of the controllers built using DMD, OMD and lrDMD whose performance was shown in Fig. 3.7. The results follow the same trend as Fig. 3.7. Specifically, in Fig. 3.8a, we see that when the controller is activated at $t_0 = 170$, all three controllers show similar performance. However, at $t_0 = 190$ in Fig. 3.8b, OMD and lrDMD incur much lower cost compared to DMD. Finally, in Fig. 3.8c, the lrDMD controller is the only controller able to stabilize the flow in the given time horizon and also incurs lower cost compared to the DMD and OMD controllers.

Method	$t_0 = 170$		$t_0 = 190$		$t_0 = 210$	
	$T = 50$	$T = 100$	$T = 50$	$T = 100$	$T = 50$	$T = 100$
DMD	✓	✓	✓	✗	✗	✗
OMD	✓	✓	✓	✓	✓	✗
lrDMD	✓	✓	✓	✓	✓	✗

Table 3.5.: Summary of the feedback control performance of DMD, OMD and lrDMD based controllers for flow over inclined flat plate. The time at which the controller was activated is indicated by t_0 and $T = \{50, 100\}$ refers to the time separation of corresponding data snapshots in the data matrix used to construct the controllers. ✓ indicates that the flow converged to steady state and ✗ indicates that the flow became unstable.

We repeat this numerical experiment using data matrices with consecutive snapshots that are 50 and 100 iterations apart. Table 3.5 shows the results of feedback based control at different initial conditions of the flow for DMD, OMD and lrDMD based controllers. As the separation between the flow snapshots increases, the actuation is being applied further apart in time which makes it harder to suppress the vortex shedding process. OMD and lrDMD controllers are successful in controlling the flow at all conditions except when the snapshots are 100 timesteps apart and the



(a)

(b)

(c)

Figure 3.8.: lrDMD enables efficient feedback control of flow over an inclined flat plate. Strength of the actuation when the controller was switched on at (a) $t_0 = 170$ (b) $t_0 = 190$ and (c) $t_0 = 210$ for a 35° inclined flat plate using DMD, OMD and lrDMD based reduced-order controllers.

initial condition is at $t_0 = 210$ at which the flow becomes unstable. DMD however fails more often especially when the snapshots are 100 timesteps apart, it is only able to control when $t_0 = 170$.

3.5. Conclusion

To summarize, we introduce a method to approximate the dynamics of an unsteady fluid flow by a rank-constrained linear representation. We solve for the optimal linear map of a fixed user defined rank that reconstructs the data snapshots by minimizing the L_2 norm of the reconstruction error. Two methods of solving the optimization problem are presented and their limitations are discussed. The first method is a subspace projection method that is very fast and provides a good approximation to the optimal solution of the optimization problem. The second method employs a gradient descent approach which is guaranteed to converge at a local minimum. Reduced-order models generated using lrDMD are shown to produce lower reconstruction errors compared to Optimal Mode Decomposition (OMD) and Dynamic Mode Decomposition (DMD) while incurring comparable computational times. These reduced-order models are used to construct low-rank full-state feedback controllers to control model fluid flows.

We employ LQR based feedback control using the reduced-order models on the linearized Ginzburg-Landau equation in the globally unstable regime. lrDMD is able to stabilize the system at a much lower rank approximation as compared to DMD. The reduced-order models are also used to find optimal actuator location using brute force sampling approach. The true optimal actuator location is obtained using lrDMD at a rank-5 approximation, whereas DMD requires a rank-5 approximation to provide the true optimal actuator location. We also employed LQR based feedback control on unsteady flow over an inclined flat plate. OMD and lrDMD perform equally well in controlling the flow whereas DMD incurs much higher costs in the control for all cases.

There are a number of future directions to extend this research. The current method finds a low-rank description of an autonomous, possibly nonlinear, dynamical system. One possible direction is to extend the current method to incorporate the effect of control input similarly to DMDc (Dynamic Mode Decomposition with control) [119]. Section 6.1.1 formulates this extension and discusses possible methods to make this extension. Another extension is to consider partially observed system with low-order hidden state dynamics. There is also the possibility of allowing arbitrary sampling times for the snapshots as is often the case in experimental data. An alternate approach of reduced-order modeling when using DMD and ERA based methods is to first generate models with higher rank and then perform balanced truncation [53] to obtain the final model of desired rank. This approach is called *overspecification* and has been shown in perform better than just DMD or ERA (Section 3.5 of [125]). It would be interesting to see if overspecification applied to lrDMD and OMD would result in similar improvements as seen in DMD and ERA. In conclusion, this study shows the potential of using a linear map that is a mapping between different subspaces to construct reduced-order controllers for unsteady high-dimensional systems. We observe that these linear maps fit the data better and can lead to better performing reduced-order controllers.

CHAPTER 4

DATA-DRIVEN SENSOR PLACEMENT

This chapter describes a data-driven sensor placement method that leverages recent advancements in Koopman-based data-driven reduced-order modeling methods. Section 4.1 describes the motivation for the proposed method and Section 4.2 outlines the data-driven sensor placement problem. In Section 4.3, we pose the sensor placement optimization problem and derive the adjoint equations that can efficiently compute the gradient of the objective function with respect to the sensor locations. The reconstruction, prediction and control application results are shown in Section 4.4, and Section 4.5 concludes the chapter.

4.1. Motivation

Sensor and actuator placement is critical for state estimation and control of fluid flows in various applications such as weather forecasting [29], hydraulic fluid machinery [54], rapid detection of infectious diseases [49], increasing lift of airfoils [128] and suppression of noise in high-speed jets [112]. Appropriately-placed sensors improve the performance of the observers, leading to better analysis and control decisions. The need to place limited numbers of sensors and actuators in a large space of possible locations makes exploring the solution space challenging. The problem of a large solution space is further aggravated by the multi-scale, and possibly multi-physics, nature of fluid dynamics. Upon discretization, these features lead to a high-dimensional dynamical system that makes exact methods for optimal sensor and actuator placement intractable [33]. Consequentially, several approximate and heuristic methods have been proposed that place sensors in the regions of high instability [6, 13] or large structural sensitivity [50, 111] characterized by regions of both high instability and sensitivity. Greedy algorithms that optimize scalar measures of observability and controllability have also been proposed [106]. However, these methods require complete knowledge of the flow dynamics that might not be available or achievable. An alternative is to use data collected from experiments or numerical simulations to guide the sensor placement.

Several data-driven methods for sensor placement make use of the fact that the dynamics of fluid flows can sometimes be approximated by a low-dimensional representation. Such methods include gappy Proper Orthogonal Decomposition (POD) [26, 150] that selects sensor locations that will enable the POD modal coefficients to be determined most accurately, and discrete empirical interpolation methods (DEIM) [32, 42] that provide sampling strategies for a near-optimal reconstruction of nonlinear terms in reduced-order models. Manohar *et al.* [105] introduced a sparse sensor placement method, SSPOR, that extended the EIM approach to allow for oversampling, where the number of sensors can exceed the number of dimensions of the low-dimensional representation. This allows greater reduction in reconstruction error compared to the DEIM approach. Additionally, several compressive sensing based methods [15, 43] have been proposed that allow signal reconstruction with sparse measurements. However, these data-driven methods do not model the dynamics of the fluid to aid in sensor and actuator placement. Data-driven reduced-order modeling methods that construct a linear approximation of the flow dynamics allow us to use tools from linear control theory for control of complex fluid flows. With this motivation, we focus on the problem of sensor placement using data-driven reduced-order modeling and a linear observer-based objective function.

We leverage recent developments in Koopman-based data-driven modeling of fluid flows (see Section 2.1.3) to solve the sensor placement problem. These modeling methods have enabled stability analysis [107, 142], estimation [139–141] and control [18, 39, 71, 89, 118] of fluid flows. Data-driven models such as Dynamic Mode Decomposition (DMD) [34, 132] and its variants [34, 37, 48, 57, 63, 64, 129, 154] provide a finite-dimensional approximation of the Koopman operator. These methods are closely related to more traditional system identification methods such as Observer/Kalman Filter Identification (OKID) [147], N4SID [145, 146] and Eigensystem Realization Algorithm (ERA) [79], which have been used for data-driven sensor and actuator selection in fluid flows [21]. For a given linear approximation of the flow dynamics, we use the Kalman filter to estimate the hidden state variable. In the absence of the full-order model, we use an empirical approximation of the estimation error covariance matrix to construct the objective function parameterized by the sensor parameters. The optimal sensor parameters are selected through gradient-based minimization of the objective function, where the gradient is computed using an adjoint framework. Since we use reduced-order modeling along with adjoint-based optimization, our method scales well with both the dimension of the full-order system and the number of sensors. The proposed method can also be targeted to serve specific applications using different objective functions. We demonstrate this by proposing different objective functions for prediction and control applications.

4.2. Preliminaries

Consider the following discrete time stochastic dynamical system,

$$\begin{aligned} \mathbf{q}_{k+1} &= f(\mathbf{q}_k) + h(\mathbf{u}_k) + \mathbf{w}_k, \\ \mathbf{z}_k &= g(\mathbf{q}_k) + \mathbf{v}_k, \end{aligned} \tag{4.1}$$

with time indexed by k and initial condition $\mathbf{q}_0$. The output of the system $\mathbf{z}_k \in \mathbb{C}^p$ is a function of the state variable $\mathbf{q}_k \in \mathbb{C}^m$ and the control input $\mathbf{u}_k \in \mathbb{C}^d$. In fluid experiments, the output is generally single or multiple local sensor measurements of the flow variables, such as temperature or pressure, with associated measurement noise. For many fluid flows, h and g can be represented as a linear functions of the control input $\mathbf{u}_k$ and the state variable $\mathbf{q}_k$, respectively. We represent g as an observation matrix $C_\Theta \in \mathbb{C}^{p\times m}$ parameterized by the sensor profile specifications Θ that include sensor location and shape parameters. Similarly, h is represented by a control matrix $B \in \mathbb{C}^{m\times d}$. The plant noise is denoted by $\mathbf{w}_k$ while the measurement noise is denoted by $\mathbf{v}_k$. We assume that $\mathbf{w}_k$ and $\mathbf{v}_k$ are zero mean Gaussian variables with covariance matrices $W \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{p\times p}$ respectively. If we assume that the state variable $\mathbf{q}$ lies in a low-dimensional linear subspace of dimension r, we get the following reduced-order model

$$\begin{aligned} \mathbf{a}_{k+1} &= L^H f(L\mathbf{a}_k) + L^H B\mathbf{u}_k + L^H \mathbf{w}_k, \\ \mathbf{z}_k &= C_\Theta L\mathbf{a}_k + \mathbf{v}_k, \end{aligned} \tag{4.2}$$

where $L \in \mathbb{C}^{m \times r}$ is an orthonormal basis of the r-dimensional subspace, $\mathbf{a} \in \mathbb{C}^r$ is the projection of the state variable $\mathbf{q}$ onto the space spanned by the columns of L and the initial condition is $\mathbf{a}_0 = L^H \mathbf{q}_0$. Note that $L^H \mathbf{w}_k$ is a Gaussian random variable with zero mean and covariance matrix given by $L^H W L$.

Computation of the projected dynamics $L^H f(L\mathbf{a}_k)$ requires complete knowledge of the full order dynamics described by $f(\cdot)$, which is not always available for complex fluid flows. Data-driven reduced-order modeling methods provide an alternative in such scenarios. Using $n+1$ full-state observations $\mathbf{q}^n := \{\mathbf{q}_0, \cdots, \mathbf{q}_n\}$ of the system for a series of control input $\mathbf{u}^{n-1} := \{\mathbf{u}_0, \cdots, \mathbf{u}_{n-1}\}$, these methods provide a linear approximation of the dynamics restricted to low-dimensional space such that $L^H f(L\mathbf{a}_k) \approx A\mathbf{a}_k$, where $A \in \mathbb{C}^{r \times r}$ (see Section 2.1.3). This results in the following representation of the flow dynamics.

$$\begin{aligned} \mathbf{a}_{k+1} &= A\mathbf{a}_k + L^H B \mathbf{u}_k + L^H \mathbf{w}_k, \\ \mathbf{z}_k &= C_\Theta L \mathbf{a}_k + \mathbf{v}_k. \end{aligned} \tag{4.3}$$

For a given set of full-state observations $\mathbf{q}^n$ and control inputs $\mathbf{u}^{n-1}$, a reduced-order model of the flow A and control matrix B, our goal is to find sensor parameters Θ (which determine the matrix C_Θ) that would provide the optimal filtered estimates for the full-state observations. Under the assumption of Gaussian zero-mean noise, we use the Kalman Filter equations [148] that provide closed-form expressions for the filtered estimate of the full-state variables $\mathbf{q}^n$ (see Section 2.2.1).

We construct an objective function that penalizes the error between the full-state observations $\mathbf{q}^n$ and the filtered estimates of the state variable. A gradient-based optimization framework is employed to find the optimal sensor parameters Θ. The gradient of the objective function with respect to the sensor parameters is computed using adjoint equations. In the next section, we formally pose the optimization problem and describe the details of the gradient-based optimization method.

4.3. Mathematical Formulation

We formulate the sensor placement problem as a constrained optimization problem. First, we describe the constraints of the optimization problem. We then propose two objective functions, for flow reconstruction/prediction applications and flow control applications, respectively. Finally, we derive the adjoint equations for the optimization problem that are used to efficiently compute the gradient of the objective function with respect to the sensor locations.

4.3.1. Constraints

For the reduced-order dynamical model of the fluid flow (Eq. (4.3)), let the estimate of $\mathbf{a}_k$ given observations $\mathbf{z}^\ell := \{\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_\ell\}$ and control inputs $\mathbf{u}^\ell := \{\mathbf{u}_0, \cdots, \mathbf{u}_\ell\}$ be denoted by $\mathbf{a}_{k|\ell}$. The estimate for the full-state variable $\mathbf{q}_k$ will be similarly denoted as $\mathbf{q}_{k|\ell} = L\mathbf{a}_{k|\ell}$. Using the steady state Kalman filter equations (see Section 2.2.1 for details), we get the following recursion for the estimate $\mathbf{a}_{k|k}$,

$$
\begin{aligned}
\mathbf{a}_{0|0} =& E[\mathbf{a}_0], \\
\mathbf{a}_{k+1|k} =& A\mathbf{a}_{k|k} + L^H B\mathbf{u}_k, \\
\mathbf{a}_{k+1|k+1} =& \mathbf{a}_{k+1|k} + K(\mathbf{z}_{k+1} - C_\Theta L\mathbf{a}_{k+1|k}), \\
\Sigma =& A\Sigma A^H + L^H W L - A\Sigma L^H C_\Theta^H S^{-1} C_\Theta L \Sigma A^H, \\
K =& \Sigma L^H C_\Theta^H S^{-1}, \\
S =& C_\Theta L \Sigma L^H C_\Theta^H + V,
\end{aligned}
$$

where at each iteration k, $\mathbf{a}_{k|k} \in \mathbb{C}^r$, $\mathbf{a}_{k+1|k} \in \mathbb{C}^r$ are the estimates, $\Sigma \in \mathbb{C}^{r\times r}$ and $S \in \mathbb{C}^{r\times r}$ are error covariance matrices and $K \in \mathbb{C}^{r\times p}$ is the Kalman filter gain matrix.

4.3.2. Objective Function

Reconstruction and Prediction Applications Here, we consider the setting in which we are interested in finding sensor locations such that the Kalman filter estimate of the state variable is close to the observed values. To that end, we define the objective function as the L_2 norm of the estimation error of the state variable,

$$J(\mathbf{q}^n, \Theta) = \frac{1}{2} \sum_{k=1}^{n} \left\| \mathbf{q}_k - L\mathbf{a}_{k|k} \right\|^2, \tag{4.4}$$

where Θ denotes the sensor parameters that determine the estimate $\mathbf{a}_{k|k}$ of the hidden state a_k given the observations $\{\mathbf{z}_0, \ldots, \mathbf{z}_k\}$. This objective function can be seen as a data-driven approximation of the trace of the estimation error covariance matrix averaged over time. This approach is closely related to A-optimal design [7, 76] that minimizes the trace of the inverse of the Fisher information matrix[1].

Control Applications For control applications, we are interested in getting the correct control input at each time iteration. Under the linear-quadratic control framework [12, 56, 67, 82, 83] (see Section 2.2.2), the observer-based control input $\mathbf{u}_k$ at iteration k is given by $\mathbf{u}_k = -G\mathbf{a}_k$, where G is called the reduced-order control gain matrix. In terms of the full-state variable $\mathbf{q}_k$, the control input will be $-GL^H\mathbf{q}_k$. This is the motivation to use a weighted estimation error given by

$$J'(\mathbf{q}^n, \Theta) = \frac{1}{2} \sum_{k=1}^{n} \left\| G(L^H\mathbf{q}_k - \mathbf{a}_{k|k}) \right\|^2. \tag{4.5}$$

Due to the principle of separation of estimation and control for linear systems [153], the control matrix G is independent of the sensor locations with simplifies the derivation of the gradient of the objective function J' with respect to the sensor locations. This objective function can be viewed as weighted norm of the estimation error where the directions that contribute to the feedback control input are given higher weights.

[1] The inverse of the Fisher information matrix gives the lower bound for the estimation error covariance matrix of any unbiased estimator (such as the Kalman filter). This is known as the Cramer-Rao Inequality [45].

4.3.3. Optimization Problem

The constrained optimization problem we seek to solve is

$$
\begin{aligned}
\min_{\Theta} \quad & J(\mathbf{q}^n, \Theta) \\
\text{s.t.} \quad & \mathbf{a}_{0|0} = E[\mathbf{a}_0], \\
& \mathbf{a}_{k+1|k} = \ A\mathbf{a}_{k|k} + L^H B\mathbf{u}_k, && \forall k \in [n-1] \\
& \mathbf{a}_{k+1|k+1} = \ \mathbf{a}_{k+1|k} + K(\mathbf{z}_{k+1} - C_\Theta L\mathbf{a}_{k+1|k}), && \forall k \in [n-1] \\
& \Sigma = A\Sigma A^H + L^H W L - A\Sigma L^H C_\Theta^H S^{-1} C_\Theta L \Sigma A^H, \\
& K = \Sigma L^H C_\Theta^H S^{-1}, \\
& S = C_\Theta L \Sigma L^H C_\Theta^H + V, \\
\text{where} \quad & \mathbf{a}_{k|k} \in \mathbb{C}^r, && k \in [n] \\
& \mathbf{a}_{k+1|k} \in \mathbb{C}^r, && k \in [n-1] \\
& \Sigma \in \mathbb{C}^{r\times r}, K \in \mathbb{C}^{r\times p}, S \in \mathbb{C}^{r\times r},
\end{aligned}
\tag{4.6}
$$

and $[n]$ denotes the set $\{0, 1, \cdots, n\}$. Note that the optimization problem has $O(r^2 + rn + rp)$ variables and $O(r^2 + rn + rp)$ constraints. Importantly, the number of variables and constraints both are independent of the size m of the state variable $\mathbf{q}$ and scale linearly with the length n of the time horizon and the number of sensors p. This scaling of the size of our optimization problem is favourable in engineering flow applications, where the size m of the state vector is much larger than the number of observations n and the number of sensors p.

4.3.4. Adjoint Equations

We use the Lagrangian formalism to derive the adjoint equations to solve the optimization problem (Eq. (4.6)). We first construct a Lagrangian as follows,

$$\begin{aligned}\mathcal{L} = J - \boldsymbol{\eta}_{0|0}^H(\mathbf{a}_{0|0} - \mathbf{a}_0) - \sum_{k=0}^{n-1} \boldsymbol{\eta}_{k+1|k}^H(\mathbf{a}_{k+1|k} - A\mathbf{a}_{k|k} - L^H B\mathbf{u}_k) \\ - \sum_{k=1}^{n} \boldsymbol{\eta}_{k|k}^H(\mathbf{a}_{k|k} - \mathbf{a}_{k|k-1} - KC_\Theta(\mathbf{q}_k - L\mathbf{a}_{k|k-1})) \\ - \operatorname{tr}(\Lambda^H(\Sigma - A\Sigma A^H - W + A\Sigma C_\Theta^H S^{-1} C_\Theta \Sigma A^H)) \\ - \operatorname{tr}(H^H(K - \Sigma C_\Theta^H S^{-1})) - \operatorname{tr}(T^H(S - C_\Theta \Sigma C_\Theta^H - V)),\end{aligned} \tag{4.7}$$

where $\boldsymbol{\eta}_{k|k} \in \mathbb{C}^r, \boldsymbol{\eta}_{k|k-1} \in \mathbb{C}^r, \Lambda \in \mathbb{C}^{r\times r}, H \in \mathbb{C}^{r\times p}$ and $T \in \mathbb{C}^{r\times r}$ are adjoint variables, each corresponding to the forward variables $\mathbf{a}_{k|k}, \mathbf{a}_{k|k-1}, \Sigma, K$ and S respectively. J is defined in Eq. (4.4). Taking the variation of the Lagrangian $\mathcal{L}$ with respect to the adjoint variables and setting it to zero gives us the Kalman equations which are the constraints of our optimization problem.

If we set the variation with respect to $\mathbf{a}_{k|k}, \mathbf{a}_{k|k-1}, \Sigma, K$ and S to zero, we get the following adjoint equations,

$$\begin{aligned}\boldsymbol{\eta}_{n|n} &= -(L^H\mathbf{q}_n - \mathbf{a}_{n|n}) \\ \boldsymbol{\eta}_{k|k-1} &= \boldsymbol{\eta}_{k|k} - L^H(C_\Theta^H K^H \boldsymbol{\eta}_{k|k}), \\ \boldsymbol{\eta}_{k|k} &= A^H \boldsymbol{\eta}_{k+1|k} - (L^H\mathbf{q}_k - \mathbf{a}_{k|k}), \\ H &= \sum_{k=1}^{n} \boldsymbol{\eta}_{k|k}(C_\Theta(\mathbf{q}_k - L\mathbf{a}_{k|k-1}))^H, \\ \Lambda &= ZA^H\Lambda AZ^H + ZHS^{-T}C_\Theta, \\ T &= K^H(A^H\Lambda A\Sigma L^H C_\Theta^H - H)S^{-T},\end{aligned}$$

where $Z = (I - L^H C_\Theta^H K^H)$. At the optimal solution, both the adjoint equations and the Kalman filter equations must be simultaneously satisfied. Note that while the Kalman filter equations had nonlinear terms, all the adjoint equations are linear. Specifically, the equation for Λ is in the form of a discrete Lyapnov equation which allows us to use efficient algorithms to solve it [14].

The optimality condition is that the variation of $\mathcal{L}$ with respect to the sensor parameters is also zero. This gradient is given by,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \Theta} &= \frac{dC_\Theta}{d\Theta} \cdot \frac{\partial \mathcal{L}}{\partial C_\Theta}, \\ \frac{\partial \mathcal{L}}{\partial C_\Theta} &= \sum_{k=1}^{n} K^H \boldsymbol{\eta}_{k|k}^H (\mathbf{q}_k - L\mathbf{a}_{k|k-1})^H - S^{-1} C_\Theta L \Sigma A^H (\Lambda + \Lambda^H) A \Sigma L^H,\end{aligned} \tag{4.8}$$

where $dC_\Theta / d\Theta$ depends on the sensor model. We use a gradient descent to find the optimal value of Θ while solving the Kalman filter and the adjoint equations at each iteration to get the gradient. When the gradient with respect to the sensor location Θ is sufficiently close to zero, we have reached the optimal solution. Any first-order optimization method can be used to find the optimal solution. In this study we use gradient descent with the Armijo rule to determine the step-size at each iteration [19]. Adjoint-based optimization methods have been used to perform control [84], sensitivity analyses [131] and stability analyses [98] of fluid flows in previous studies.

Remark 4.3.1. In some applications it might be useful simultaneously to find the optimal model parameters for the state transition matrix A and the reduced-order basis L that minimizes the objective function J defined in Eq. (4.4). This formulation can be easily adapted to allow for that by computing the variation of the Lagrangian $\mathcal{L}$ with respect to A and L, respectively, and then using chain rule following Eq. (4.8). The gradient and details of model parameter optimization are shown in Appendix A.2.

4.4. Results

4.4.1. Linearized Ginzburg Landau Equation

In this section we apply the optimal sensor placement method on the linearized complex Ginzburg-Landau (GL) equation, previous described in Section 3.4.1. There have been many studies focusing on the stability [13] as well as optimal sensor and actuator placement [33, 113] for this system. The GL equations are described as follows.

$$\begin{aligned}\frac{\partial q}{\partial t} + \nu \frac{\partial q}{\partial x} &= \mu(x) q + \gamma \frac{\partial^2 q}{\partial x^2}, \\ \text{with} \quad \mu(x) &= \mu_0 - c_u^2 + \mu_2 x^2 / 2,\end{aligned}$$

where the real part of $q(x,t)$ represents a velocity or stream function perturbation amplitude. For this study we have taken the value of these constants μ, c_u, μ_0 and μ_2 corresponding to the supercritical, globally unstable regime case in [33, 130], also shown in Table 3.3. We model our sensors with Gaussian spatial support such that for each sensor at location $x_{s,i}$ with $i \in \{1, \cdots, p\}$

$$z_i(t) = \int_{-\infty}^{\infty} \exp\left(-\frac{(x - x_{s,i})^2}{2\sigma_i^2}\right) q(x,t)dx. \tag{4.9}$$

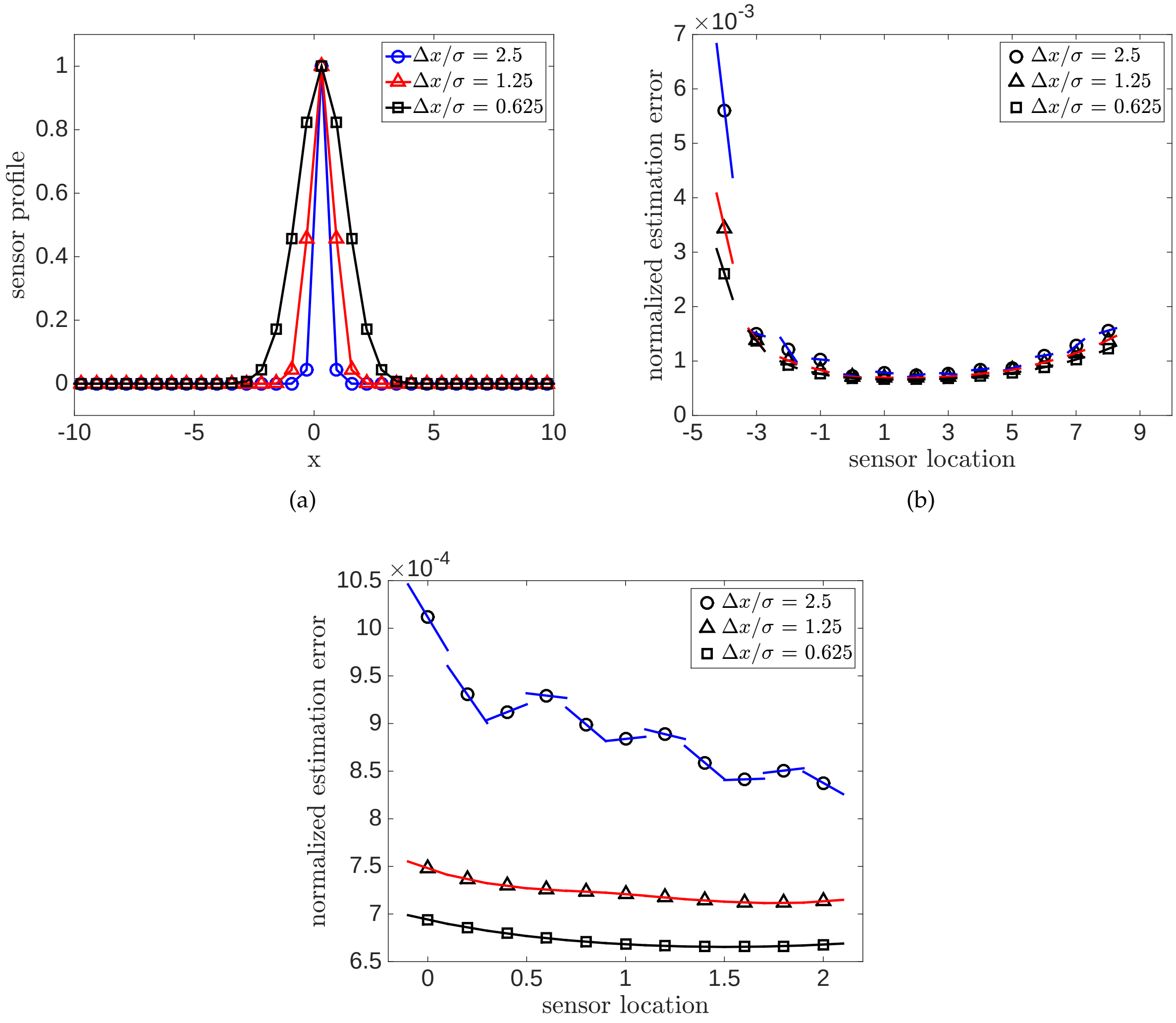
where $z_i(t)$ is the i^{th} component of the observation $\mathbf{z}(t) \in \mathbb{C}^p$ at time t and $\sigma_i \in \mathbb{R}^+$ determines the shape of the sensor profile.

In discrete space we use a spectral formulation of the derivative operators based on Hermite polynomials evaluated at the Hermite nodes [13]. Time discretization is performed using forward Euler time-stepping scheme. In this study we consider $m = 220$ grid points with the computational domain $x \in [-85, 85]$ and a time-step of $dt = 1$. For more details about the system the reader is referred to [13]. We denote the state transition matrix by $F \in \mathbb{C}^{m \times m}$ and the state variable at iteration k as $\mathbf{q}_k \in \mathbb{C}^m$. The integral in Eq. (4.9) is discretized using the trapezoidal integration rule. As such, the sensor model translates to $\mathbf{z}_k = C\mathbf{q}_k$ for each iteration k where $C = \widehat{C}M \in \mathbb{C}^{p \times m}$, in the absence of observation noise. Each row $\widehat{c_i}$ of $\widehat{C}$, corresponds to the spatial discretization of $\exp(-(x - x_{s,i})^2/2\sigma_i^2)$ into a $1 \times m$ vector and $M \in \mathbb{R}^{m \times m}$ is a diagonal weighting matrix corresponding to the trapezoidal integration rule. More formally, the matrix $C = [C_{i,j}], \forall i \in \{1, \ldots, p\}, j \in \{1, \ldots, m\}$ is defined as follows,

$$C_{i,j} = \begin{cases} \exp\left(-(x_j - x_{s,i})^2/2\sigma_i^2\right)(x_2 - x_1)/2, & \text{when } j = 1, \\ \exp\left(-(x_j - x_{s,i})^2/2\sigma_i^2\right)(x_{j+1} - x_{j-1})/2, & \text{when } 1 < j < m, \\ \exp\left(-(x_j - x_{s,i})^2/2\sigma_i^2\right)(x_m - x_{m-1})/2, & \text{when } j = m. \end{cases}$$

Figure 4.1a shows sensor profiles centered at $x_s = 0$ with three shape parameters σ. The discrete space and discrete time system in the presence of noise is described by the following equations.

$$\begin{aligned} \mathbf{q}_{k+1} &= F\mathbf{q}_k + \mathbf{w}_k, \\ \mathbf{z}_k &= C\mathbf{q}_k + \mathbf{v}_k, \end{aligned} \tag{4.10}$$



(a)

(b)

(c)

Figure 4.1.: Localization of sensor profiles increases the non-convexity of the objective function. (a) The sensor profiles for three values of variance $\sigma \in \{0.25, 0.5, 1\}$ at the same sensor location. (b) The normalized estimation error for different sensor locations using the three sensor profiles. The straight lines indicate the gradient with respect to the sensor location computed using the adjoint formulation. (c) The normalized estimation error for different sensor locations at a higher resolution clearly showing the non-convexity for $\sigma = 0.25$. The legend depicts the $\Delta x/\sigma$, where Δx is the smallest distance between consecutive grid points in the computational grid.

where $\mathbf{v}$ and $\mathbf{w}$ are zero mean complex Gaussian random variables with covariance matrices $W = wI_m$ and $V = vI_p$. For the training data, we set the scalars w and v to 0.01.

Effect of Sensor Localization

In this section we look closely at the objective function (Eq. (4.4)), defined by the reconstruction error, for the the sensor placement problem (Section 4.3.2). Specifically, we explore the effect of sensor model parameters on the location of the minima and the non-convexity of the objective function.

Suppose we seek to place a single sensor ($p = 1$) in the physical domain with a given shape parameter σ. As such, the sensor parameter we need to optimize is $\theta = \{x_{s,1}\}$. We consider the three values of the shape parameter $\sigma \in \{0.25, 0.5, 1\}$. As shown in Figure 4.1a, the shape parameter σ determines the apparent width of the sensor profile. The relation between the sensor width and the grid resolution is represented by the ratio $\Delta x/\sigma$, where Δx is the smallest distance between consecutive grid points in the computational grid. For the grid with $m = 220$ points, we have $\Delta x = 0.625$.

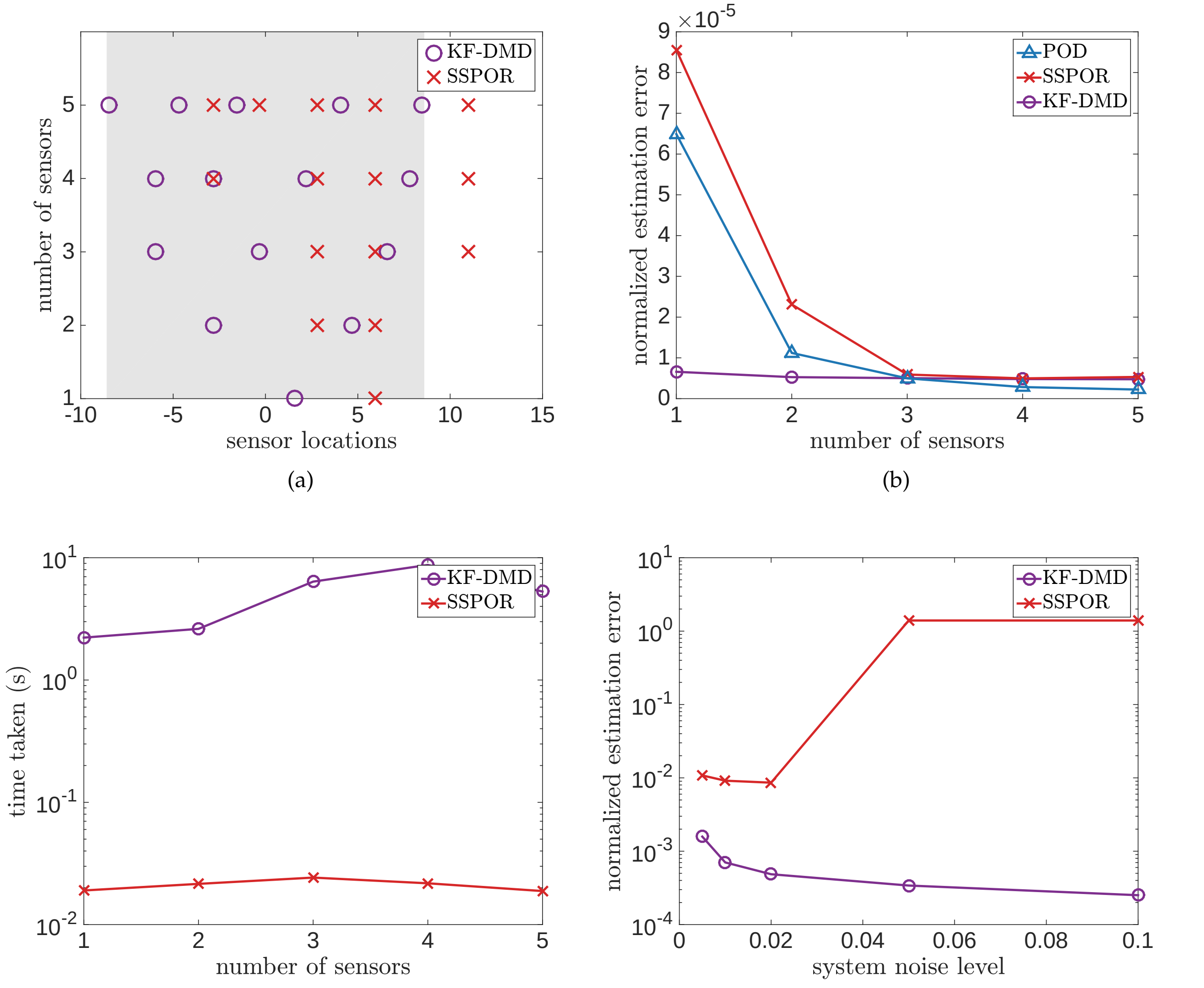
Since we are only interested in the effect of the sensor profile on the objective function, we use the full-order state transition matrix F to construct the observer that finds the estimate $\mathbf{q}_{k|k}$ of the state $\mathbf{q}_k$ at each iteration k. We set the initial condition as a Gaussian centered at $x_0 = 8$ with variance of $\sigma_0 = 5$ and collect snapshots for $n = 200$ iterations. Figure 4.1b shows the normalized reconstruction error

$$\text{error} = \frac{\sum_{k=1}^{n} \left\| (\mathbf{q}_k - \mathbf{q}_{k|k}) \right\|^2}{\sum_{k=1}^{n} \left\| \mathbf{q}_k \right\|^2}$$

for the three sensor shape parameters at different sensor locations. The lines in the plot represent the gradient computed using the adjoint formulation. Figure 4.1c shows the normalized estimation error with a finer resolution for the sensor location. In this figure we can clearly see that although the objective function is convex for $\sigma \in \{0.5, 1\}$, it is non-convex for $\sigma = 0.25$. In general, we observe non-convexity in the objective function as the apparent width of the sensor profile approaches the grid resolution. This means that solving the optimization problem for $\sigma = 0.25$ will be more difficult compared to larger σ values. One way to overcome this difficulty is to use the solution of a problem instance with a larger σ as initial guess for the optimization problem. We employ this strategy in the next section.

Effect of Dynamic Modeling

In this section we show the effect of dynamic modeling on the resulting estimation error. We also compare our results with another data-driven method called SSPOR [105]. SSPOR places



(a)

(b)

(c)

(d)

Figure 4.2.: The proposed method significantly outperforms SSPOR in terms of reconstruction using sparse sensing. (a) The location of sensors obtained using KF-DMD and SSPOR for different number of sensors. The amplification region in the physical domain is shaded gray. (b) The normalized estimation error for increasing number of sensors using KF-DMD, SSPOR and POD. (c) Time taken by KF-DMD and SSPOR for placing different number of sensors. (d) The normalized estimation error for increasing system-level noise using KF-DMD and SSPOR.

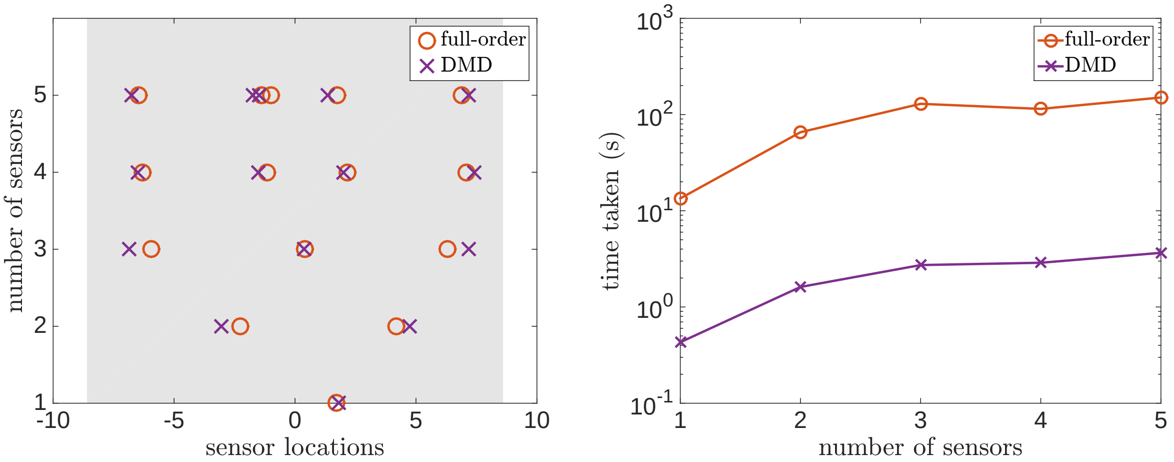
point-sensors in the physical domain to generate a reconstruction of the full state variable. It employs a heuristic to solve the D-optimal[2] sensor selection problem while neglecting the system noise and only considering the measurement noise. In comparison, our method models the system and the measurement noise as zero-mean Gaussian random variables and solves the A-optimal sensor placement problem using a data-driven approximation of the error covariance

[2] D-optimality criteria is to minimize the determinant of the covariance matrix of the estimation error. Geometrically this is equivalent to minimizing the volume of the uncertainty ellipsoid for the estimates.

matrix. We model a point sensor as a Gaussian with low variance ($\sigma = 0.25$, see Fig. 4.1a) with the sensor location given by the mean of the Gaussian. As such, we have $\Theta = \{x_{s,1}, \cdots, x_{s,p}\}$ where p is the number of sensors. To overcome the non-convexity associated with localized sensor profiles, we successively decrease the sensor profile shape parameter from 1, halving it at each step until we reach 0.25 and solving for the optimal sensor locations Θ at each step. We use the same training data with 200 snapshots as in the previous section and use DMD to find a rank $r = 10$ approximation of the state transition matrix. Recall that the system noise level v and measurement noise level w are set to 0.01. Since we use the Kalman filter to find the optimal sensor location and DMD for the reduced-order modeling, we refer to our method as 'KF-DMD' in the following. Note that DMD can be replaced with other reduced-order modeling methods that provide a linear approximation of the flow dynamics and our approach will still be applicable.

Figure 4.2a shows the sensor locations obtained using our method (KF-DMD) and SSPOR for different numbers of sensor locations $p \in \{1, 2, 3, 4, 5\}$. The grey shaded region depicts the amplification region $x \in [-8.6, 8.6]$ in the physical domain. While KF-DMD places all the sensors in the amplification region, we see SSPOR places sensor locations outside this region when number of sensors $p > 2$.

Figure 4.2b shows the normalized estimation error for KF-DMD and SSPOR-DMD along with SSPOR for different numbers of sensors p. The POD results show p-rank approximation of the input data matrix. It provides the theoretical lower limit for methods like SSPOR that do not use dynamical modeling for estimating the state variable. When the number of sensors is more than 3, the POD results are better than KF-DMD and SSPOR. However, in practice we want to use as few sensors as possible. For fewer sensors than three, we see KF-DMD perform much better than SSPOR and POD in reconstruction. Compared to SSPOR, we can see that KF-DMD gives superior performance for any number of sensors being placed in the domain. KF-DMD also provides the flexibility of user-defined sensor profiles while SSPOR can only place point sensors. This performance comes at the cost of computational time as seen in Figure 4.2c. An additional advantage of KF-DMD is robustness towards noise, shown in Figure 4.2d. The figure shows normalized estimation error incurred by KF-DMD and SSPOR while placing a single sensor for increasing system level noise v. KF-DMD incurs lower errors for all values of system noise than SSPOR. To summarize, we show that dynamical modeling, using methods such as DMD, not



(a) (b)

Figure 4.3.: Data-driven reduced-order modeling provides a good approximation of the optimal sensor locations at a reduced cost. (a) The location of sensors obtained using the DMD reduced-order model and the full-order model (baseline) for different number of sensors. (b) Time taken to place different number of sensors using the DMD reduced-order model and the full-order model.

only provides the ability to do prediction, but also improves reconstruction when the number of sensors is small.

Data-driven Optimal Sensor Placement

In the data-driven setting, we do not have access to the full-order model F. In this case, we use non-intrusive reduced-order modeling methods to find a low-order approximation of F. Similar to the previous section, we use DMD to learn a rank $r = 10$ approximation of the state transition matrix from the $n = 200$ snapshots of the flow. The resulting low-rank state transition matrix A is used for placing sensors in the physical domain. We compare the results of this completely data-driven approach to sensor locations obtained using the full-order matrix F under the same settings. We refer to the results obtained using the full-order model as the baseline. For both the cases, we fix the shape parameter of all the sensors to $\sigma = 2$. Alternatively, the shape parameters of the sensors can also be optimized, which we show in Appendix A.3.

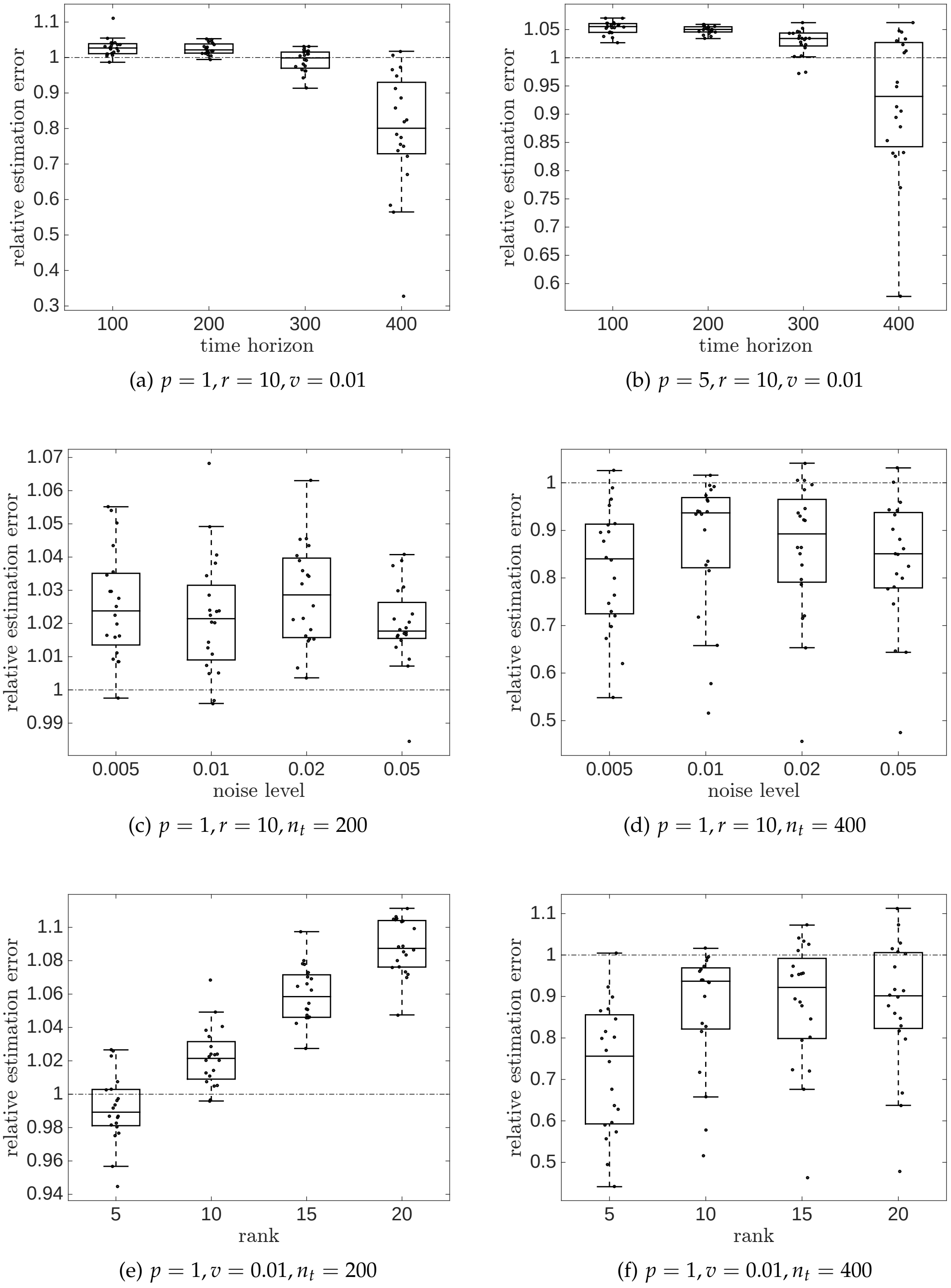
Figure 4.3a shows the resulting sensor locations using the reduced-order model (purple cross) and the full-order model (orange circle). We can see that the reduced-order model gives a very good approximation of the optimal sensor locations even for placement of multiple sensors simultaneously. Figure 4.3b shows the time taken by the reduced-order model compared to the full-order model. Here we see that the reduced-order model takes more than an order of magnitude less time compared to the full-order model to get the optimal sensor locations.

Next, we compare the performance of the completely data-driven approach with the baseline. Figure 4.4 show the relative estimation error from the comparison of the two methods. A relative estimation error of greater than one means that the data-driven approach performs better than the baseline. We demonstrate the performance of our method at different values of four independent parameters which are number of sensors (p), rank of the reduced-order model (r), noise level (v) and time horizon of estimation (n_t). For each set of parameter values, we generate testing data by running 20 independent simulations with different initial conditions. The initial conditions are Gaussian functions with $\sigma = 5$ and mean samples from $8 + \mathcal{N}(0, 1)$. Recall that the training data for both the methods has $n = 200$.

Surprisingly, in Fig. 4.4a and Fig. 4.4b we see that, even though the baseline uses the full-order model, our data-driven approach performs better than the baseline when time horizon of the testing data is less than the time horizon of the training data $n = 200$. The relative estimation error is close to 1 even when $n_t = 300$ but deteriorates for $n_t = 400$. This demonstrates the robustness of our proposed method in terms of the initial condition as well as the time horizon.

Figure 4.4c and 4.4d show the relative error is only moderately affected by the noise level in the testing data. We show the relative error for $p = 1$, $r = 10$ and two different time horizons, $n_t = 200$ (Figure 4.4c) which matches the training data and $n_t = 400$ (Figure 4.4d). Note that the training data was generated with a noise level of $v = 0.01$. In Figure 4.4d we see that for $n_t = 400$, when the peak performance of our method is achieved when the noise level in the testing data exactly matches the noise level that was used in the training data.

Finally, in Figure 4.4e and 4.4f we look at the effect of rank r on the relative estimation error for fixed values of p, v and n_t. Figure 4.4e shows the results for $n_t = 200$ and Figure 4.4f shows the results for $n_t = 400$. Both the figures show a positive trend in the relative estimation error for increasing rank of the reduced-order model. The improvement is stronger when the time horizon is $n_t = 200$ which matches the time horizon of the training data. These results show that increasing the rank of the reduced-order model improves the predictive power of the proposed method.



(a) $p = 1, r = 10, v = 0.01$

(b) $p = 5, r = 10, v = 0.01$

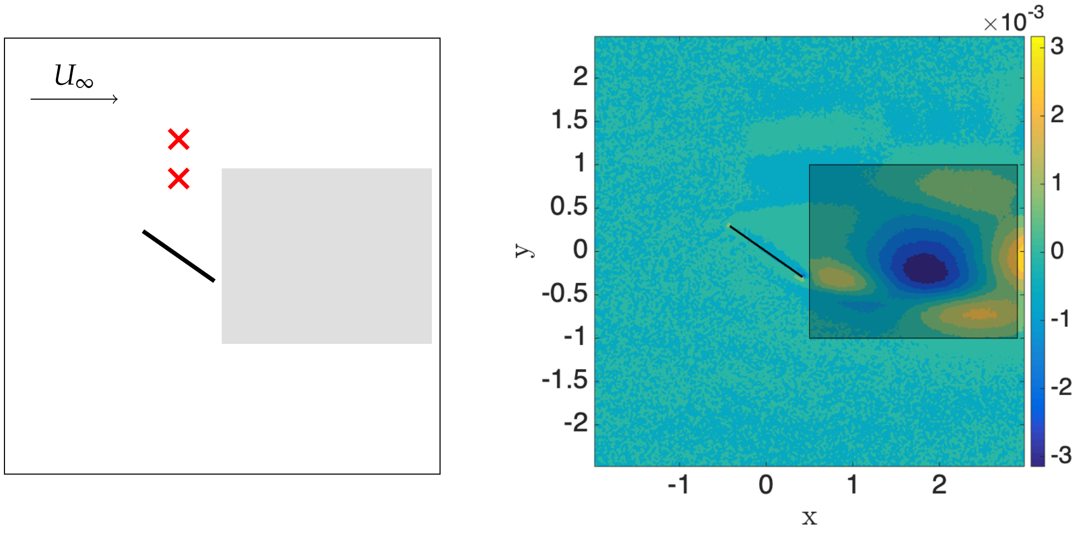
(c) $p = 1, r = 10, n_t = 200$

(d) $p = 1, r = 10, n_t = 400$

(e) $p = 1, v = 0.01, n_t = 200$

(f) $p = 1, v = 0.01, n_t = 400$

Figure 4.4.: Data-driven reduced-order modeling generates low error for reconstruction as well as prediction. We demonstrate the performance of our method at different values of four independent parameters – number of sensors (p), rank of the reduced-order model (r), noise level (v) and time horizon of estimation (n_t). We show the relative estimation error of the reduced-order model compared to the full-order model (baseline) is shown for 20 independent simulations with random initial conditions for each set of test parameters. Relative error greater than 1 indicates that the reduced-order model outperformed the full-order model. The training data for all the results is the same with a time-horizon of $n = 200$.



(a) (b)

Figure 4.5.: Schematic for flow over an inclined flat plate geometry. (a) The black line indicates the position of the flat plate in the computational domain. The red crosses show the actuator location and the gray shaded region is the region of interest to place the sensors. (b) The region of interest to place sensors is chosen to encompass the locations where the value of the control gain matrix G is large.

4.4.2. Observer-based Feedback Control of Flow over Inclined Flat Plate

In the previous section, we looked at the performance of the proposed method for reconstruction and prediction applications. In this section we show that our proposed method can be used for strategically placing sensors to yield good observer-based feedback control performance. Our goal is to stabilize the two-dimensional uniform flow approaching an inclined flat plate. The freestream flow is at a low Reynolds number of 100 and the flat plate is inclined at an angle of 35°. At these conditions, it has been shown that the steady state of the flow is unstable and the flow exhibits periodic vortex shedding [4]. We use an observer-based feedback controller (described in Section 2.2.2) to bring the system to steady state in the presence of system and measurement noise with a completely data-driven approach.

Computational Setup

The flow is simulated using the fast immersed boundary method developed in [35]. It is an efficient method to solve the incompressible Navier-Stokes equations. In order to achieve uniform flow conditions in the far field, a multi-domain approach is employed. The domain of interest is considered to be embedded in a series of domains, each twice-as-large as the preceding but with the same number of the uniform grid points. The numerical parameters of the flow are taken to follow the work of [4]. The grid size used is 250×250 and the domain of interest is given by $[-2, 3] \times [-2.5, 2.5]$ where the lengths are non-dimensionalized by the chord length of the flat plate, L. The center of the flat plate is located at the origin. Five domains, each with the same number of grid points are used for an effective computational domain that is 2^4 times larger than the domain of interest. The time-step is taken as $dt = 0.01L/U_\infty$ where U_∞ is the freestream velocity. We compute the steady-state solution $\overline{\mathbf{q}}$ by performing Newton-GMRES iterations on the nonlinear flow solution until convergence.

An actuator is placed near the leading edge of the flat plate. We use a simple model of a localized body force [3] at the actuator location near the leading edge of the flat plate. The instantaneous vorticity field generated by impulse control input of the actuator is

$$B(r) = c[(1 - ar_1^2)\exp(-ar_1^2) - (1 - ar_2^2)\exp(-ar_2^2)] \tag{4.11}$$

where $r_i^2 = (x - x_{c,i})^2 + (y - y_{c,i})^2$ for $i = 1, 2$. We set $(x_{c,1}, y_{c,1})$ and $(x_{c,2}, y_{c,2})$ to $(0, 1.342)$ and $(0, 0.89)$ respectively. The constants a and c determine the shape and strength of the control, respectively. In this study we use $a = 20$ and $c = 2$. These values were used in our previous study for stabilizing the flow with a full-state feedback controller [130] and therefore we know that the system is controllable with this actuator. Figure 4.5a shows that computational domain with the position of the flat plate (thick black line) and the actuator (red cross). Our goal is to place a Gaussian sensor given by,

$$C(r) = \exp\left(\frac{-r^2}{2\sigma^2}\right), \tag{4.12}$$

where $r = (x - x_s)^2 + (y - y_s)^2$, with a fixed variance of $\sigma = 2$, in the region of interest depicted by gray rectangle in Figure 4.5a. The region of interest has been chosen to enclose the locations where the absolute value of the control gain matrix is high, as shown in Figure 4.5b. The contours of control gain matrix looks noisy because the input data used for the reduced-order modeling is noisy. The x coordinate in the region of interest lies in $[0.5, 2.9]$ and the y coordinate lies in

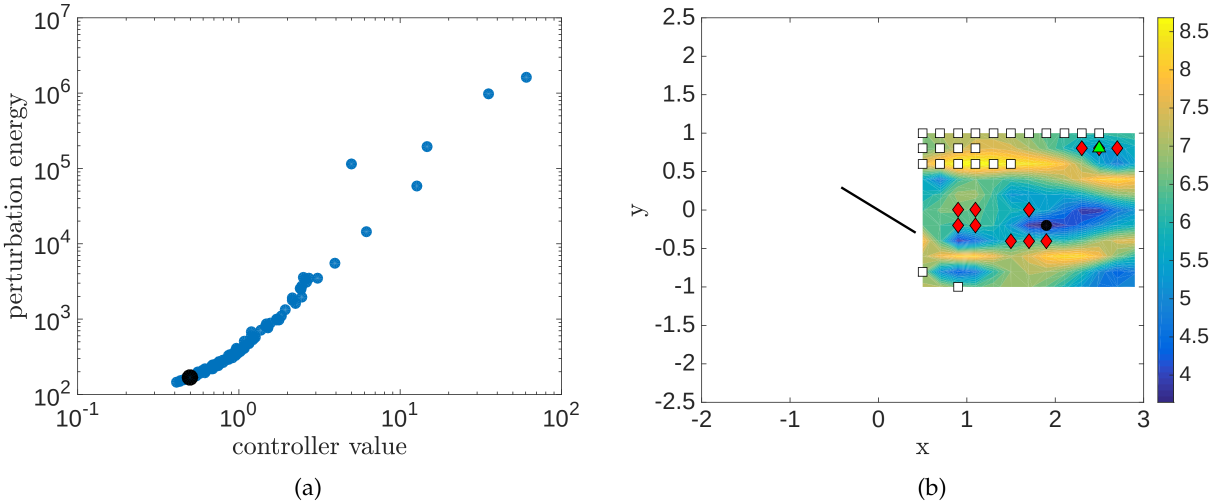


Figure 4.6.: Data-driven sensor placement shows near-optimal performance for observer-based feedback control of flow over an inclined flat plate. (a) Perturbation and controller energy for 143 independent simulations each with a different sensor location. The red marker shows the optimal sensor location predicted by the proposed method. (b) The landscape of the log of the predicted control objective function. The black marker indicates the predicted optimal sensor location while the red marker shows the sensor locations that outperformed the prediction. Green marker is the true optimal sensor location and the white markers indicate simulations that were numerically unstable.

$[-1, 1]$. This control gain matrix is build using a data-driven reduced-order model of the flow and is independent of the sensor location (as discussed in Section 2.2.2). It signifies the region in which the controller will be most sensitive to perturbations.

Governing Equations

Let the nonlinear evolution of the vorticity field be described as

$$\begin{aligned} \mathbf{q}_{k+1} &= \phi_T(\mathbf{q}_k) + Bu_k + \mathrm{diag}(\alpha(T))\mathbf{w}_k, \\ \mathbf{y}_k &= \psi(\mathbf{q}_k) + \mathrm{diag}(\gamma)\mathbf{v}_k, \end{aligned} \tag{4.13}$$

where k is the iteration number, ϕ_T is the nonlinear solver that advances the state $\mathbf{q}$ by T timesteps (each timestep has $dt = 0.01L/U$), B is the actuator described in Eq. (4.11) while $\mathbf{w}_k$ and $\mathbf{v}_k$ are vectors of standard normal Gaussian variables with appropriate dimensions. The system noise level $\alpha(T)$ is a linear function of T and the measurement noise is independent of T. Similar to our study in Chapter 3, we set $\gamma = 0.01$ and $\alpha = 5 \times 10^{-4}$. Note that at $T = 20$, the measurement and system noise are of the same strength. When we collect full-state observations for the training data, we set $\psi(\mathbf{q}_k) = \mathbf{q}_k$ and during observer-based feedback control for a given sensor location we set $\psi(\mathbf{q}_k) = C\mathbf{q}_k$ where C is described in Eq. (4.12).

Training Data

The snapshots for the study are generated from the impulse response to the actuator with the steady-state $\overline{\mathbf{q}}$ as the initial condition. The steady state of the flow is subtracted from the snapshots so that we work with the dynamics of the perturbations (deviation from the steady state). We generate two sets of training data, first to evaluate the performance of the controller with the objective function defined in Eq. (4.5) and second for building the reduced-order model.

For the first set of training data to use in the objective function we generate 250 snapshots with observations of system described in Eq. (4.13) with $T = 20$. For learning the reduced-order model using DMD (see Section 2.1.3), we generate snapshots with observations of system described in Eq. (4.13) with $T = 5$. The data matrices used by the DMD algorithm are constructed with 250 pairs of snapshots that are 4 iterations apart so that the total timesteps between the pair of snapshots is 20 and matches the first set of training data. Following our previous work [130], a linear approximation of the flow dynamics of rank $r = 20$ is computed using DMD. In our previous study, we stabilized this flow with a full-state feedback controller built using a rank $r = 20$ state transition matrix [130].

Setup of the Optimization Problem

Since our goal is to perform flow control, we employ the control-oriented objective function (Eq. (4.5)). The control gain matrix G is computed by solving a discrete Ricatti equation, as described in Section 2.2.2. The β parameter, which determines the trade-off between minimizing the perturbation and the control input in the reduced-order system, is set to 10^7. This is because the size m of the state vector $\mathbf{q}$ is much larger than the size d of the control input $\mathbf{u}$. As discussed in Section 4.3.2, the control gain matrix G is independent of the sensor locations, which simplifies the gradient of the new objective function J' with respect to the sensor locations.

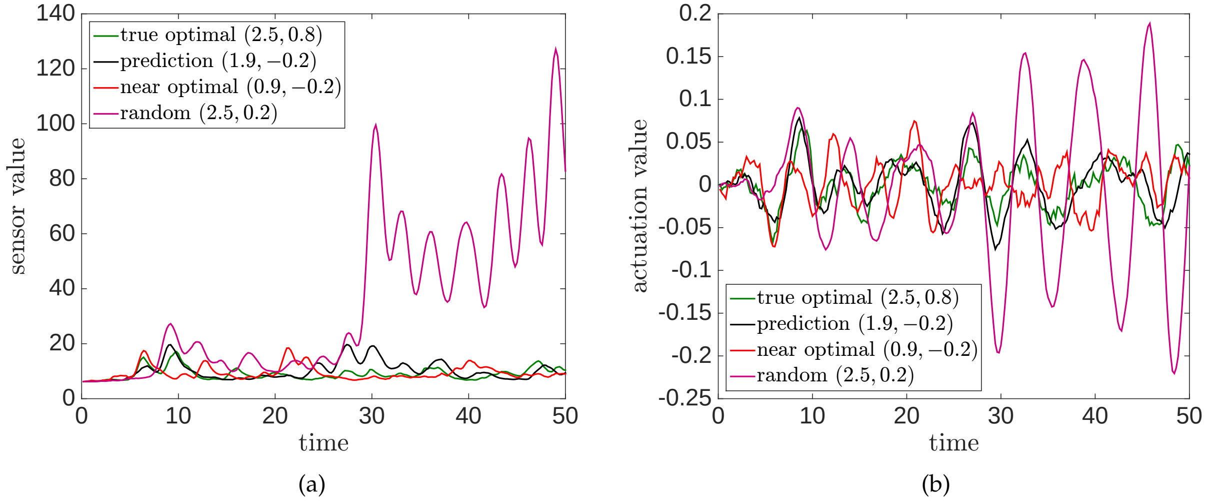


Figure 4.7.: The performance of the controller with the sensor at the predicted optimal sensor location is close to the performance of the controller with the sensor at the true optimal sensor location. Value of the (a) sensor and (b) actuator as a function of time for simulations with the sensor placed at four different sensor locations.

Optimal Sensor Placement

We discretize the region of interest with a uniform grid within $[0.5, 2.9] \times [-1, 1]$ with $\Delta x = 0.2$ and $\Delta y = 0.2$, i.e. $x_s \in \{0.5, 0.7, \cdots, 2.7, 2.9\}$ and $y_s \in \{-1, -0.8, \cdots, 0.8, 1\}$, to get 143 candidate sensor locations.We choose the candidate sensor location that has the smallest value of the objective function in Eq. (4.5). The predicted optimal candidate sensor location reported by our method is $(x_s, y_s) = (1.9, -0.2)$.

To evaluate the performance of our method, we run 143 independent nonlinear control simulations, one for each candidate sensor location in the region of interest, for 250 iterations (same as the length of the training data). Figure 4.6a shows the perturbation energy and the input control energy of the 143 simulations. The positive correlation between the two quantities is due to the use of a feedback controller. The predicted optimal solution predicted by our method is shown with a black marker. Even though it is not the true optimal solution, there are only 11 sensor locations that perform better than our prediction. This means that our prediction is in the top 90th percentile of the probed sensor locations in the region of interest. The sensor locations that perform better than our prediction are shown in Figure 4.6b. The true optimal solution $(2.5, 0.8)$ is shown with a green marker whereas our prediction of optimal sensor location is shown with a black marker. The contour plots show the prediction of estimation error using the DMD reduced-order model. The white markers represent sensor locations for which the controller was not able to stabilize the flow and the numerical simulation blew up. Running all the 143 nonlinear simulations took 10 hours on an Intel Xeon Phi 7250 with 64 cores. On the other hand, the optimal sensor location prediction using our method took only a few minutes ($\approx$ 10 mins) on a single core of Intel Core i7 for the optimization method to converge and training data worth a single nonlinear simulation.

Figure 4.7 shows the sensor values and actuator value for 4 simulation corresponding to different sensor locations. We show that the sensor value observed and the actuator response of the simulation with the true optimal sensor $(2.5, 0.8)$ location are very close to the simulation with the optimal sensor location predicted by our method $(1.9, -0.2)$. For comparison we show the results corresponding to a sensor location $(0.9, -0.2)$ that outperforms our prediction and a randomly chosen sensor location $(2.5, 0.2)$. These results show that our method can be used to find near-optimal sensor placement for control applications with a completely data-driven approach.

4.5. Conclusion

We introduce a completely data-driven method for sensor placement in fluid flows using full-state noisy observations. The full-state observations are first used to generate a linear approximation of the dynamics of the flow. Any modeling technique that generates a linear model of the system dynamics can be used to generate the reduced-order model. Under the assumption of zero-mean Gaussian noise, Kalman filter equations provide closed-form expressions for the filtered estimates of the state variable in terms of the sensor locations. These expressions provide constraints to the sensor placement optimization problem. We propose different objective functions for reconstruction, prediction and flow control applications. For instance, for accurate reconstruction or prediction, we propose minimization of the trace of an empirical approximation of the estimation error covariance matrix. We derive adjoint equations to efficiently compute to gradient of the objective function with respect to multiple sensor locations simultaneously. The performance of our method for reconstruction and prediction is demonstrated on the complex linearized Ginzburg-Landau equations in the globally unstable regime. Further, using a control-oriented objective function, we place sensors in the wake of over an inclined flat plate to yield the near-optimal observer-based feedback control performance.

There are several avenues for future research. This method in its current form can be used to solve time-varying optimal sensor locations and simultaneous optimization of sensor and model parameters. It would be interesting to explore some information theoretic objective functions as well. We plan to extend the current approach to simultaneously place actuators and sensors in the flow. We can further increase the applicability of the method for sensor placement in nonlinear regimes of the flow by incorporating nonlinear terms in the reduced-order model or successive linearization of the flow dynamics.

CHAPTER 5

MODAL ANALYSIS OF MULTIPHASE FLOWS

This chapter addresses data-driven modal analysis and interface tracking of multiphase flows. Section 5.1 provides background and motivation for the problem. The rest of the chapter is organized in two main sections. The first section, Section 5.2, describes modal analysis of instabilities of a liquid core within a swirling co-axial airblast atomizer. The second section, Section 5.3, describes a novel method for interface reconstruction and tracking in multiphase flows. Section 5.4 presents conclusions of the study and proposes directions for future research.

5.1. Motivation

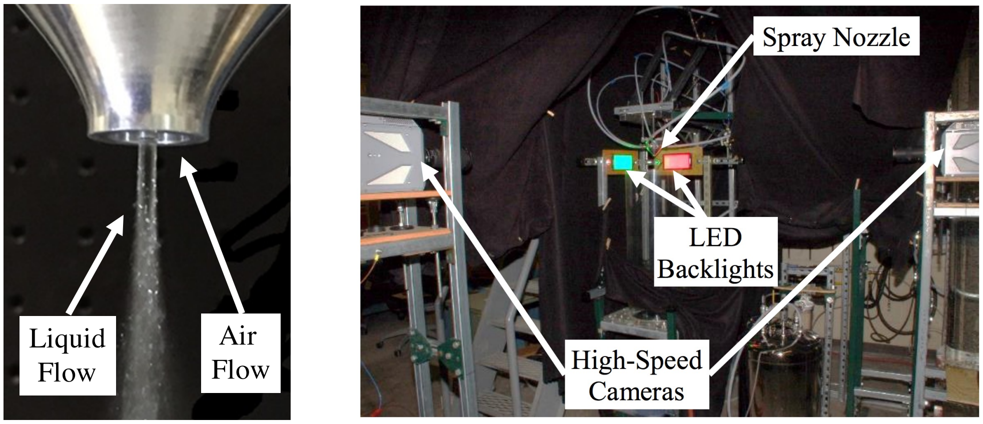
Liquid sprays are ubiquitous in engineering devices with applications in food processing, coating, printing and combustion chamber fuel injection [9]. The efficiency of these devices relies heavily on the atomization of the liquid spray and its dispersion into an ambient gas. For example, increased atomization of the liquid fuel into a combustion chamber leads to more effective evaporation and increases the efficiency of the fuel combustion [94]. Devices, such as airblast atomizers, that enhance the spray atomization and dispersion by triggering instabilities in the flow can therefore positively impact numerous industries.

The primary breakup of a liquid core by an airblast atomizer is a complex phenomenon. Several instabilities are simultaneously triggered in the flow that result in droplet generation and dispersion. One way to extract the spatio-temporal flow features is to perform modal analysis on the shadowgraph or back-lit image data of the spray. Several studies have applied projection-based modal analysis techniques such as Proper Orthogonal Decomposition (POD) [31] and Dynamic Mode Decomposition (DMD) [132, 144] on multiphase flows [11, 30, 70, 91, 92]. However, there are several limitations in these analyses. Firstly, these studies only use a single camera to capture the images of the flows, which leads to a projected view of the three-dimensional features of the flow. Secondly, although projection-based modal analysis provides valuable information about the spatio-temporal structure of the instabilities, they are not able to resolve spatial or temporal discontinuities, such as the liquid-gas interface, in the data. As a result, the information about the position and shape of the interface separating the liquid and gas phases in the flow is not recovered in these studies.

In this study we provide solutions to both these limitations. To accurately visualize the three-dimensional large-scale instabilities of the liquid jet, two synchronized high-speed cameras are used to simultaneously capture black-lit images of the air-bast atomized liquid jet from orthogonal directions. Instabilities due to flapping, swirling and radial expansion of the liquid jet are recognized. We show that these different forms of instabilities can not be accurately characterized with a single camera. We also propose a two-step approach for interface tracking using modal decomposition techniques. More specifically, we apply DMD on the optical flow field extracted from the back-lit images, which generates a rank-reduced spatially and temporally coherent velocity field that approximately reconstructs the flow field present in the back-lit imaging data. We use optical flow methods to compute a dense estimation of the flow field using two snapshots at a time. DMD is then applied on the flow field data to extract a reduced order model of the flow. The reduced-order velocity field is used to advect the interface location forward in time for reconstruction and prediction applications. The method is demonstrated on a toy problem of an oscillating drop, a numerically-simulated planar liquid jet with a gas co-flow using reduced order velocity extracted from volume of fluid data, and an experimental liquid spray with co-axial air-blast atomizer using back-lit imaging data.

5.2. Modal Analysis of Flow Instabilities

The use of Dynamic Mode Decomposition (DMD) [132, 144] to perform data-driven modal analysis of multiphase flows can be split into two parts, (i) identification of spatio-temporal modes of flow instability and (ii) estimating the relative amplitude of the identified instability modes in the flow. We provide details for both these components in A.4.1 and A.4.2, respectively. In this



(a) Spray nozzle

(b) Experimental setup

Figure 5.1.: The experimental setup to take stereographic images of liquid jet in a swirling co-axial airblast atomizer. (a) Orthogonal placement of the two synchronized high-speed cameras is shown. The flow is illuminated with LED backlights of different frequencies in the two directions. (b) A close-up view of the spray nozzle exit showing the liquid and the air flow.

section, we present the results of applying the modal analysis on a real experimental multiphase flow. Subsection 5.2.1 describes the experimental setup while subsection 5.2.2 describes the DMD results on the experimental data.

5.2.1. Experimental Setup

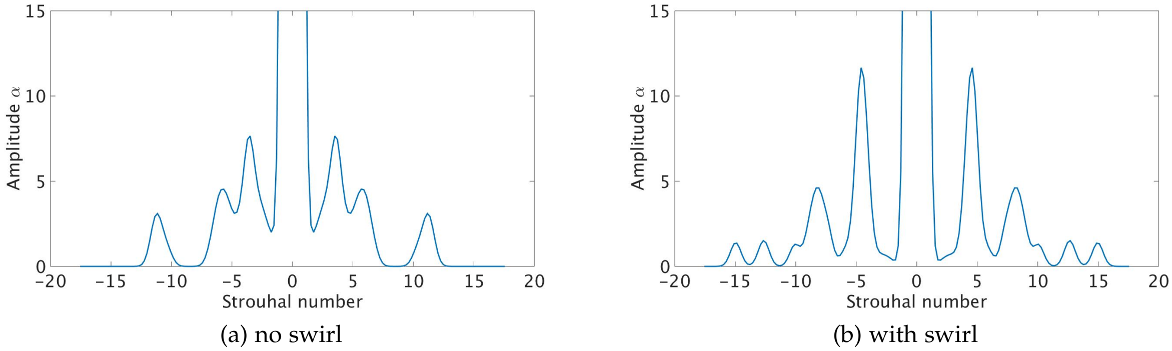
A two-fluid coaxial atomizer is used in this study (see Fig. 5.1a). The liquid is injected through the inner jet surrounded by a gas stream through an annulus shaped nozzle. The liquid nozzle has an inner diameter of $d_\ell = 2.1$ mm and an outer diameter of $D_\ell = 2.7$ mm, while the gas nozzle has an inner diameter of $d_g = 10$ mm. The liquid Reynolds number Re_ℓ and gas Reynolds number Re_g are defined as follows

$$Re_\ell = \frac{U_\ell d_\ell}{\mu_\ell}, \tag{5.1}$$

$$Re_g = \frac{U_g \sqrt{d_g^2 - D_\ell^2}}{\mu_g}, \tag{5.2}$$

where U_ℓ is the mean liquid velocity, μ_ℓ is the kinematic viscosity of the liquid, U_g is the mean gas velocity and μ_g is the kinematic viscosity of the gas.

We use distilled water as the liquid and compressed air as the gas. The gas injected through



(a) no swirl

(b) with swirl

Figure 5.2.: Amplitude spectrum of the DMD modes for the two experimental cases, case 1 (with no swirling gas velocity) and case 2 (with swirl ratio of 0.5) considered in the study. For both the cases, The liquid Reynolds number $Re_\ell = 1100$ and gas Reynolds number $Re_g = 21300$.

the gas nozzle can be imparted both axial and tangential velocities which can be controlled independently. We use PID controllers to regulate the liquid velocity and the gas velocities (both axial and tangential). Further details of the nozzle configuration can be found in [95, 103].

Stereographic back-lit imaging of the flow is performed using two synchronized high-speed cameras mounted perpendicular to each other. Figure 5.1b shows the placement of the cameras relative to the spray nozzle. The cameras are fitted with frequency-selective filters that match the illumination frequency of the high-intensity LED panels facing them. We capture images of the flow at a frame rate of 10000 frames per second and an exposure of 99 microseconds. More details on the back-lit imaging process can be found in [110].

5.2.2. DMD Spectrum and Modes

We consider two cases, case 1 and case 2, with similar flow conditions in this study. Both the cases have same liquid Reynolds number $Re_\ell = 1100$ and gas Reynolds number $Re_g = 21300$. The swirl ratio, defined as the ratio of the tangential gas flow rate to the axial gas flow rate, is 0 for case 1 and 0.5 for case 2. The Strouhal number for the oscillations of frequency f is defined as,

$$St = \frac{f d_\ell}{U_\ell}. \tag{5.3}$$

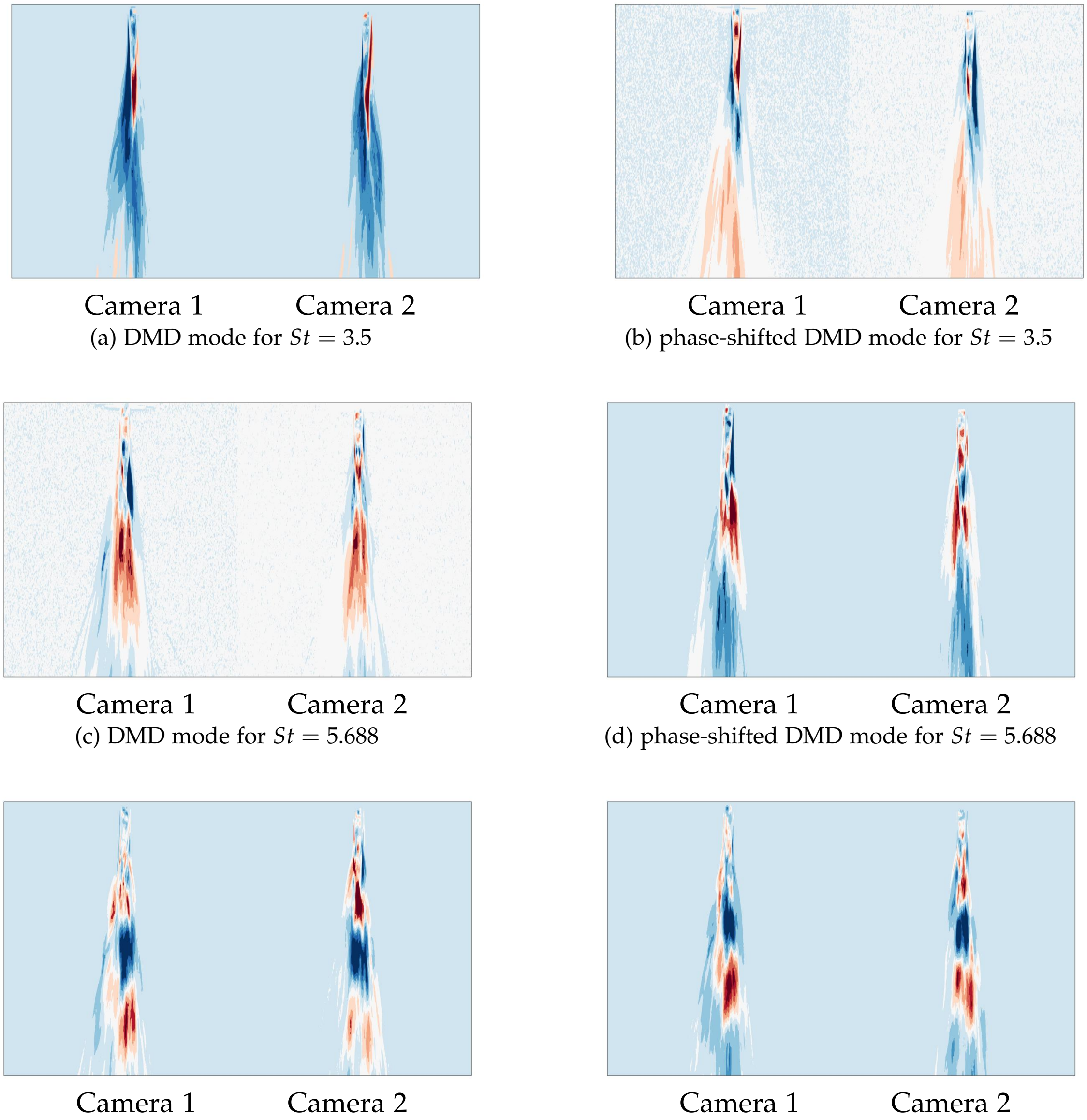
As described in A.4.1, DMD decomposes the spatio-temporal field $w(x, t)$ as linear combination of r modes as follows.

$$w(x,t) = \sum_{k=1}^{r} a_k(t)\phi_k(x)\alpha_k$$

where, α_k is the amplitude of the k-th mode, $\phi_k(x)$ is the spatial structure and $a_k(t)$ is the temporal dynamics of the mode. The decomposition is performed in a way so that the temporal dynamics $a_k(t)$ for each mode is given by $a_k(t) = \exp(i\omega_k t)$, where $i = \sqrt{-1}$ and $\omega_k \in \mathbb{C}$ determines the frequency and the growth or decay rate of the mode. Since the input flow field $w(x,t)$ for our problem has no imaginary parts, DMD will generate oscillatory modes (with $\omega_k > 0$) in conjugate pairs. This is because the complex eigenvalues in the eigenvalue decomposition of matrices with real entries occur in conjugate pairs. These pairs of modes oscillate in the same frequency but are *phase-shifted* from each other by $\pi/2$.

In all the figures showing the DMD modes, the positive and negative regions are depicted using red and blue colors, respectively. The modes extracted from images of the cameras are shown adjacent to each other in each figure. The limits of colormap for the images for all the DMD modes is $[-0.01, 0.01]$ using the amplitude definition given in Appendix A.4.

Case 1 (without swirl) Fig. 5.2a shows the amplitudes of oscillations with different frequencies for the swirl-free flow. The amplitude spectrum is symmetric about the origin as expected because the DMD algorithm generate two modes for each frequency. These modes are oscillating at a 90° phase difference with each other with the same frequency. The high amplitude around $St = 0$ is the contribution of the mean flow shown in Fig. A.4a in Appendix A.5. This mode provides information about the steady behaviour of the flow such as spray angle [23]. We see three other distinct peaks in the amplitude spectrum. The first peak is at $St = 3.5$ with the associated modes shown in Fig. 5.3a and Fig. 5.3b. We split these modes into three regions, described in Fig. 5.4, and characterize each region separately. In 'Region 1', we see that both the DMD modes and the phase-shifted modes have low intensity and the characteristics of the dynamics are not clear. However, in 'Region 2' we see a positive (red) lobe on the left and a negative (blue) lobe on the right for the DMD mode from the perspective of both camera 1 and 2. For the phase-shifted DMD mode, the perspective of camera 1 has the same parity as the DMD mode while camera 2



(a) DMD mode for $St = 3.5$

(b) phase-shifted DMD mode for $St = 3.5$

(c) DMD mode for $St = 5.688$

(d) phase-shifted DMD mode for $St = 5.688$

(e) DMD mode for $St = 11.16$

(f) phase-shifted DMD mode for $St = 11.16$

Figure 5.3.: DMD modes at different Strouhal numbers for the flow case 1 (with no swirling gas). For both the cases, The liquid Reynolds number $Re_\ell = 1100$ and gas Reynolds number $Re_g = 21300$.

has a negative lobe surrounded by positive region. This can be characterized as predominantly a flapping mode, although the phase-shifted DMD mode from the perspective of camera 2 exhibits features of radial expansion. Finally, in 'Region 3', we see positive intensity regions from both the perspective of both the cameras for the DMD mode, while the phase-shifted mode shows asymmetry, with a negative lobe on the right and positive lobe on the left. Again, this can be characterized as a mixture of flapping and radial expansion.

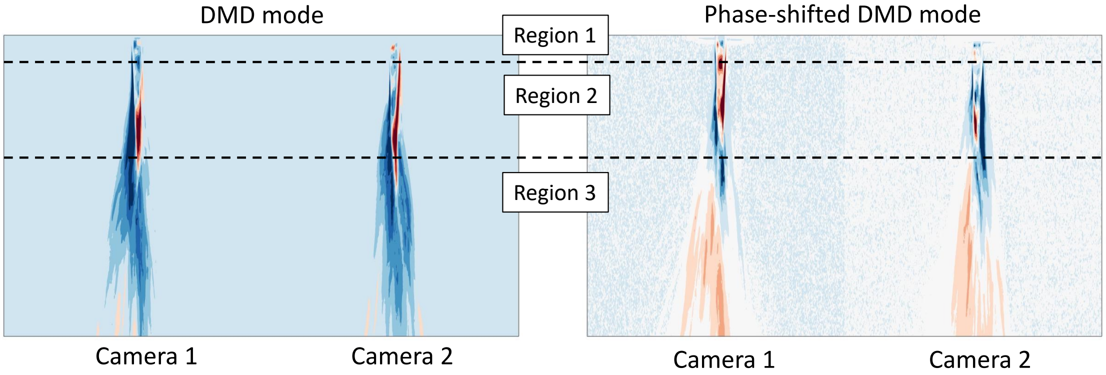
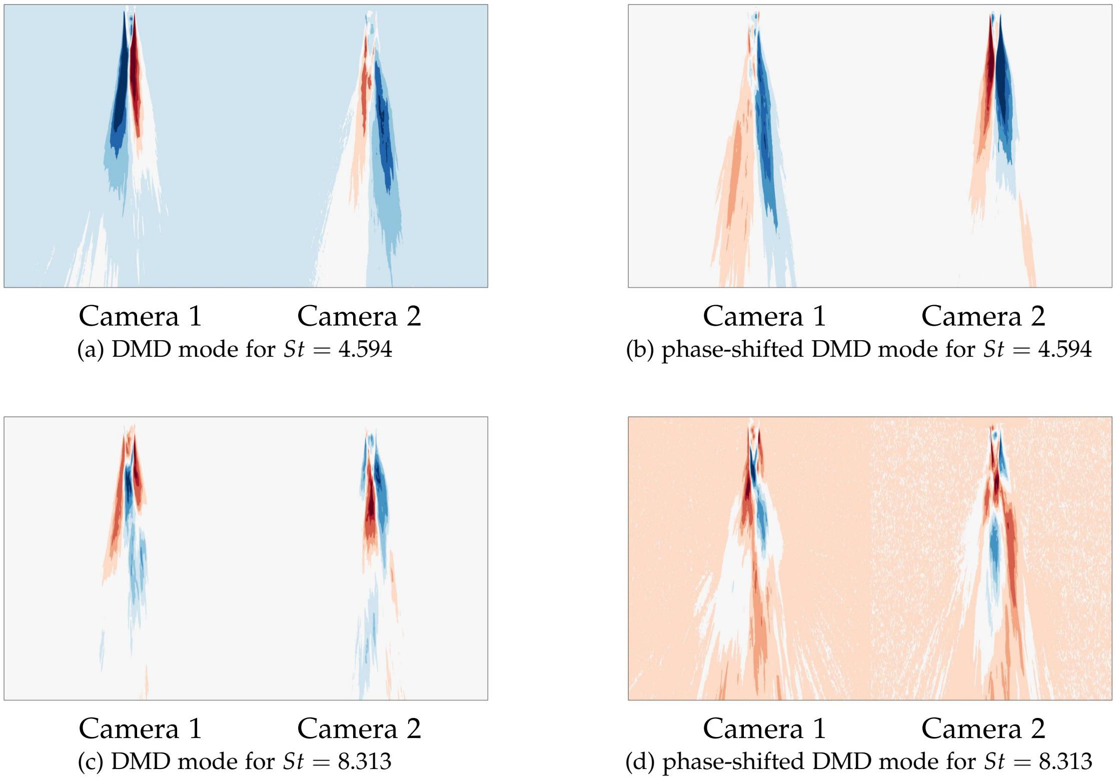


Figure 5.4.: Three regions showing different spatial structures in the mode corresponding to $St = 3.5$ of a liquid jet surrounded by a co-axial airblast atomizer.

The second and the third peak occur at $St = 5.688$ and $St = 11.16$ respective. Since 11.16 is close to twice of 5.688, we infer that the third mode is a higher harmonic of the second mode. The DMD modes for these frequencies are also shown in Fig. 5.3, where see a chain of alternating positive and negative lobes in these modes vertically stacked. These are characterized as 'wavy-breakup' modes in [92]. Interestingly, the number of lobes is higher in the modes that correspond to the higher harmonic frequency.

Case 2 (with swirl) Fig. 5.2b shows the amplitudes of oscillations with different frequencies while Fig. A.4a in A.5 shows the mean of all the flow snapshots, for the flow with swirl ratio of 0.5. This time we see two large peaks at $St = 4.6$ and 8.3, and two small peaks at higher frequencies of $St = 12.7$ and 14.9. In this study we ignore the smaller peaks and concentrate on the larger peaks at $St = 4.6$ and 8.3. The modes associated with $St = 4.6$ are shown in Fig. 5.5a and Fig. 5.5b. These modes have been characterized as flapping modes by the authors in [92]. However looking at the two phase-shifted modes together, we can infer that these are in fact modes associated with swirling of the liquid jet. In Fig. 5.5a we can see a positive lobe on the left and a negative lobe right for camera 1 while a negative lobe on the left and positive lobe on the right for camera 2. If the phase-shifted mode also had the same structure then this would have been consistent with flapping of the jet. However we see in Fig. 5.5b that from the perspective of both camera 1 and camera 2, the positive lobe on the right and negative lobe on the left. This means that the jet is in fact swirling about its axis which appears as though the jet is flapping when observed with a single camera. Fig. 5.6 shows a schematic of the evolution of this mode with a top view of the



(a) DMD mode for $St = 4.594$

(b) phase-shifted DMD mode for $St = 4.594$

(c) DMD mode for $St = 8.313$

(d) phase-shifted DMD mode for $St = 8.313$

Figure 5.5.: DMD modes at different Strouhal numbers for the flow case 2 (with swirl ratio of 0.5). For both the cases, The liquid Reynolds number $Re_\ell = 1100$ and gas Reynolds number $Re_g = 21300$.

liquid jet. This behaviour can be correctly characterized only if the liquid jet is imaged from two orthogonal perspectives.

For $St = 8.3$ the modes are shown in Fig. 5.5c and Fig. 5.5d. We see that some of the modes have a negative/positive lobe sandwiched between two positive/negative lobes while other modes have a clear asymmetry. This shows that the oscillation associated with this mode is mixture between radial expansion and flapping modes.

5.3. Interface Reconstruction and Tracking

In the previous section, we saw that while modal analysis of the experimental multiphase flow revealed the temporal dynamics and spatial structure of flow instabilities, the interface location is unclear. In this section we describe a novel data-driven interface reconstruction and tracking method for multiphase flows. We use data-driven modeling in conjunction with optical flow

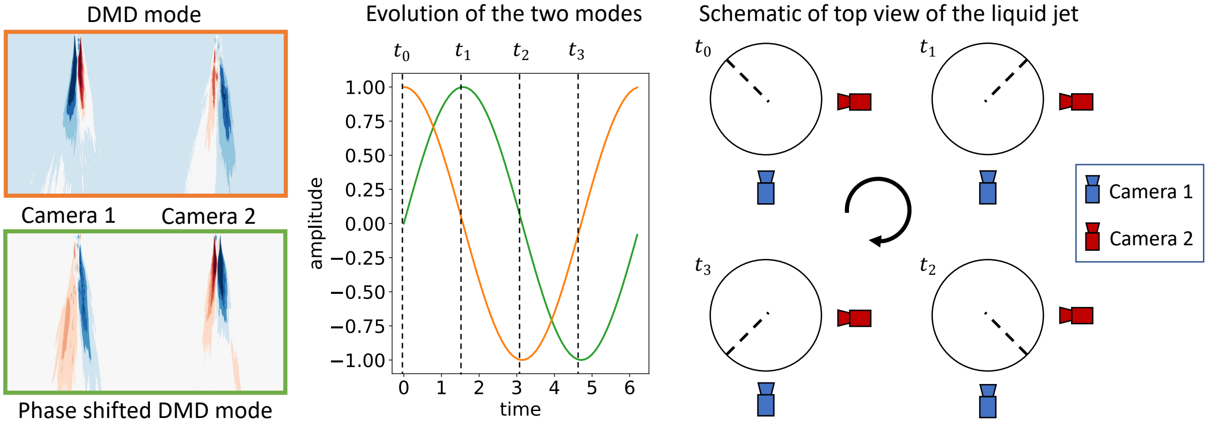


Figure 5.6.: This schematic shows the evolution of the swirling mode ($St = 4.6$) of a liquid jet in a co-axial airblast atomizer with gas swirl velocity ratio 0.5. Dashed lines in the diagrams on the right depict the position of the liquid jet with the center being the location of the nozzle.

methods to construct spatially and temporally coherent velocity field of the flow. Specifically, we propose a two step approach to compute the reduced-order velocity field. First, we use optical flow methods to compute a dense estimation of the flow field using two snapshots at a time. DMD is then applied on the flow field data to extract a reduced order model of the flow. This velocity field can be used to advect the liquid-gas interface forward in time.

5.3.1. Optical Flow Methods

Velocity field estimation from experimental data is of fundamental importance in fluid mechanics. Experimental image data generally comprises tracer particles populated in the domain of interest and advected by the surrounding flow field. The motion of the tracer particles in time is used to infer the underlying velocity field. One of the most widely used methods for flow velocity estimation is Particle Image Velocimetry (PIV) [123, 149] which uses local spatial correlation between consecutive images for global estimation of the velocity field. When the particle density is low, Particle Tracking Velocimetry (PTV) [102, 104] is used instead.

For multiphase flows, the most common non-intrusive visualization methods do not use tracer particles, such as visual light shadowgraphy [66, 68], back-lit imaging [55] and X-ray flow visualization [59, 60]. In back-lit imaging, an intense LED light is arranged behind the object of interest (e.g., a spray) and directed into a camera. A shadow of the liquid region is then captured where the liquid region is darker than the gas. In this study we are interested in recovering the velocity of the liquid-gas interface during the primary atomization of a liquid spray visualized through back-lit imaging.

Optical flow methods are used to construct a vector field that captures the apparent motion implied by consecutive images in time. Following the seminal work by [69], considerable progress has been made on applying optical flow methods to estimate fluid flow velocities [97, 121, 155]. For a detailed review of optical flow methods used for fluid flow estimation and their comparison with correlation based methods, the reader is referred to [96].

The optical flow problem is defined as follows. Let $\Omega \in \mathbb{R}^2$ be an image section and $(x, y) \in \Omega$ any point on it. Let t be the time within an interval $[0, T]$. The given image data can be represented by an intensity function

$$I : \Omega \times [0, T] \rightarrow I(x, y, t). \tag{5.4}$$

Our goal is to compute the velocity field $w(x, y, t) = [u(x, y, t), v(x, y, t)]^\top$ that minimizes the following objective function

$$J = \int_\Omega \| I(x + u\mathrm{d}t, y + v\mathrm{d}t, t + \mathrm{d}t) - I(x, t) \| \; \mathrm{d}x \tag{5.5}$$

at each time t. In the following sections we will describe the popular Horn-Schunck method and a streamfunction based formulation of the optical flow problem.

Horn-Schunck Method

The Horn-Schunck (HS) method [69] models the evolution of the flow as governed by the advection equation,

$$I_t(x, y, t) + w(x, y, t) \cdot \nabla I(x, y, t) = 0 \tag{5.6}$$

where the operator ∇ is the gradient operator such that $\nabla I(x, y, t) = [I_x(x, y, t), I_y(x, y, t)]^\top$ and $I_t(x, y, t)$ is the time-derivative of $I(x, y, t)$. This approximation holds for small displacements $w\,\mathrm{d}t$ between consecutive snapshot images. The Horn-Schunck vector field $w(x, y, t)$ is inferred by using the minimizer of

$$E(w) = \int_{\Omega} (I_t(x, y, t) + w(x, y, t) \cdot \nabla I(x, y, t))^2 d\Omega + \alpha \int_{\Omega} \|\text{div}(w)\|^2 \, d\Omega \tag{5.7}$$

where only the first term depends on the image data and the second term is a regularization term that penalizes the divergence of the velocity field and α is a user-selected parameter. Therefore the underlying velocity field is given by

$$w^* = \arg\min_{w} E(w) \tag{5.8}$$

This optimization problem is solved iteratively using the method described in [69] and shown here for completeness. Using calculus of variations for the energy function given in Equation 5.7, we get that u and v must satisfy the coupled equations,

$$I_x^2 u - I_x I_y v = \alpha^2 \nabla^2 u - I_x I_t, \tag{5.9}$$

$$I_x I_y u - I_y^2 u = \alpha^2 \nabla^2 v - I_y I_t. \tag{5.10}$$

At any time t, and spatial coordinates x and y, the spatial and temporal derivatives of the image intensity field needed in Eq. (5.7) are approximated by the following simple finite difference estimates,

$$\begin{aligned} I_x(x, y, t) \approx \frac{1}{4}\Big(& I(x + \mathrm{d}x, y, t) - I(x, y, t) + \\ & I(x + \mathrm{d}x, y + \mathrm{d}y, t) - I(x, y + \mathrm{d}y, t) + \\ & I(x + \mathrm{d}x, y, t + \mathrm{d}t) - I(x, y, t + \mathrm{d}t) + \\ & I(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t) - I(x, y + \mathrm{d}y, t + \mathrm{d}t)\Big), \end{aligned} \tag{5.11}$$

$$\begin{aligned} I_y(x, y, t) \approx \frac{1}{4}\Big(& I(x, y + \mathrm{d}y, t) - I(x, y, t) + \\ & I(x + \mathrm{d}x, y + \mathrm{d}y, t) - I(x + \mathrm{d}x, y, t) + \\ & I(x, y + \mathrm{d}y, t + \mathrm{d}t) - I(x, y, t + \mathrm{d}t) + \\ & I(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t) - I(x + \mathrm{d}x, y, t + \mathrm{d}t)\Big), \end{aligned} \tag{5.12}$$

$$\begin{aligned} I_t(x,y,t) \approx \frac{1}{4}\Big(& I(x,y,t+\mathrm{d}t) - I(x,y,t) + \\ & I(x+\mathrm{d}x,y,t+\mathrm{d}t) - I(x+\mathrm{d}x,y,t) + \\ & I(x,y+\mathrm{d}y,t+\mathrm{d}t) - I(x,y+\mathrm{d}y,t) + \\ & I(x+\mathrm{d}x,y+\mathrm{d}y,t+\mathrm{d}t) - I(x+\mathrm{d}x,y+\mathrm{d}y,t)\Big). \end{aligned} \tag{5.13}$$

where dx and dy are given by the pixel resolution of the images and dt is the camera frame rate.

We solve these equations iteratively. Using a finite difference based approximation for the Laplacians of u and v, we get the following iterative recursion for the new estimate for the velocities (u^{n+1}, v^{n+1}) based on the old estimates (u^n, v^n),

$$u^{n+1} = \bar{u}^n - I_x[I_x\bar{u}^n + I_y\bar{v}^n + I_t]/(\alpha^2 + I_x^2 + I_y^2), \tag{5.14}$$

$$u^{n+1} = \bar{u}^n - I_x[I_x\bar{u}^n + I_y\bar{v}^n + I_t]/(\alpha^2 + I_x^2 + I_y^2), \tag{5.15}$$

where

$$\begin{aligned} \bar{u}^n = & \frac{1}{6}\Big(u^n(x+\mathrm{d}x,y,t) + u^n(x,y+\mathrm{d}y,t) \\ & \quad + u^n(x-\mathrm{d}x,y,t) + u^n(x,y-\mathrm{d}y,t)\Big) + \\ & \frac{1}{12}\Big(u^n(x+\mathrm{d}x,y+\mathrm{d}y,t) + u^n(x+\mathrm{d}y,y-\mathrm{d}y,t) \\ & \quad + u^n(x-\mathrm{d}x,y+\mathrm{d}y,t) + u^n(x-\mathrm{d}x,y-\mathrm{d}y,t)\Big), \end{aligned} \tag{5.16}$$

$$\begin{aligned} \bar{v}^n = & \frac{1}{6}\Big(v^n(x+\mathrm{d}x,y,t) + v^n(x,y+\mathrm{d}y,t) \\ & \quad + v^n(x-\mathrm{d}x,y,t) + v^n(x,y-\mathrm{d}y,t)\Big) + \\ & \frac{1}{12}\Big(v^n(x+\mathrm{d}x,y+\mathrm{d}y,t) + v^n(x+\mathrm{d}y,y-\mathrm{d}y,t) \\ & \quad + v^n(x-\mathrm{d}x,y+\mathrm{d}y,t) + v^n(x-\mathrm{d}x,y-\mathrm{d}y,t)\Big). \end{aligned} \tag{5.17}$$

The natural boundary conditions for this discretization is a zero normal derivative at all boundaries. It is interesting to note that the new estimates of the velocity at some point (x, y), do not depend directly on the previous estimates of the velocity at that point.

Streamfunction Formulation

One of the limitations of optical flow methods is the color constancy constraint [61]. A direct consequence of this constraint is that only the velocity field normal to the interface can be recovered from the image data. To circumvent this problem, we use a global streamfunction [100, 101] to describe the underlying velocity field. The streamfunction is estimated by formulating a cost function similar to Equation (5.8) with an appropriate regularization term. The physical significance of using a streamfunction is the incompressibility constraint that is valid for fluid flows at low Mach numbers. Similar formulations using physics based constraints such as Hemholtz decomposition of the velocity fields [36] and spatial domain decomposition of the fluid flows [58] have been used successfully for atmospheric flow estimation from satellite images.

We impose the incompressibility constraint on the flow by introducing a streamfunction ϕ such that

$$(u, v) = \nabla_H \phi \tag{5.18}$$

where ∇_H is the symplectic gradient so that $\nabla_H \phi = (-\phi_y, \phi_x)$. The streamfunction is computed as the minimizer of

$$\begin{aligned} E_\alpha(\phi) = \int_\Omega (I_t(x,t) + \nabla I(x,t) \cdot \nabla_H \phi(x,t))^2 d\Omega &+ \\ \alpha \int_\Omega \phi_{xx}^2 + \phi_{xy}^2 + \phi_{yx}^2 + \phi_{yy}^2 d\Omega& \end{aligned} \tag{5.19}$$

where the second term is a regularization term similar to that used in the Horn-Schunck method. The optimization problem is to find the streamfunction that minimizes the cost function $E_\alpha(\phi)$ for some given α. We apply zero-velocity normal velocity boundary conditions at all sides of the domain which prevents the entry or exit of the flow from the domain. A second-order central finite difference stencil is applied to construct the first order spatial derivatives and the symplectic gradient operator. We define the operator A by

$$A\text{vec}(\phi) = \text{vec}(\nabla I \nabla_H \phi). \tag{5.20}$$

where vec is a function that vectorizes the matrix ϕ. The second order derivatives are computed using multiple operations of the first order derivatives. We denote the variation of the regularization term using the matrix B defined as follows,

$$B = D_{xx}^T D_{xx} + D_{xy}^T D_{xy} + D_{yx}^T D_{yx} + D_{yy}^T D_{yy}, \tag{5.21}$$

where the matrix D_{ij} is such that $D_{ij}\text{vec}(\phi)$ is the finite difference approximation of $\text{vec}(\phi_{ij})$ for $i, j \in \{x, y\}$. The minimizer of the objective function in Equation (5.19) is given by the solution of the following linear equation,

$$(A^T A + \alpha(B + B^T))\text{vec}(\phi) = -A^T \text{vec}(I_t). \tag{5.22}$$

We solve this equation using GMRES (Generalized Minimal Residual Algorithm) [127].

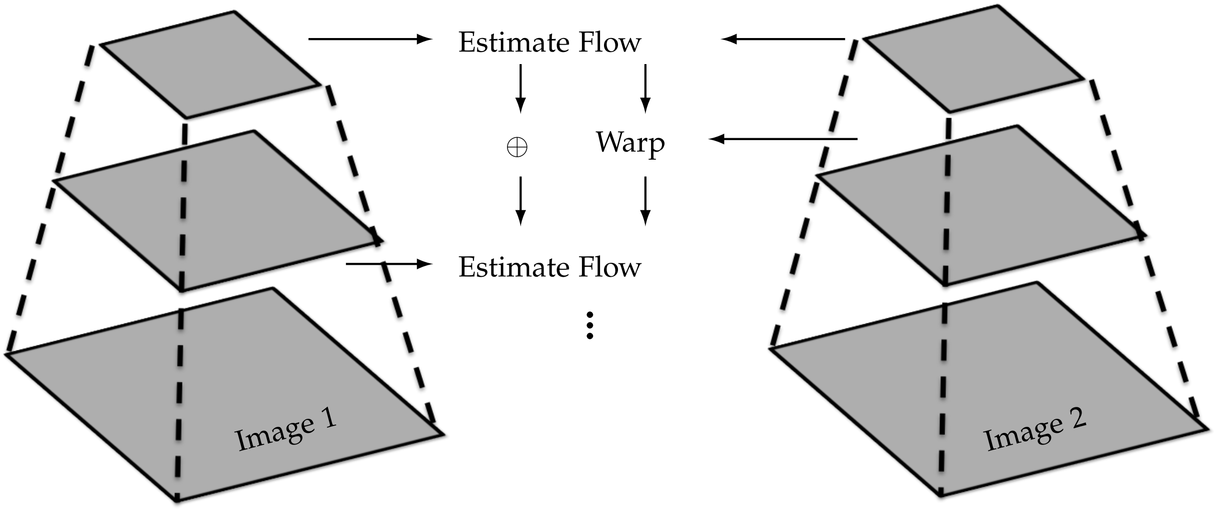


Figure 5.7.: Schematic of the coarse-to-fine resolution in the multi-resolution scheme used for optical flow estimation between two consecutive images. The ⊕ symbol is used to denote addition.

Multi-Resolution Scheme

The variational formulation used in optical flow estimation is limited to small displacements $w\mathrm{d}t$ between consecutive images. In order to handle large displacements of the interface and liquid droplets, as might occur when the image frame rate is low, we use a multiresolution scheme around the variational methods described in the sections above [61]. The multiresolution scheme, also known as pyramidal scheme, works by generating a series of incrementally coarse-grained images. This scheme can be applied with any optical flow estimation method. Let the two images at consecutive time steps be I_1 and I_2. For some optical flow estimation method, let the estimated velocity field be given by

$$w = F(I_1, I_2). \tag{5.23}$$

For a $d-$level pyramidal scheme, we first down-sample the images d times to generate two series of images with increasing resolution, $\{I_1^0, I_1^1, \cdots, I_1^d\}$ and $\{I_2^0, I_2^1, \cdots, I_2^d\}$ from I_1 and I_2 respectively such that $I_1^d = I_1$ and $I_2^d = I_2$. At each level, flow estimation is conducted on subsampled images to generate a coarse-grained velocity field. This velocity field is used to warp the finer level images and the whole process is repeated. More formally, at any level k, the estimated velocity field $w^k = (u^k, v^k)$ is given by

$$w^k = w^{k-1} + F(I_1^k, I_2^k(x + u^{k-1}\mathrm{d}t, y + v^{k-1}\mathrm{d}t)), \tag{5.24}$$

where the recursion is started with $w^0 = F(I_1^0, I_2^0)$. A schematic of the procedure is shown in Figure 5.7. In this study we use a 5 step pyramidal scheme with a 2× refinement at each level for both the Horn-Schunck and the streamfunction formulations.

5.3.2. Image Reconstruction

In this section we describe the method to advect the interface location using the estimated flow field. At each time t, image reconstruction using the estimated flow field $w^r(x,t)$ is performed using a displacement vector given by

$$\mathrm{d}x^r = \int_t^{t+dt} w^r \,\mathrm{d}t. \tag{5.25}$$

The resulting image intensity field at each time is

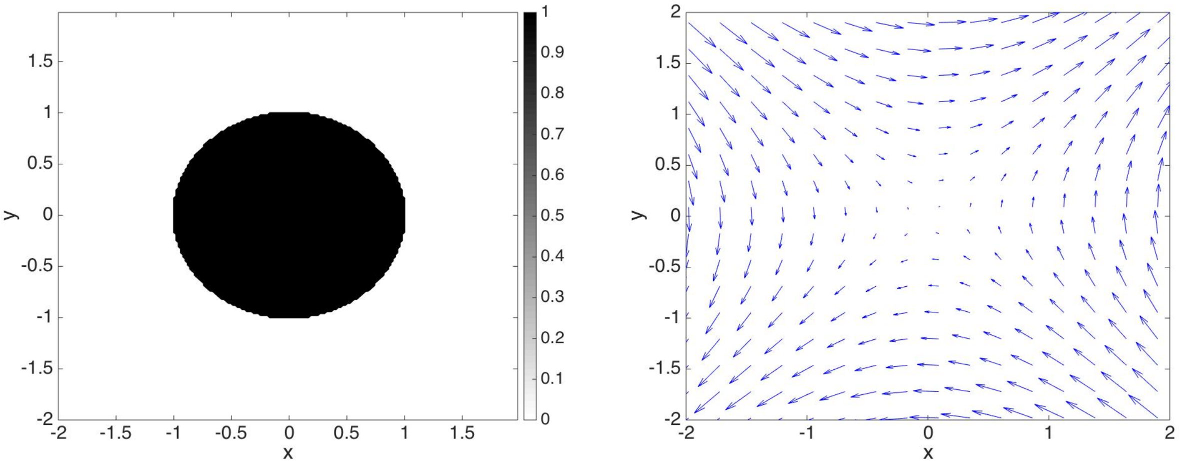
$$I^r(x, t + \mathrm{d}t) = I(x - \mathrm{d}x^r, t). \tag{5.26}$$

In the discrete setting, we perform a linear interpolation of the intensity field onto the pixel locations. This allows for the reconstruction to have a sharp interface, which is crucial for interpretable and accurate multiphase flow prediction.

5.3.3. Numerical and Experimental Results

In this section we present the results of applying the proposed method on a fabricated toy problem, a planar liquid jet undergoing primary atomization and on back-lit images of a liquid jet with a co-axial gas flow. We compare the optical flow based model reduction with Dynamic Mode Decomposition (DMD) for the same input data of volume-of-fluid in the computational domain. Model reduction using Horn-Schunck in conjunction with DMD will be referred to as 'HS-DMD' while we use 'stream-DMD' for streamfunction based optical flow estimation followed by application of DMD.

Toy Problem



(a) initial condition

(b) velocity field at $t = 0.3$

Figure 5.8.: (a) The initial condition for the flow advected by the velocity field corresponding to the streamfunction in Equation (5.27). (b) The velocity field generated by the streamfunction at time $t = 0.3$.

The toy problem represents an oscillating drop and we choose data that is representative of back-lit imaging data that is obtained from experiments. The initial snapshot at $t = 0$ is shown in Figure 5.8 which represents a circular blob of fluid of radius $r = 1$ placed in the center of a

$[-2,2] \times [-2,2]$ domain. The liquid is represented by the dark region and the surrounded white region represents the gas phase. Similar to the volume of fluid formulation, the value $q = 1$ represents the liquid phase and $q = 0$ represents the gas phase. We advect the circular blob of liquid using a velocity field derived from the following streamfuction

$$\phi = (x^2 - y^2)\sin(\omega t) \tag{5.27}$$

where $\omega = 3\pi/2$. The computational domain in discretized into a uniform grid of $[201 \times 201]$ points and constant timestep of $\Delta = 0.1$ is used to advance the flow in time.

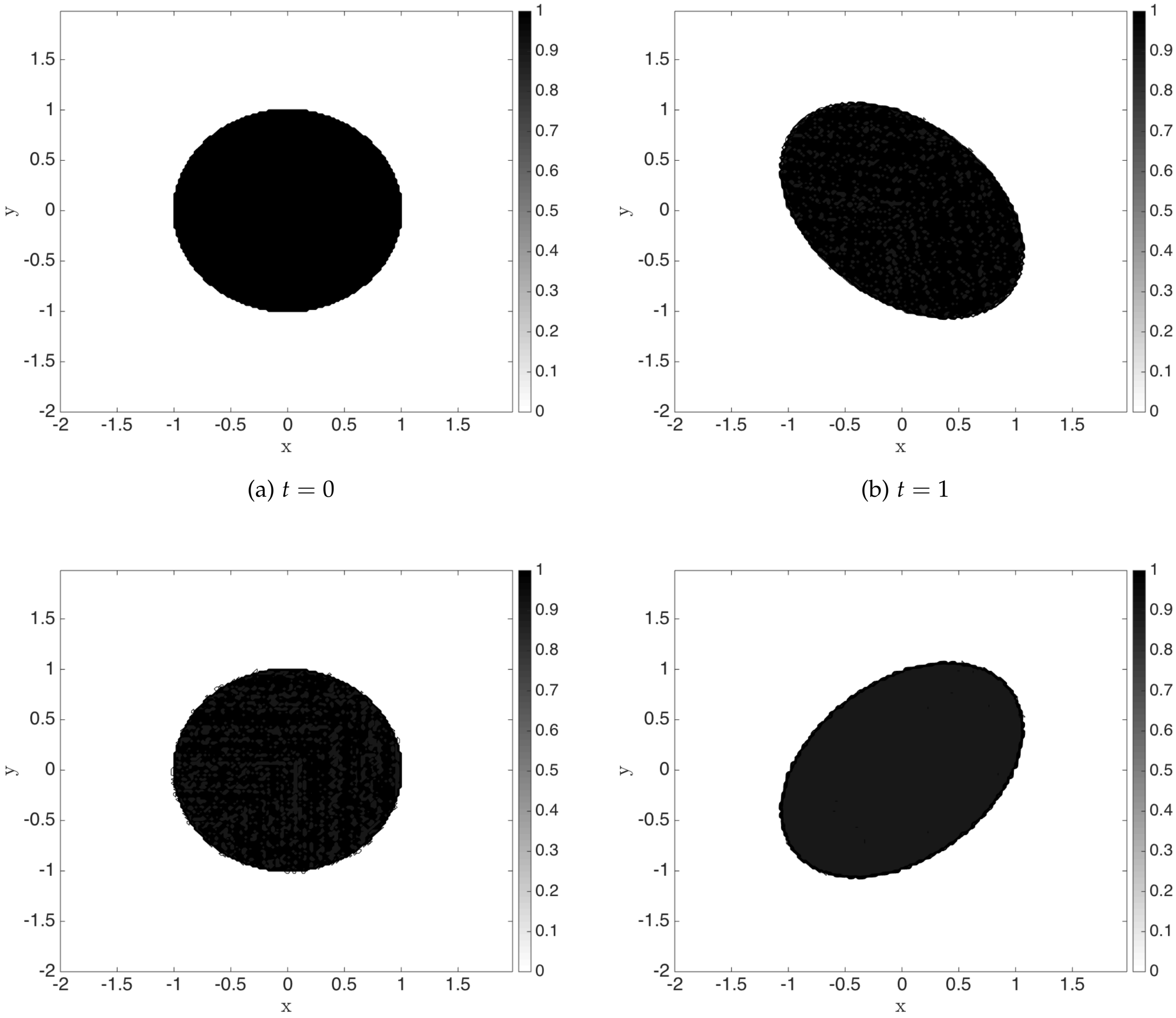
The resulting image snapshots have $[201 \times 201]$ pixels and we collect 41 such snapshots for a total time period of $T = 4$. The spatial and temporal resolution is low enough so that the interface is not completely resolved which is similar to the features that are often observed in real experimental multiphase flow images. Figure 5.9 shows images of the oscillating droplet at 4 different time points. The period of oscillation of the droplet is $2\pi/\omega = 1.33$ and therefore the total time period consisted of 3 periods of oscillations of the droplet.

We compare the reconstruction of HS-DMD and stream-DMD with regular DMD for time $t = 1$ in Figure 5.10. The interface is completely smeared and diffused in the DMD reconstruction in Figure 5.10b. On the other hand, the reconstruction using HS-DMD and stream-DMD both have sharp interfaces close to the true snapshot shown in Figure 5.10a.

Reconstruction and Prediction Error

The quality of a data-driven reduced order model can be judged by the reconstruction and prediction error. Reconstruction error is defined by the difference between the training data used to learn the reduced order model and the reduced order data obtained from the model for the same time points. In this study, we use the relative L_2 error to find the difference between the reduced order and full order data. Prediction error is defined by the relative L_2 error between the reduced order data, from the full order data, generated beyond the time points that were used to learn the reduced order model.

To test the reconstruction and prediction capabilities of the reduced order models, we generate



(a) $t = 0$

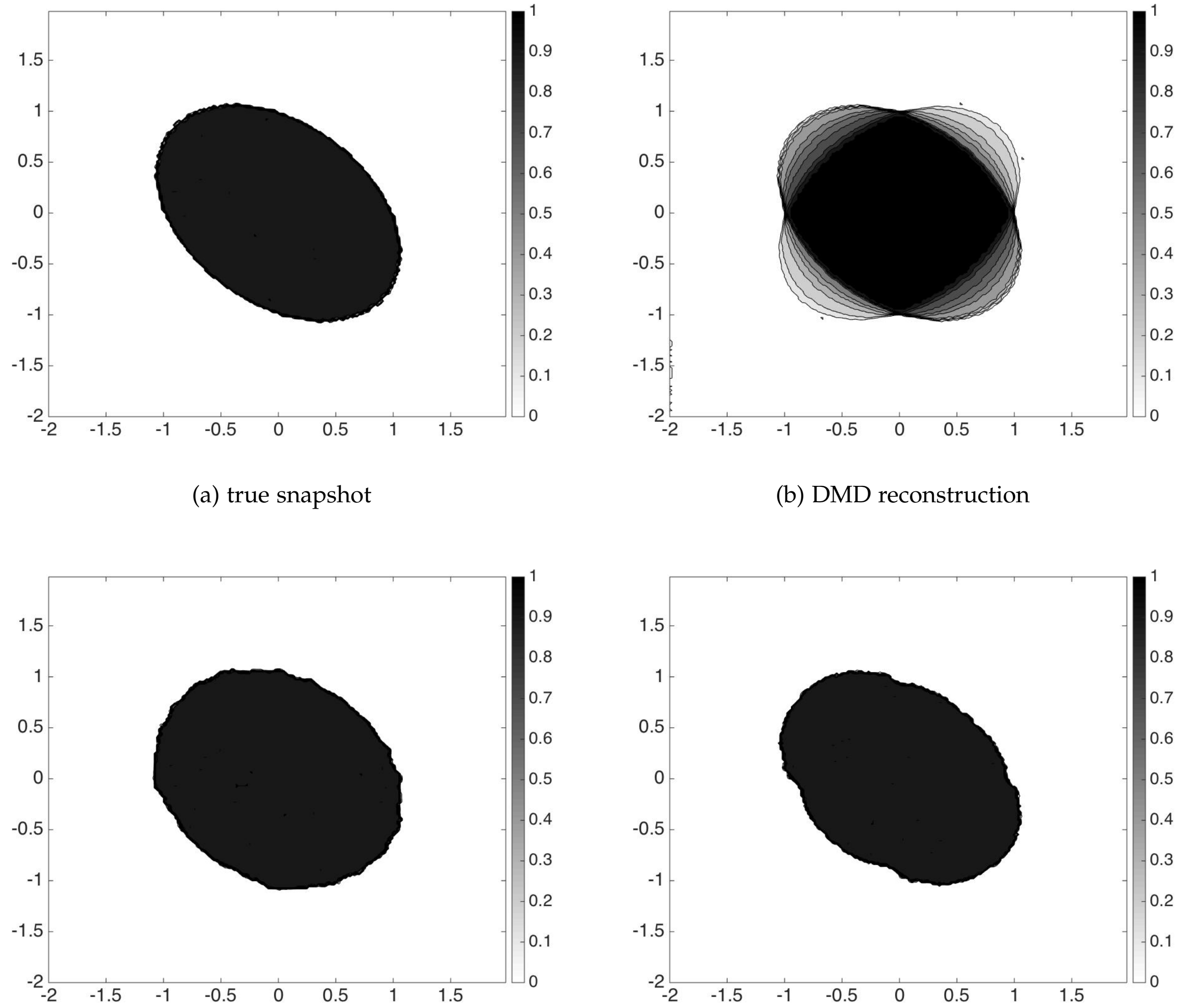
(b) $t = 1$

(c) $t = 2$

(d) $t = 3$

Figure 5.9.: Snapshots of the oscillating liquid drop at different time instances. The period of oscillation is $2\pi/\omega = 1.33$.

20 snapshots of training data for $t \in [0, 2]$. We compute the reconstruction error for the time period $t \in [0, 2]$ and prediction error for $t \in [2, 4]$. Figure 5.11 shows the reconstruction and prediction error as a function of time for DMD, HS-DMD and stream-DMD. Although DMD is able to reconstruct the first few snapshots accurately, it incurs considerably higher errors compared to HS-DMD and stream-DMD during the reconstruction period (i.e., $t \in [0, 2]$). Towards to end of the prediction period for $t \in [3.5, 4]$, we see DMD and HS-DMD showing comparable errors. However, the L_2 error alone is not a good representative of the reduced order model performance, since DMD reconstruction and prediction loses all interpretation since the interface is not sharp and well



(a) true snapshot

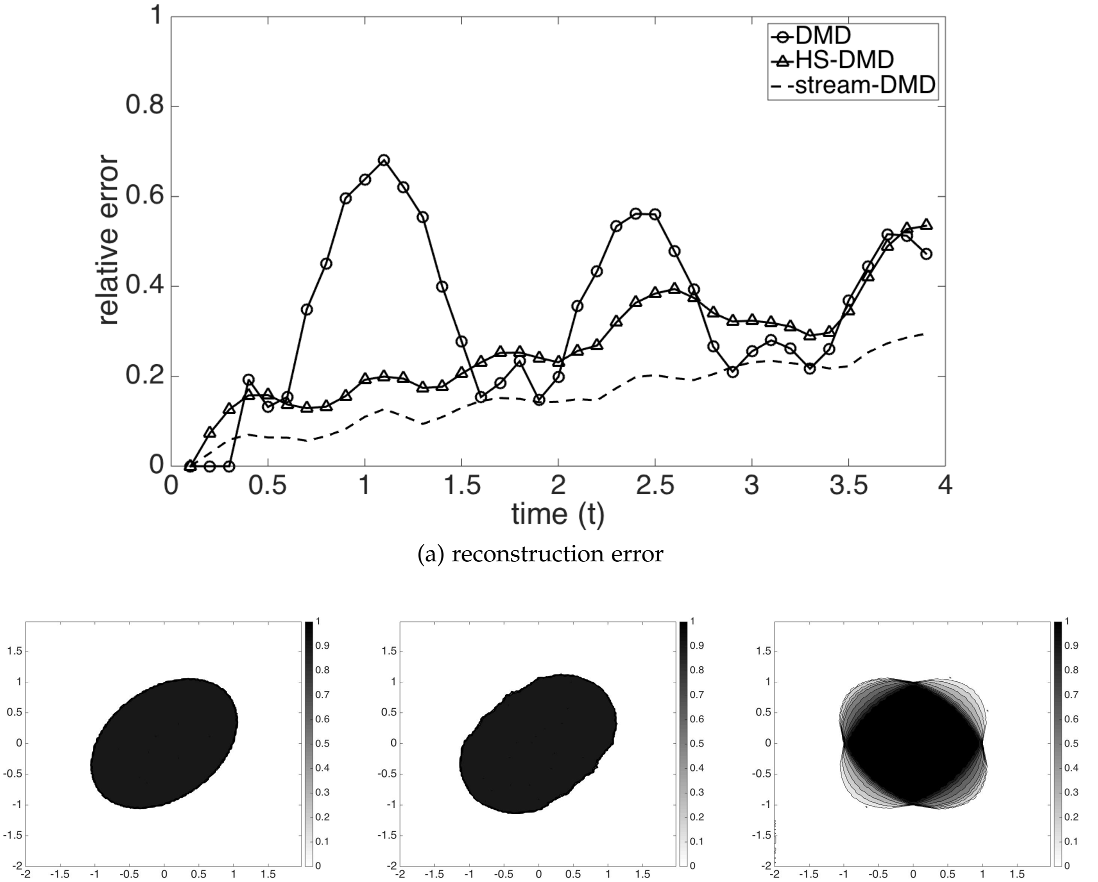
(b) DMD reconstruction

(c) HS-DMD reconstruction

(d) stream-DMD reconstruction

Figure 5.10.: Comparison of (a) the true droplet snapshot at $t = 1$ to the reconstruction generated using a rank 20 approximation generated using (b) DMD, (c) HS-DMD and (d) stream-DMD. The training data consists of snapshots until $t = 2$ for HS-DMD and stream-DMD whereas regular DMD uses all the snapshots from $t \in [0, 4]$.

defined. For instance, even though the relative error is lower for the DMD prediction compared to the HS-DMD prediction at $t = 3$, the HS-DMD prediction (Fig. 5.11c) more closely resembles the true snapshot (Fig. 5.11b) compared to the DMD prediction (Fig. 5.11d). Finally, stream-DMD outperforms both DMD and HS-DMD over almost the entire period in both reconstruction and prediction.



(a) reconstruction error

(b) true snapshot

(c) HS-DMD prediction

(d) DMD prediction

Figure 5.11.: (a) Relative error of reduced order reconstruction ($t \in [0,2]$) and prediction ($t \in [2,4]$) for DMD (○), HS-DMD (△) and stream-DMD (). The training data consists of snapshots until $t = 2$ for all the reduced order models. (b) True snapshot of the oscillating drop at $t = 3$, while (c) and (d) show the HS-DMD and the DMD predictions respectively.

Liquid Volume Conservation

We next look at the conservation of liquid in the computational domain with the reduced order models. We define the *total liquid proportion* as the fraction of area in the entire computational domain that is occupied by the liquid phase. This area can only be defined in the presence of a sharp interface separating the liquid and the gas phases. Therefore, we only include the optical flow based reduced order models in this analysis.

Since the velocity field is incompressible and no liquid phase is entering or leaving the computational domain, the area of liquid in the domain should remain unchanged during the course of the simulation. The total liquid proportion is given by the area of the initial drop divided by the

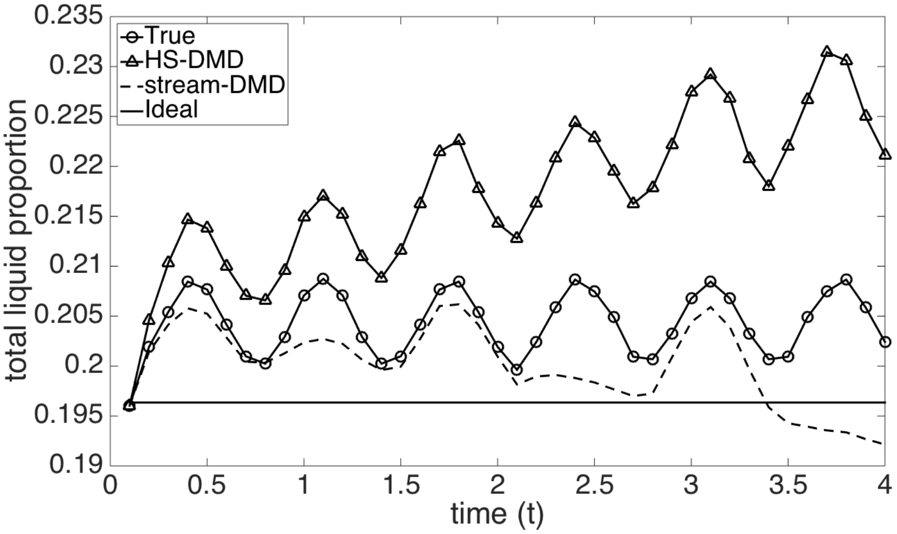


Figure 5.12.: Total liquid volume in the reconstruction ($t \in [0,2]$) and prediction ($t \in [2,4]$) periods of the flow for HS-DMD (△) and stream-DMD (- - - -). The total liquid proportion is shown with (○) and the ideal accurate volume in the continuum limit is shown in solid line (——)

area of the entire computational domain. However due to numerical discretization, even the full order model has some deviation from the full conservation of the total liquid volume.

Figure 5.12 shows the true total liquid volume has an oscillation around a constant mean and the ideal liquid volume proportion in the continuum limit, given by $\pi/16$, is shown by a solid black line. The total liquid proportion for HS-DMD and stream-DMD are also shown in Figure 5.12. HS-DMD reconstruction and prediction has a consistently higher liquid proportion than the true liquid proportion. The total liquid proportion of the flow generated by the HS-DMD model oscillated around a mean that monotonically increases in time. On the other hand, stream-DMD follows the true liquid proportion reasonably well in the reconstruction period of $t \in [0,2]$ but loses liquid volume gradually in the prediction period of $t \in [2,4]$. The superior performance of stream-DMD compared to HS-DMD in total liquid volume conservation can be explained by the volume preserving nature of the divergence-free velocity field generated by the streamfunction formulation. This example demonstrates that physics-based constraints can bring significant improvement in the performance of reduced order models.

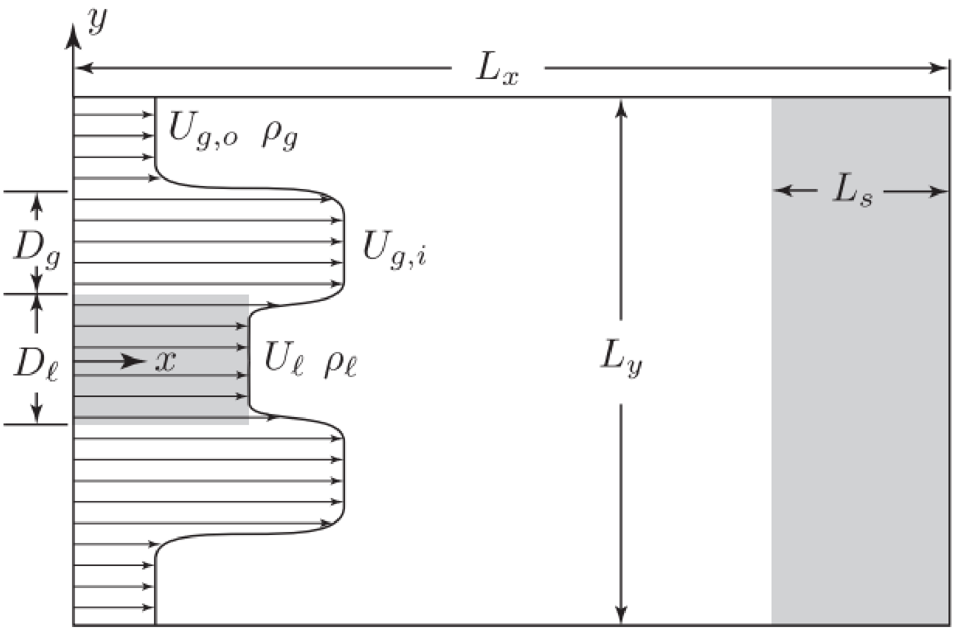


Figure 5.13.: A schematic of the computational domain of a planar liquid jet atomization simulation with relevant scales indicated. The leftmost gray region indicates the liquid column width, whereas the rightmost gray region indicates the sponge region

Numerical Planar Liquid Jet

To demonstrate the model reduction methodology on a more realistic flow, we construct a reduced order model for a planar liquid jet with gas co-flow. The configuration of the flow is schematically shown in Figure 5.13. We use the NGA multiphase flow solver for the simulation ([41]). We impose the velocity profile of the incoming flow in Figure 5.13 using hyperbolic tangent functions such that

$$\begin{aligned} u(0,y,t) =&U_{g,o} + \frac{1}{2}(U_{g,i} - U_{g,o}) \tanh\left(y + \frac{D_\ell}{2} + D_g\right) + \\ &\frac{1}{2}(U_\ell - U_{g,i}) \tanh\left(y + \frac{D_\ell}{2}\right) + \frac{1}{2}(U_{g,i} - U_\ell) \tanh\left(y - \frac{D_\ell}{2}\right) + \\ &\frac{1}{2}(U_{g,o} - U_{g,i}) \tanh\left(y - \frac{D_\ell}{2} - D_g\right). \end{aligned} \tag{5.28}$$

The simulation uses a volume-of-fluid (VOF) formulation [117] for tracking and locating the liquid-gas interface. We restrict ourselves to only the VOF data as input for the model reduction in order to resemble the back-lit images that can be captured for experimental flows. The flow parameters used for the numerical simulation are shown in Table 5.1. All quantities in Table 5.1

Parameter	Description	Value
D_ℓ	Incoming liquid jet diameter	5
D_g	Gas co-flow effective diameter	5
L_x, L_y	Computational domain size	10
L_s	Sponge size	1.5
$U_{g,i}$	Gas co-flow inner velocity	10
$U_{g,o}$	Gas co-flow outer velocity	10
ρ_g	Gas relative density	1.2×10^{-3}
$v_{\max}$	Perturbation velocity amplitude	0.25
σ	Perturbation kernel width	0.5
ω	Perturbation velocity frequency	π

Table 5.1.: Flow parameters used for the numerical simulation of a planar liquid jet with a gas co-flow. All quantities are normalized by the liquid jet diameter D_ℓ, the liquid jet velocity U_ℓ and liquid jet density ρ_ℓ.

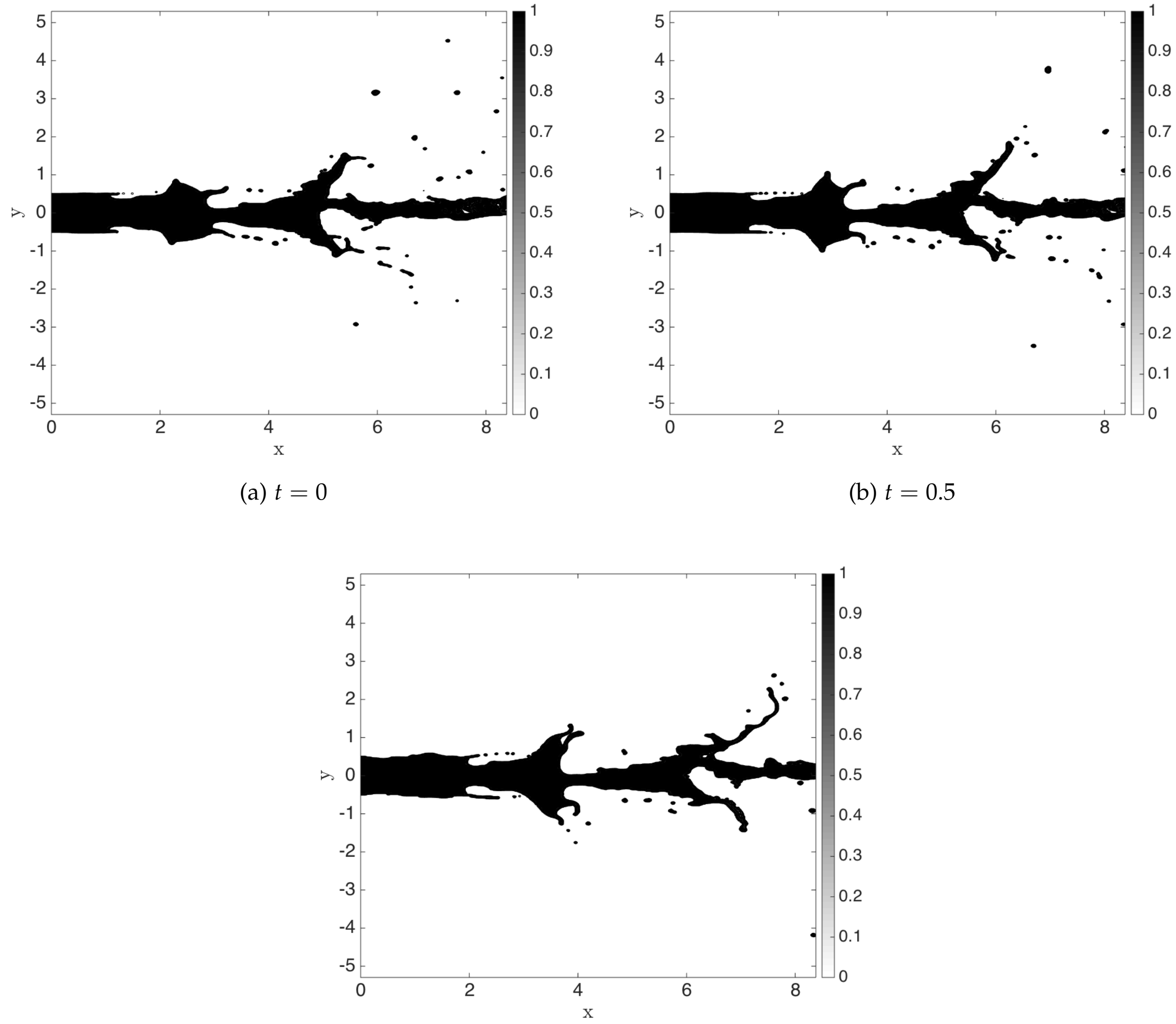
are normalized by the liquid jet diameter D_ℓ, liquid jet velocity U_ℓ and liquid jet density ρ_ℓ. The Weber number for the flow is defined as $We = \rho_g U_g^2 \delta_g / \gamma = 33$, where δ_g is the inflow vorticity thickness and γ is the surface tension coefficient. The gas and liquid phase Reynolds numbers are $Re_g = \rho_g U_g D_g / \mu_g = 3.5 \times 10^6$ and $Re_\ell = \rho_\ell U_\ell D_\ell / \mu_\ell = 1 \times 10^6$ respectively.

In order to reduce the time it takes for the liquid jet to break up and atomize, we modulate the incoming jet with a sinusoidal vertical velocity perturbation. This perturbation is performed at $x = 0$ and localized using Gaussian kernels at $y_1 = D_\ell/2$ and $y_2 = -D_\ell/2$, i.e. at the upper and lower edge of the incoming liquid jet. The imposed vertical perturbation velocity, as a function of x, y and time t, is of the form

$$v(0, y, t) = v_{\max}\left(e^{-(y+y_1)^2/\sigma^2} + e^{-(y+y_2)^2/\sigma^2}\right)\sin(\omega t). \tag{5.29}$$

The perturbation parameters such as $v_{\max}, \sigma$ and ω are given in Table 5.1. Snapshots of the flow are collected at an interval of $0.1 D_\ell / U_\ell$ and sample snapshots are shown in Figure 5.14. The liquid phase is indicated by black pixels while the gas phase is indicated by white pixels.

Similar to the toy problem, we look at the reconstruction and prediction error for the planar jet using DMD, HS-DMD and stream-DMD. Using a training period of only 5 snapshots from $t \in [0, 0.5]$, we will try to predict the flow for the time period $t \in [0, 1]$. We will see that although DMD is able to reconstruct the flow very accurately for $t \in [0, 0.5]$, it looses the prediction capabilities completely for $t \in [0.5, 1]$. Figure 5.15 shows the comparison of the true snapshot



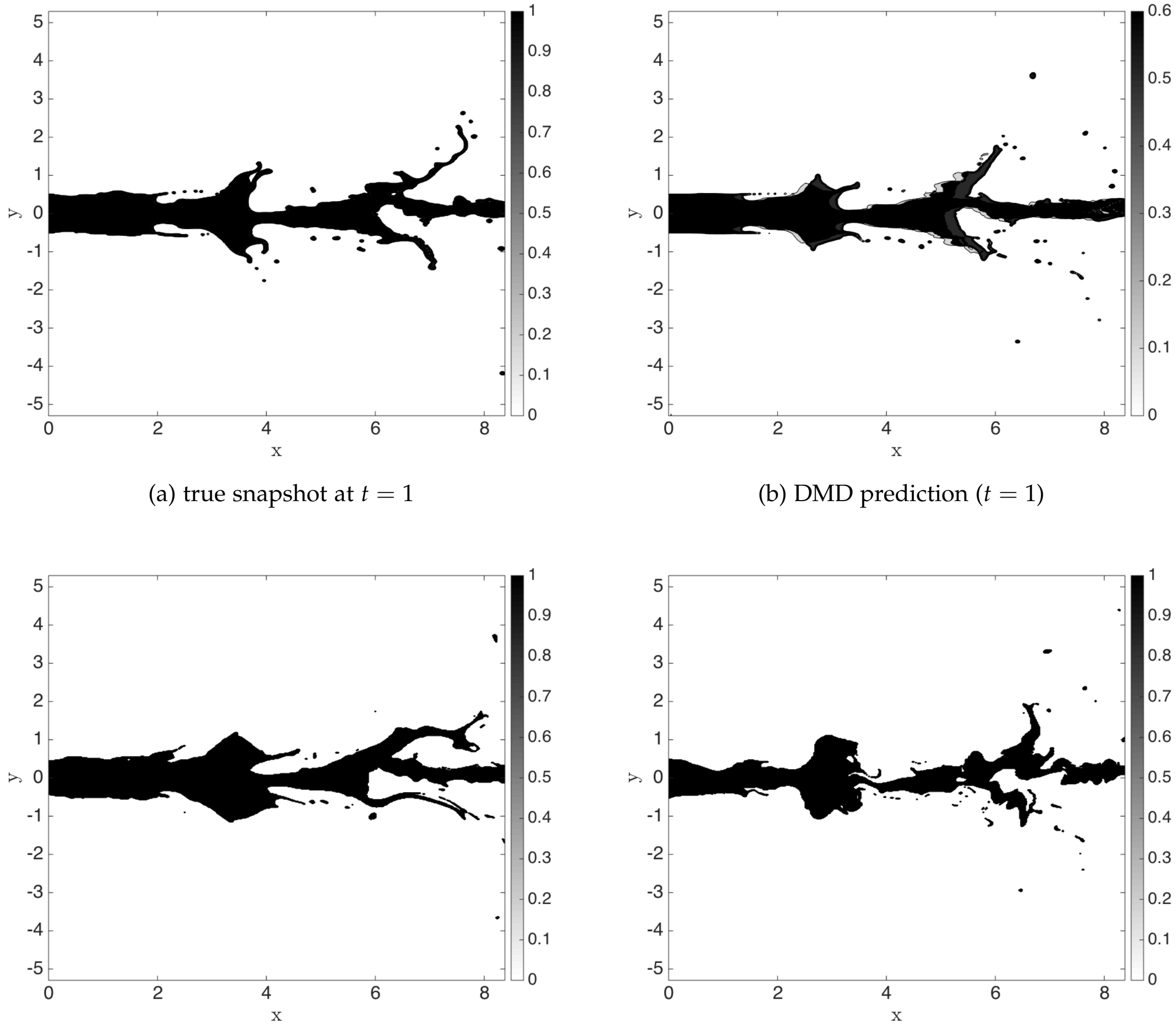
(a) $t = 0$

(b) $t = 0.5$

(c) $t = 1$

Figure 5.14.: Snapshots of volume of fluid of a planar liquid jet in gas co-flow at different time instances.

with reduced order prediction from DMD, HS-DMD and stream-DMD for $t = 1$. The DMD prediction at $t = 1$ in Figure 5.15b is nearly similar to the initial snapshot of the flow at $t = 0$. In fact, all the prediction for $t \in [0.5, 1]$ using DMD results in the same snapshot that looks identical for the flow at $t = 0$. This is because the advection of the VOF field is a highly nonlinear process which is approximated with a linear function in the DMD method. Although it leads to accurate reconstruction when the flow snapshots are in the span of the columns of the data matrix, DMD fails to predict the evolution of the flow in the testing time horizon. On the contrary, optical flow based methods are able to learn the overall motion of the flow and are able to predict the



(a) true snapshot at $t = 1$

(b) DMD prediction ($t = 1$)

(c) HS-DMD prediction ($t = 1$)

(d) stream-DMD prediction ($t = 1$)

Figure 5.15.: Comparison of the (a) true snapshots with the reduced order prediction obtained by (b) DMD, (c) HS-DMD and (d) stream-DMD compared for the numerical planar liquid jet with a gas co-flow.

future flow more reliably, as can be seen in Figure 5.15. HS-DMD can be seen to significantly outperform stream-DMD for the flow prediction at $t = 1$. This can be explained by the incorrect boundary conditions imposed for the streamfunction formulation. We use zero normal velocity as the boundary conditions while in reality the liquid phase enters and leaves the domain from the left and the right boundaries respectively. This discrepancy causes significant accumulation of error for the predictions using the streamfunction formulation. Incorporation of more general boundary conditions for the streamfunction formulation is part of ongoing investigation.

Finally, we look at the reconstruction and prediction error for the reduced order models for for

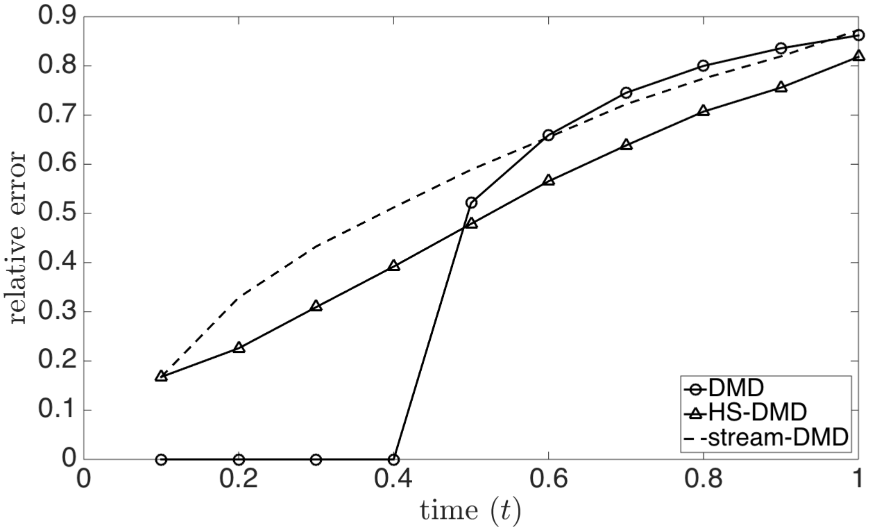


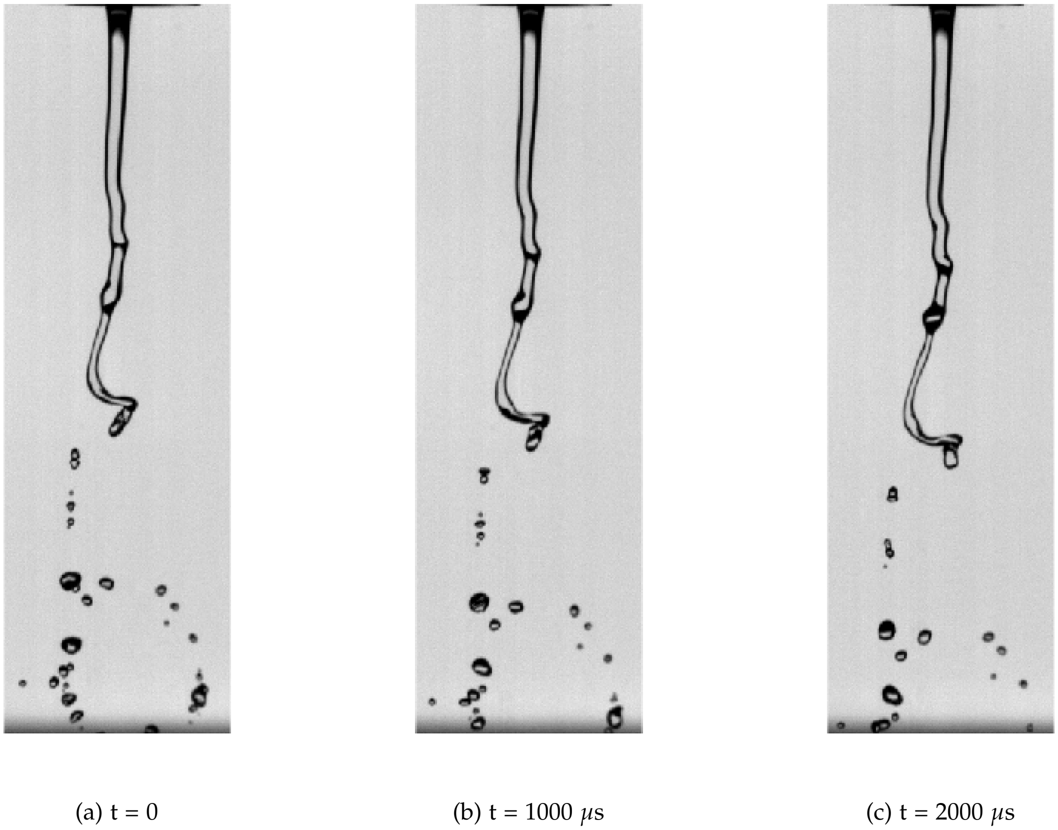
Figure 5.16.: Relative error of reduced order reconstruction ($t \in [0, 0.5]$) and prediction ($t \in [0.5, 1]$) for DMD (○), HS-DMD (△) and stream-DMD (- - - -) for a numerical planar liquid jet. The training data consists of snapshots until $t = 0.5$ for all the reduced order models.

all the reduced order models in Figure 5.16. As expected, even though DMD is able to reconstruct the flow accurately, the relative error shoots up in the prediction region $t \in [0.5, 1]$. Due to the discrepancy in the boundary conditions, we see HS-DMD outperform stream-DMD in both the reconstruction and prediction of the planar liquid jet. HS-DMD outperforms both stream-DMD and DMD in the prediction of the flow for all time points in the prediction region.

Experimental Liquid Jet with Co-axial Gas Atomizer

We now test the interface tracking algorithms on the experimental data described in Section 5.2.1. Since we are using only images from one of the cameras, We consider the flow configuration with no swirl velocity in the gas stream (case1). As such, both liquid and gas flows only have axial velocity as they exit their respective nozzles. Back-lit imaging of the flow is performed using a high-speed camera that captures images of the flow at a frame rate of 5000 frames per second (an image every 200 μs) with a exposure of 99μs. For this study, we set the liquid Reynolds number $Re_\ell = 1100$ and gas Reynolds number $Re_g = 21300$. Figure 5.17 shows the images of the flow at different times with intact liquid core as well as the droplets.

Our goal is to use only five snapshots of the flow, separated by $\Delta t = 200\mu$s, and predict the flow at five more time-steps in the future, so we have the flow images from $t = [0, 1000\mu\text{s}]$ and



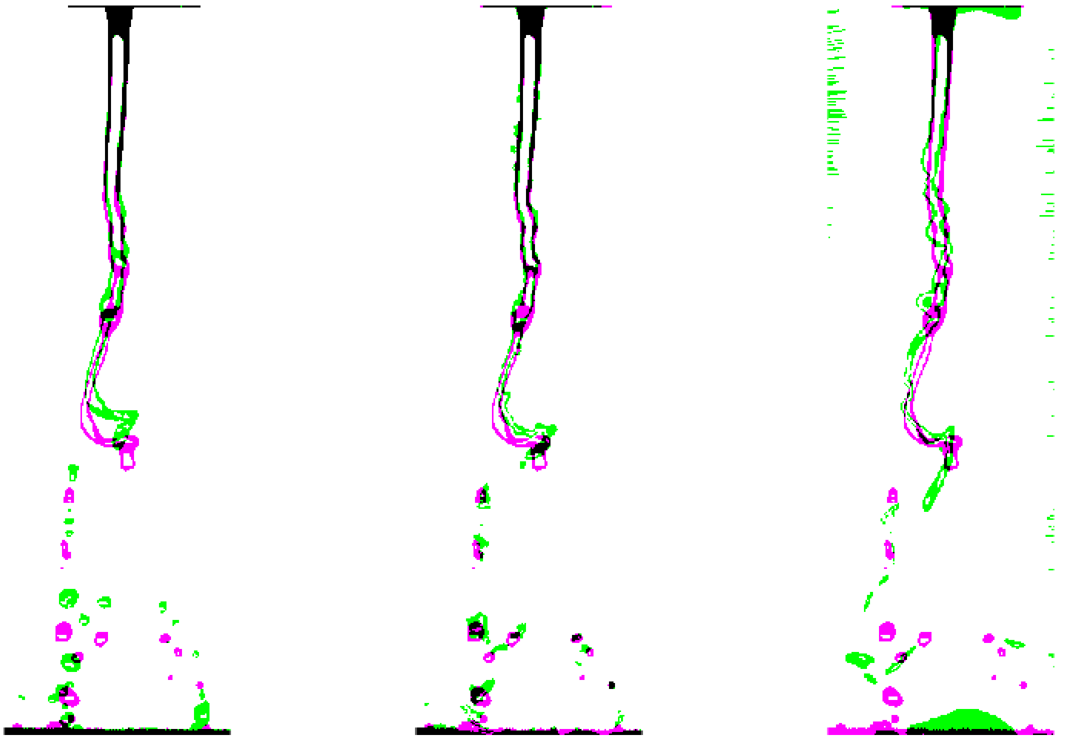
(a) t = 0

(b) t = 1000 μs

(c) t = 2000 μs

Figure 5.17.: Back-lit images of the experimental liquid jet with co-axial gas atomizer collected every 200 μs.

we need to predict the flow at $t = 2000\mu$s. We employ the same procedure as described in the Section on the numerical jet. Figure 5.18 shows the comparison of the ground-truth with the prediction using the reduced order models. For ease of visualization, we only show the interface locations for the ground-truth and the predictions. The green contours represent the ground-truth, magenta contours are the predicted interface locations and the black contours are regions where the prediction matches the ground-truth exactly. Just like the numerical jet case, the DMD results look similar to the flow at $t = 0$. HS-DMD and stream-DMD on the other hand advect the jet core as well as the droplets downstream. The stream-DMD method gets high errors at the boundaries of the image. This is due to the boundary conditions not matching the ground truth. Although the streamfunction formulation shows good results for problems when the boundary conditions are known, in practice it has significant errors at the boundaries. A fruitful direction for future work is to incorporate boundary effects in the streamfunction formulation to reduce or even remove the error at the boundaries. Figure 5.19 shows the reconstruction and prediction error for the



(a) DMD prediction (b) HS-DMD prediction (c) stream-DMD prediction

Figure 5.18.: Comparison of the true snapshot of the experimental liquid jet with co-axial gas atomizer with prediction obtained by (a) DMD, (b) HS-DMD and (c) stream-DMD. Only the interface is visualized for clear comparison. The green contours represent the ground-truth, the magenta contours are the prediction and black contours are the regions where the true snapshot and the prediction coincide.

reduced order models. Similar to the numerical jet, DMD is able to reconstruct the flow but has significant relative errors in the prediction region $t \in [1000\mu\text{s}, 2000\mu\text{s}]$. The stream-DMD method has the worst performance due to the high errors at the boundaries, while HS-DMD outperforms both stream-DMD and DMD in the prediction of the flow.

5.4. Conclusion

In this study, we perform data-driven modal analysis and interface tracking of multiphase flows. Specifically, we use Dynamic Mode Decomposition (DMD) [132] to analyze the flow of liquid core within a swirling co-axial airblast atomizer. We perform back-lit imaging of the flow with two synchronized high-speed cameras from orthogonal directions. Our study is composed of two

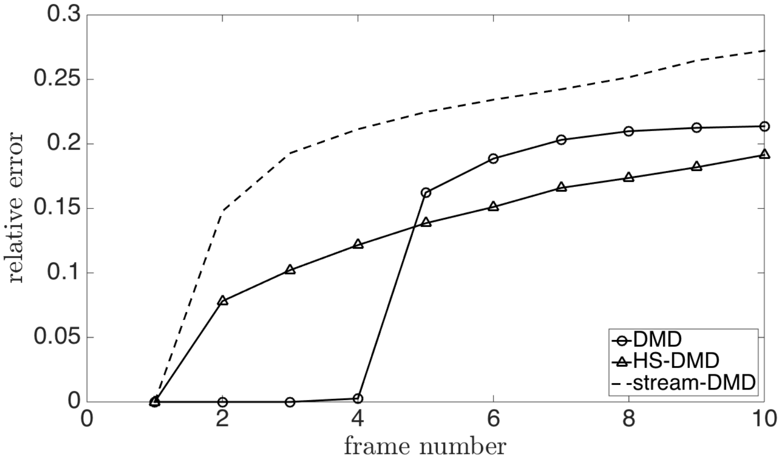


Figure 5.19.: Relative error of reduced order reconstruction ($t \in [0, 1000\mu s]$) and prediction ($t \in [1000\mu s, 2000\mu s]$) for DMD (○), HS-DMD (△) and stream-DMD (- - - -) for an experimental liquid jet with co-axial gas atomizer. The training data consists of snapshots until $t = 1000\mu s$ separated with $\Delta t = 200\mu s$ for all the reduced order models.

parts. In the first part, we perform data-driven modal analysis of the flow to extract large-scale instabilities of the liquid jet. The stereographic imaging of the flow using two camera allows to capture the three-dimensional nature of the instabilities. While this analysis provides valuable information about the spatio-temporal structure of flow instabilities, they are not sufficient to resolve the location and shape of the interface. In the second part of the study, we propose a two-step method of interface reconstruction and tracking using data-driven reduced order modeling. We use optical flow methods to estimate the flow, which is used to construct a reduced order model. This model is then used to track and predict the position of the liquid-gas interface in the flow. The method is demonstrated on a fabricated toy problem, primary atomization of a planar liquid jet and an experimental liquid jet with a co-axial gas atomizer. The reduced order model is shown to preserve the sharpness of the interface and reliably reconstruct and predict the interface location which is critical for understanding and controlling multiphase flows.

There are several avenues for future research. The current flow estimation methods do not leverage synchronized images from multiple images. Extending the proposed method to integrate images from multiple cameras will lead to more accurate flow estimation and, as a result, better interface reconstruction and prediction. Another important application of reduced order modeling of multiphase flows is for real-time control applications. Accurate prediction and reconstruction of the flow will enable real-time image-based feedback control of multiphase flows [23, 115].

CHAPTER 6

DISCUSSION AND FUTURE DIRECTIONS

In this dissertation, we introduce novel data-driven methods of reduced-order modeling, control and sensing of fluid flows. First, we introduce low-rank Dynamic Mode Decomposition (lrDMD), which is a data-driven non-intrusive reduced-order modeling method that approximates the dynamics of the system using a linear map between different subspaces. This is a generalization of the approach of existing methods that fit an endomorphic linear function to model the flow dynamics. We show that the flexibility of using different input and output subspaces leads to the design of better feedback controllers. The reduced-order model is constructed by solving a rank-constrained matrix optimization problem. On the algorithmic side, we developed two methods to solve the problem. The first method, is a computationally efficient subspace projection method that provides an approximate solution. The second method is a gradient-based method that is guaranteed to converge to a local minimum. We provide guidelines for the usage of these methods to construct the reduced-order model. On the application side, we demonstrate the superior performance of our method, compared to existing methods, on stabilization and estimation of optimal actuator location for the linearized Ginzburg-Landau equations in the supercritical regime. We also show that, compared to existing methods, feedback control using our method leads to faster and more efficient suppression of vortex shedding in the wake of an inclined flat plate in uniform flow.

Second, we developed a sensor placement framework for fluid flows, which can work in conjunction with any reduced-order modeling technique that provides a linear description of the flow dynamics. We use the Kalman filter equations to provide closed-form expressions of the estimate of the state vector for a given placement of sensors. An adjoint-based framework is used to find the optimal sensor locations. We propose different objective functions for the prediction and control applications. For flow reconstruction and prediction, we minimize the trace of an empirical approximation of the estimation error covariance matrix. We demonstrate this approach with accurate reconstruction and prediction of solutions of the complex linearized Ginzburg-Landau equations in the globally unstable regime. For control applications, we propose using weighted estimation error, where the weights are determined by the control gain matrix used in the feedback controller. We show that our method places sensors enabling better estimates, compared to existing methods, for flow reconstruction and prediction applications. Using this approach, we also place sensors in the wake of over an inclined flat plate to yield the near-optimal observer-based feedback control performance.

Third, we demonstrate that current methods for data-driven modeling of fluid flows are not suitable for reconstruction and prediction of the interface location in multiphase flows. To that end, we develop a novel data-driven two-step approach for reduced order modeling of multiphase flows using imaging data. Our method uses Dynamic Mode Decomposition (DMD) on the optical flow field estimated from consecutive image snapshots. We demonstrate the performance of our method on a real experiment of liquid jet surrounded by a coaxial airblast atomizer that is visualized using back-lit imaging. Our method is able to accurately reconstruct and predict the flow while preserving the sharpness of the liquid-gas interface.

6.1. Future Work

There are several avenues for future work. Here we describe two research directions in detail. First, we formulate a way to incorporate the effect of control in the low-rank Dynamic Mode Decomposition (lrDMD) framework. Second, we propose a new method for stochastic flow control using hidden Markov models.

6.1.1. lrDMD with Control

The lrDMD framework can easily be extended to incorporate the effect of control. Consider a dynamical system with state vector $\mathbf{q} \in \mathbb{R}^m$ such that

$$\mathbf{q}_{k+1} = f(\mathbf{q}_k, \mathbf{u}_k),$$

where $\mathbf{u} \in \mathbb{R}^p$ is the control input and the subscripts denote the time iteration. We have access to a sequence of time snapshots of the state vector which we represent as a data matrices $X, Y \in \mathbb{R}^{m\times n}$ formed by n pairs of data snapshots as follows,

$$X := [\mathbf{q}_0 | \dots | \mathbf{q}_{n-1}], \quad Y := [\mathbf{q}_1 | \dots | \mathbf{q}_n].$$

We additionally have access to a sequence of control inputs which we arrange in the following matrix

$$\Omega := [\mathbf{u}_0 | \dots | \mathbf{u}_{n-1}].$$

Our goal is to use matrices X, Y and Ω to obtain a low-order approximation of the function $f(\cdot)$. To this end, we construct a linear approximation of f in both the state $\mathbf{q}$ and $\mathbf{u}$ such that

$$f(\mathbf{q}, \mathbf{u}) = \hat{A}\mathbf{x} + \hat{B}\mathbf{u}.$$

Along with a rank-constraint on the system dynamical matrix, we get the following optimization problem

$$\min_{L,D,R,B} \left\| Y - LDR^T X - LB\Omega \right\|_F^2, \tag{6.1}$$

where $D \in \mathbb{R}^{r\times r}$, $B \in \mathbb{R}^{r\times p}$, $L, R \in \mathbb{R}^{m\times r}$ and $L^T L = R^T R = I_r$ (where I_r is the $r \times r$ identity matrix) such that $\hat{A} = LDR^T$ is the r-ranked matrix approximating the state dynamics and $\hat{B} = LB$ accounts for the effect of control input on the state evolution.

For a fixed L and R we observe that the objective function is convex for both D and B. The optimality conditions for the solution of D and B for fixed L and R are

$$D(R^T X X^T R) + B(\Omega X^T R) = L^T Y X^T R,$$
$$B(\Omega\Omega^T) + D(R^T X \Omega^T) = L^T Y \Omega^T,$$

which can be rewritten as

$$\begin{bmatrix} D & B \end{bmatrix} \begin{bmatrix} R^T X X^T R & \Omega X^T R \\ R^T X \Omega^T & \Omega \Omega^T \end{bmatrix} = \begin{bmatrix} L^T Y X^T R & L^T Y \Omega^T \end{bmatrix} \tag{6.2}$$

We propose two ways to solve this problem. Both methods work with an initial guess for the optimal (L, R) which can be obtained from leading singular vectors of the data matrices as shown in Section 3.3. The first method is alternating minimization by solving for optimal (D, B) for fixed (L, R) by solving Eq. (6.2) followed by gradient based minimization of the objective function (6.1) for fixed (D, B). The second method is to use Schur complements to get closed form expressions for the optimal $(D^*(L, R), B^*(L, R))$, substitute that in the objective function 6.1 and use gradient based methods to find on optimal (L, R) solution.

6.1.2. Stochastic Flow Control using Hidden Markov Models

The data-driven methods proposed in this thesis construct finite dimensional approximation of the Koopman operator. Alternatively, there has also been work on using the dual of the Koopman operator, called the Perron-Frobenius operator, for flow modeling and control. While Koopman operator acts on the space of functions of the state vector, the Perron-Frobenius operator describes the evolution of measures (or probability densities). A classical approach to approximate the Perron-Frobenius operator is Ulam's method [40], in which the state space is partitioned into finite number of disjoint sets and the transition probabilities between these sets is numerically estimated. The resulting state transition matrix is shown to approximate the Perron-Frobenius operator. More recent data-driven methods of approximating the Perron-Frobenious operator draw inspiration from dynamic mode decomposition and exploit the duality between the Koopman and the Perron-Frobenius operator [51, 72].

In this subsection, we propose using input-output hidden Markov models, i.e. partially observed Markov models parameterized by control input, to model the dynamics of the system and formulate a control framework that drives the system to the desired asymptotic behaviour. The proposed method can be viewed as a generalization of cluster-based reduced-order models (CROM) [81] which approximates the Perron-Frobenius operator and has been used to reduce the size of recirculation area of a separating flow over a smoothly contoured ramp. We pose the control problem as a combinatorial optimization problem and demonstrate the performance of our method by driving the asymptotic probability distribution of the Lorenz system to a desired distribution.

Preliminaries

Consider a finite state dynamical system with the following state-space description

$$q_k = f(q_{k-1}, u_{k-1}),$$
$$y_k = g(q_k),$$

where u_k is the input, y_k is the output and q_k is the state variable at time iteration k. Let $\mathcal{Q}$ be the set of feasible states with $|\mathcal{Q}| = m$, Σ be the set of possible outputs with $|\Sigma| = p$ and $\Omega = [\omega_i]$ be the set of possible inputs with $|\Omega| = d$.

We look at the probabilistic version of this system where the current input and state distribution is used to estimate the state distribution and output distribution at the next time step. Since $f(\cdot)$ is a function of both q and u, we consider d state transition matrices $\mathbf{A} = [A_1, \ldots, A_d]$ where the entries of the transition matrix $A_i = [a_{s,t,i}]$ denote the probability of transition from state s to state t under the action of control input $i \in \Omega$. We define the distribution of the initial condition q_0 with $P(q_0 = s) = p_{0,s}$. The output function $g(\cdot)$ is modeled by the emission probability matrix $E = [e_{s,i}]$ whose entries $e_{s,i}$ indicate the probability of emission of output i from state s. This is known as an input-output hidden Markov model (I/O HMM).

Figure 6.1 shows the schematic of the I/O HMM architecture with the Bayesian dependence of the variables indicated by arrows. The top layer contains the input variables u; the middle layer contains the hidden state variables q; and the bottom layer contains the observed output variables y. The total number of unknown parameters including all the matrices is $((m-1)^d + (m-1)p + m - 1)$.

We have the following objectives.

- **Model Learning** We need a way to learn the I/O HMM model parameters $\theta = (\mathbf{A}, E, p_0)$ from a set of observation $y^n := \{y_0, \ldots, y_n\}$ and control inputs $u^{n-1} = \{u_0, \ldots, u_{n-1}\}$.

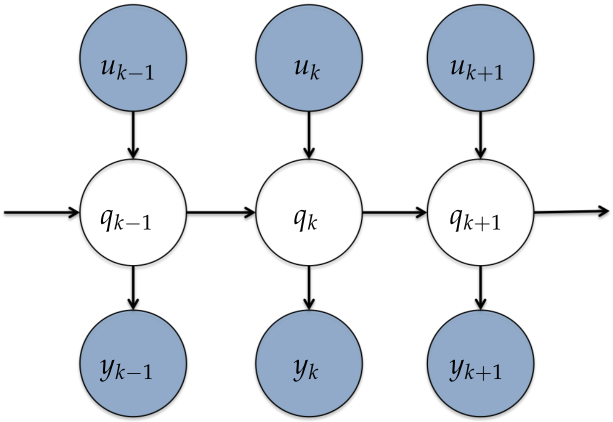


Figure 6.1.: Schematic of the I/O HMM architecture. The shaded nodes are observed information while white nodes are hidden random variables.

- **Control Framework** We need a control law to perform stochastic control of the flow using the I/O HMM model learned from data.

Model Learning

An important property of the Markov model is that given state q_k, states q_{k+1} and q_{k-1} are independent of each other, i.e.

$$q_{k+1} \perp q_{k-1} \,|\, q_k.$$

This is known as the Markovian property of the model. This property allows for the use of dynamic programming for efficient computation of posterior probability for a given set of observations y^n and control inputs u^{n-1}. We use the Baum-Welch algorithm [16] find the maximum likelihood estimate of model parameters $\theta = (\mathbf{A}, E, p_0)$ by maximizing

$$P(y^n|\theta, u^{n-1}) = \sum_{q^n} P(q^n, y^n|\theta, u^{n-1}).$$

Control Framework

In several engineering applications, control of asymptotic behavior is of interest. For ergodic systems, the infinite horizon average can be represented in terms of the spatial average weighted by the asymptotic probability density function. In this study we construct feedback controllers that manipulate the asymptotic probability density function $p^\infty \in \mathbb{R}^m$. The goal is to construct a controller that minimizes the expected cost of the asymptotic behavior of the system given by

$$E(J) = \sum_{s \in \mathcal{Q}} c(s) \cdot p^\infty_s$$

where $c(s)$ is the cost associated with the hidden state $s \in \mathcal{Q}$. The cost is given by

$$c(s) = \sum_{k \in \Sigma} E_{s,k} \tilde{c}(k)$$

where $\tilde{c}$ is the cost associated with the observation $k \in \Sigma$.

Remark 6.1.1. We can also drive the asymptotic probability distribution to a desired distribution by minimizing the following quadratic cost function

$$\sum_{s \in \mathcal{Q}} (p^d_s - p^\infty_s)^2$$

where $p^d \in \mathbb{R}^m$ is the desired distribution.

Belief Propagation

We design the control law such that at each time step, the controller chooses a control input $u \in \Omega$ based on the hidden state of the system at that time step. To do this at any given time step i we define a new variable called belief $b \in [0,1]^m$, where

$$b_s = P(q_i = s | y^i, u^{i-1}, \theta).$$

Clearly we have that $\sum\limits_{s \in \mathcal{Q}} b_s = 1$ and the belief coincides with p_0 for the initial condition. If the belief from the previous iteration $i-1$ is denoted as $\tilde{b} \in \mathbb{R}^m$, u_{i-1} is the control input at the previous iteration and y_i is the observation of the current iteration, the updated belief $b \in \mathbb{R}^m$ is given by

$$b_s = E_{s,y_i} \sum_{t \in \mathcal{Q}} A_{t,s,u_{i-1}} \tilde{b}_t.$$

We pick the hidden state with the highest belief value at each iteration for choose the control input based on the control law.

Control Law

The control law $\Lambda : \mathcal{Q} \to \Omega$ picks a control input depending on the hidden state with the higher belief at each iteration. We define a decision variable $z \in \{0,1\}^{m \times d}$ where $z_{s,k} = 1$ if and only if $\Lambda(q_s) = \omega_k$. The state transition matrix, denoted by $\widehat{A}$, of the system controlled using the control law Λ can be build as follows,

$$\widehat{A}_{s,t} = A_{s,t,\Lambda(s)}.$$

We also introduce a variable $p \in [0,1]^m$ which is the asymptotic probability distribution of the system. We require the following constraints,

$$\begin{aligned} \sum_{s=1}^{m} p_s &= 1 \\ \widehat{A} p &= p. \end{aligned}$$

Under the above constraints, we minimize the cost function

$$E(\Lambda) = \sum_{s \in \mathcal{Q}} c(s) \cdot p_s,$$

using combinatorial optimization techniques. We use Gurobi [114], which is an efficient solver for mixed linear integer programs, to perform the optimization.

Remark 6.1.2. An alternate approach would be a stochastic controller that chooses the control input based only the belief with probability given by

$$P(u_i = \Lambda(s)) = b_s$$

where b is the belief at time step i and $s \in \mathcal{Q}$.

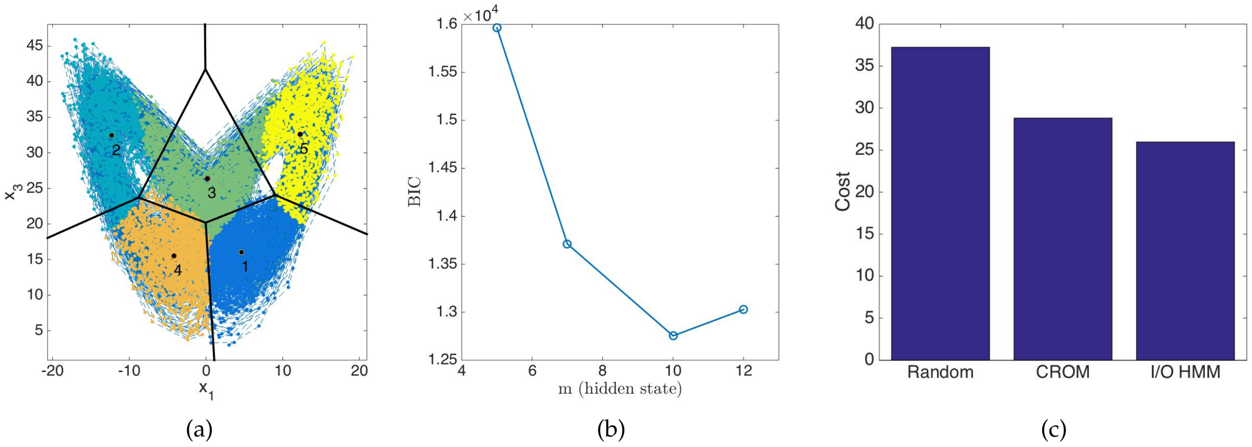


Figure 6.2.: Feedback stochastic control of the Lorenz system using data-driven reduced-order modeling. (a) Data-driven discretization of the state space of the Lorenz system into 5 clusters differentiated by color. Black dots represent the cluster centroids. (b) Bayesian Information Criteria (BIC) for increasing number of hidden states for the Lorenz system. (c) Comparison of asymptotic average cost using a random controller, CROM based controller and I/O HMM based controller.

Results

We employ I/O HMM control framework to control the transitions and manipulate the stationary distribution of the Lorenz system. Consider the Lorenz system with the following state-space description,

$$\frac{dx}{dt} = \sigma(y-x) + f(x,y,z,t)$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z$$

where $\sigma = 10, \rho = 28$, $\beta = 8/3$ and $f(x,y,z)$ is the forcing term. This system is nonlinear, chaotic and for $f = 0$ it has three unstable fixed points at $(0,0,0)$ and $(\pm\sqrt{72}, \pm\sqrt{72}, 27)$. We generate training data from this system with $\delta t = 0.1$ for a time period $T = 100$ under uniform random forcing with $\Omega = \{-10, 0, 10\}$. The state space is discretized into $p = 5$ clusters using K-means algorithm [8] as shown in Figure 6.2a. Clusters 2 (light blue) and 4 (orange) reside in the left wing of the Lorenz system while clusters 1 (blue) and 5 (yellow) belong to the right wing. Cluster 3 (green) acts as a transition cluster between the two wings. This training data is then used to generate parameterized CROM and I/O HMM with increasing number of hidden states m.

We use the Bayesian Information Criteria (BIC) [136] for selecting the I/O HMM model for given observation sequence. BIC is given by

$$BIC = \ln(2n)((m-1)^d + (m-1)p + m - 1) - 2\ln\left(\sum_{s\in\mathcal{Q}} f[s,n]\right).$$

The model with the lowest BIC is selected. Figure 6.2b shows I/O HMM model with $m = 10$ has the lowest BIC score. The cost function for the optimization at any cluster is defined as the distance of the cluster centroid for the centroid of cluster 2. This is intended to drive the system to towards the left wing of the Lorenz system. Figure 6.2c shows comparison of cost incurred by a random controller, CROM based controller and I/O HMM controller with $m = 10$. It is clear that I/O HMM based controller outperforms CROM in this case.

Outlook

The results shown above on the Lorenz system serve as a proof-of-concept for the proposed approach of stochastic control of fluid flows. In future work, this method can be applied to more realistic fluid flow data, either from numerical simulations or real experiments. Moreover, the current framework can be extended to allow adaptive online learning or reinforcement learning, in which the reduced-order model is updated as new observations are gathered during the flow control application. Similar machine learning techniques have been successfully applied for active flow control [22, 46, 122] and hold the potential to enable robust flow control strategies for several real-life applications.

REFERENCES

[1] Absil, P.-A., Baker, C. G., and Gallivan, K. A. (2007). Trust-region methods on riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330.

[2] Absil, P.-A., Mahony, R., and Sepulchre, R. (2009). *Optimization algorithms on matrix manifolds*. Princeton University Press.

[3] Ahuja, S. (2009). *Reduction methods for feedback stabilization of fluid flows*. PhD thesis, PhD thesis, Princeton University, New Jersey.

[4] Ahuja, S. and Rowley, C. W. (2010). Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators. *Journal of Fluid Mechanics*, 645:447–478.

[5] Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(1):243–247.

[6] Akervik, E., Hœpffner, J., Ehrenstein, U., and Henningson, D. S. (2007). Optimal growth, model reduction and control in a separated boundary-layer flow using global eigenmodes. *Journal of Fluid Mechanics*, 579:305.

[7] Alexanderian, A., Petra, N., Stadler, G., and Ghattas, O. (2014). A-optimal design of experiments for infinite-dimensional bayesian linear inverse problems with regularized \ell_0-sparsification. *SIAM Journal on Scientific Computing*, 36(5):A2122–A2148.

[8] Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

[9] Ashgriz, N. (2011). *Handbook of atomization and sprays: theory and applications*. Springer Science & Business Media.

[10] Athans, M. (1971). The role and use of the stochastic linear-quadratic-gaussian problem in control system design. *IEEE Transactions on Automatic Control*, 16(6):529–552.

[11] Ayyappan, D. and Vaidyanathan, A. (2020). Study on instability of circular liquid jets at subcritical to supercritical conditions using dynamic mode decomposition. *Physics of Fluids*, 32(1):014107.

[12] Bagheri, S., Brandt, L., and Henningson, D. S. (2009a). Input-output analysis, model reduction and control of the flat-plate boundary layer. *Journal of Fluid Mechanics*, 620(2):263–298.

[13] Bagheri, S., Henningson, D., Hoepffner, J., and Schmid, P. (2009b). Input-output analysis and control design applied to a linear model of spatially developing flows. *Applied Mechanics Reviews*, 62(2):020803.

[14] Barraud, A. (1977). A numerical algorithm to solve $A^T XA - X = Q$. *IEEE Transactions on Automatic Control*, 22(5):883–885.

[15] Bastani, K., Kong, Z., Huang, W., and Zhou, Y. (2016). Compressive sensing–based optimal sensor placement and fault diagnosis for multi-station assembly processes. *IIE Transactions*, 48(5):462–474.

[16] Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171.

[17] Bergmann, R. (2021). Manopt.jl. https://manoptjl.org.

[18] Berrueta, T. A., Abraham, I., and Murphey, T. (2020). Experimental applications of the Koopman operator in active learning for control. In *The Koopman Operator in Systems and Control*, pages 421–450. Springer.

[19] Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334.

[20] Bhattacharjee, D., Hemati, M., Klose, B., and Jacobs, G. (2018). Optimal actuator selection for airfoil separation control. In *2018 Flow Control Conference*, page 3692.

[21] Bhattacharjee, D., Klose, B., Jacobs, G. B., and Hemati, M. S. (2020). Data-driven selection of actuators for optimal control of airfoil separation. *Theoretical and Computational Fluid Dynamics*, 34(4):557–575.

[22] Bieker, K., Peitz, S., Brunton, S. L., Kutz, J. N., and Dellnitz, M. (2019). Deep model predictive control with online learning for complex physical systems. *arXiv preprint arXiv:1905.10094*.

[23] Bothell, J. K., Morgan, T. B., and Heindel, T. J. (2020). Image-based feedback control for a coaxial spray. *Journal of Fluids Engineering*, 142(11).

[24] Boumal, N., Mishra, B., Absil, P.-A., Sepulchre, R., et al. (2014). Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15(1):1455–1459.

[25] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*. SIAM.

[26] Bui-Thanh, T., Damodaran, M., and Willcox, K. (2004). Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516.

[27] Carlberg, K., Bou-Mosleh, C., and Farhat, C. (2011). Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181.

[28] Carleman, T. et al. (1932). Application of the theory of linear integral equations to systems of equations non-linear différentials. *Acta Mathematica*, 59:63–87.

[29] Castello, C. C., Fan, J., Davari, A., and Chen, R.-X. (2010). Optimal sensor placement strategy for environmental monitoring using wireless sensor networks. In *2010 42nd Southeastern Symposium on System Theory (SSST)*, pages 275–279. IEEE.

[30] Charalampous, G., Hadjiyiannis, C., and Hardalupas, Y. (2019). Proper orthogonal decomposition of primary breakup and spray in co-axial airblast atomizers. *Physics of Fluids*, 31(4):043304.

[31] Chatterjee, A. (2000). An introduction to the proper orthogonal decomposition. *Current Science*, pages 808–817.

[32] Chaturantabut, S. and Sorensen, D. C. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764.

[33] Chen, K. K. and Rowley, C. W. (2011). H2 optimal actuator and sensor placement in the linearised complex ginzburg–landau system. *Journal of Fluid Mechanics*, 681:241–260.

[34] Chen, K. K., Tu, J. H., and Rowley, C. W. (2012). Variants of dynamic mode decomposition: boundary condition, Koopman, and fourier analyses. *Journal of Nonlinear Science*, 22(6):887–915.

[35] Colonius, T. and Taira, K. (2008). A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197(25-28):2131–2146.

[36] Cuzol, A., Hellier, P., and Mémin, E. (2007). A low dimensional fluid motion estimator. *International Journal of Computer Vision*, 75(3):329–349.

[37] Dawson, S. T., Hemati, M. S., Williams, M. O., and Rowley, C. W. (2016). Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):42.

[38] Deem, E. A., Cattafesta, L. N., Yao, H., Hemati, M., Zhang, H., and Rowley, C. W. (2018). Experimental implementation of modal approaches for autonomous reattachment of separated flows. In *2018 AIAA Aerospace Sciences Meeting*, page 1052.

[39] Deem, E. A., III, L. N. C., Hemati, M. S., Zhang, H., Rowley, C., and Mittal, R. (2020). Adaptive separation control of a laminar boundary layer using online dynamic mode decomposition. *Journal of Fluid Mechanics*, 903:A21.

[40] Dellnitz, M. and Junge, O. (1999). On the approximation of complicated dynamical behavior. *SIAM Journal on Numerical Analysis*, 36(2):491–515.

[41] Desjardins, O., Moureau, V., and Pitsch, H. (2008). An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395–8416.

[42] Drmac, Z. and Gugercin, S. (2016). A new selection operator for the discrete empirical interpolation method – improved a priori error bound and extensions. *SIAM Journal on Scientific Computing*, 38(2):A631–A648.

[43] Du, R., Gkatzikis, L., Fischione, C., and Xiao, M. (2015). Energy efficient sensor activation for water distribution networks based on compressive sensing. *IEEE Journal on Selected Areas in Communications*, 33(12):2997–3010.

[44] Edelman, A., Arias, T. A., and Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353.

[45] Fabian, V. and Hannan, J. (1977). On the cramér-rao inequality. *The Annals of Statistics*, 5(1):197–205.

[46] Fan, D., Yang, L., Wang, Z., Triantafyllou, M. S., and Karniadakis, G. E. (2020). Reinforcement learning for bluff body active flow control in experiments and simulations. *Proceedings of the National Academy of Sciences*, 117(42):26091–26098.

[47] Flinois, T. L. and Morgans, A. S. (2016). Feedback control of unstable flows: a direct modelling approach using the eigensystem realisation algorithm. *Journal of Fluid Mechanics*, 793:41–78.

[48] Folkestad, C., Pastor, D., Mezic, I., Mohr, R., Fonoberova, M., and Burdick, J. (2020). Extended dynamic mode decomposition with learned Koopman eigenfunctions for prediction and control. In *2020 American Control Conference (acc)*, pages 3906–3913. IEEE.

[49] Fontanini, A. D., Vaidya, U., and Ganapathysubramanian, B. (2016). A methodology for optimal placement of sensors in enclosed environments: A dynamical systems approach. *Building and Environment*, 100:145–161.

[50] Giannetti, F. and Luchini, P. (2007). Structural sensitivity of the first instability of the cylinder wake. *Journal of Fluid Mechanics*, 581(1):167–197.

[51] Goswami, D., Thackray, E., and Paley, D. A. (2018). Constrained Ulam dynamic mode decomposition: Approximation of the Perron-Frobenius operator for deterministic and stochastic systems. *IEEE Control Systems Letters*, 2(4):809–814.

[52] Goulart, P. J., Wynn, A., and Pearson, D. (2012). Optimal mode decomposition for high dimensional systems. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 4965–4970. IEEE.

[53] Gugercin, S. and Antoulas, A. C. (2004). A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766.

[54] Gunder, T., Sehlinger, A., Skoda, R., and Mönnigmann, M. (2018). Sensor placement for reduced-order model-based observers in hydraulic fluid machinery. *IFAC-PapersOnLine*, 51(13):414–419.

[55] Hay, K., Liu, Z.-C., and Hanratty, T. (1998). A backlighted imaging technique for particle size measurements in two-phase flows. *Experiments in Fluids*, 25(3):226–232.

[56] He, X., Hu, W., and Zhang, Y. (2018). Observer-based feedback boundary stabilization of the Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 339:542–566.

[57] Héas, P. and Herzet, C. (2016). Low-rank approximation and dynamic mode decomposition. *arXiv preprint arXiv:1610.02962*.

[58] Héas, P., Mémin, E., Papadakis, N., and Szantai, A. (2007). Layered estimation of atmospheric mesoscale dynamics from satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4087–4104.

[59] Heindel, T. J. (2011). A review of x-ray flow visualization with applications to multiphase flows. *Journal of Fluids Engineering*, 133(7):074001.

[60] Heindel, T. J., Gray, J. N., and Jensen, T. C. (2008). An X-ray system for visualizing fluid flows. *Flow Measurement and Instrumentation*, 19(2):67–78.

[61] Heitz, D., Mémin, E., and Schnörr, C. (2010). Variational fluid flow measurements from image sequences: synopsis and perspectives. *Experiments in Fluids*, 48(3):369–393.

[62] Hemati, M., Deem, E., Williams, M., Rowley, C. W., and Cattafesta, L. N. (2016). Improving separation control with noise-robust variants of dynamic mode decomposition. In *54th AIAA Aerospace Sciences Meeting*, page 1103.

[63] Hemati, M. S., Rowley, C. W., Deem, E. A., and Cattafesta, L. N. (2017). De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets. *Theoretical and Computational Fluid Dynamics*, 31(4):349–368.

[64] Hemati, M. S., Williams, M. O., and Rowley, C. W. (2014). Dynamic mode decomposition for large and streaming datasets. *Physics of Fluids*, 26(11):111701.

[65] Hervé, A., Sipp, D., Schmid, P. J., and Samuelides, M. (2012). A physics-based approach to flow control using system identification. *Journal of Fluid Mechanics*, 702:26–58.

[66] Hewitt, G. and Whalley, P. (1980). Advanced optical instrumentation methods. *International Journal of Multiphase Flow*, 6(1-2):139–156.

[67] Högberg, M., Bewley, T. R., and Henningson, D. S. (2003). Linear feedback control and estimation of transition in plane channel flow. *Journal of Fluid Mechanics*, 481:149.

[68] Honkanen, M., Eloranta, H., and Saarenrinne, P. (2010). Digital imaging measurement of dense multiphase flows in industrial processes. *Flow Measurement and Instrumentation*, 21(1):25–32.

[69] Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, 17(1-3):185–203.

[70] Hua, J.-C., Gunaratne, G. H., Talley, D. G., Gord, J. R., and Roy, S. (2016). Dynamic-mode decomposition based analysis of shear coaxial jets with and without transverse acoustic driving. *Journal of Fluid Mechanics*, 790:5–32.

[71] Huang, B., Ma, X., and Vaidya, U. (2020). Data-driven nonlinear stabilization using Koopman operator. In *The Koopman Operator in Systems and Control*, pages 313–334. Springer.

[72] Huang, B. and Vaidya, U. (2018). Data-driven approximation of transfer operators: Naturally structured dynamic mode decomposition. In *2018 Annual American Control Conference (ACC)*, pages 5659–5664. IEEE.

[73] Huang, S.-C. and Kim, J. (2008). Control and system identification of a separated flow. *Physics of Fluids*, 20(10):101509.

[74] Illingworth, S. J. (2016). Model-based control of vortex shedding at low Reynolds numbers. *Theoretical and Computational Fluid Dynamics*, 30(5):429–448.

[75] Iñigo, J. G., Sipp, D., and Schmid, P. J. (2014). A dynamic observer to capture and control perturbation energy in noise amplifiers. *Journal of Fluid Mechanics*, 758:728–753.

[76] Joshi, S. and Boyd, S. (2008). Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462.

[77] Jovanović, M. R., Schmid, P. J., and Nichols, J. W. (2014). Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26(2):024103.

[78] Juang, J.-N. (1994). *Applied system identification*. Prentice-Hall, Inc.

[79] Juang, J.-N. and Pappa, R. S. (1985). An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics*, 8(5):620–627.

[80] Kaiser, E., Kutz, J. N., and Brunton, S. L. (2020). Data-driven approximations of dynamical systems operators for control. In *The Koopman Operator in Systems and Control*, pages 197–234. Springer.

[81] Kaiser, E., Noack, B. R., Spohn, A., Cattafesta, L. N., and Morzyński, M. (2017). Cluster-based control of a separating flow over a smoothly contoured ramp. *Theoretical and Computational Fluid Dynamics*, 31(5):579–593.

[82] Kim, J. (2003). Control of turbulent boundary layers. *Physics of Fluids*, 15(5):1093–1105.

[83] Kim, J. and Bewley, T. R. (2007). A linear systems approach to flow control. *Annual Review of Fluid Mechanics*, 39:383–417.

[84] Kim, J., Bodony, D. J., and Freund, J. B. (2014). Adjoint-based control of loud events in a turbulent jet. *Journal of Fluid Mechanics*, 741:28–59.

[85] Klus, S., Nüske, F., Peitz, S., Niemann, J.-H., Clementi, C., and Schütte, C. (2020). Data-driven approximation of the Koopman generator: Model reduction, system identification, and control. *Physica D: Nonlinear Phenomena*, 406:132416.

[86] Koopman, B. and Neumann, J. v. (1932). Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3):255.

[87] Koopman, B. O. (1931). Hamiltonian systems and transformation in Hilbert space. *Proceedings of the national academy of sciences of the united states of america*, 17(5):315.

[88] Korda, M. and Mezić, I. (2018). Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160.

[89] Korda, M. and Mezić, I. (2020). Koopman model predictive control of nonlinear dynamical systems. In *The Koopman Operator in Systems and Control*, pages 235–255. Springer.

[90] Kramer, B., Peherstorfer, B., and Willcox, K. (2017). Feedback control for systems with uncertain parameters using online-adaptive reduced models. *SIAM Journal on Applied Dynamical Systems*, 16(3):1563–1586.

[91] Kumar, A. and Sahu, S. (2018). Liquid jet breakup unsteadiness in a coaxial air-blast atomizer. *International Journal of Spray and Combustion Dynamics*, 10(3):211–230.

[92] Kumar, A. and Sahu, S. (2019). Large scale instabilities in coaxial air-water jets with annular air swirl. *Physics of Fluids*, 31(12):124103.

[93] Kwakernaak, H. and Sivan, R. (1972). *Linear optimal control systems*, volume 1. Wiley-Interscience New York.

[94] Lefebvre, A. (1985). Fuel effects on gas turbine combustion—ignition, stability, and combustion efficiency. *Journal of Engineering for Gas Turbines and Power*, 107(1):24–37.

[95] Li, D., Bothell, J., Morgan, T., Heindel, T., Aliseda, A., Machicoane, N., and Kastengren, A. (2018). Quantitative analysis of an airblast atomizer in the near-field region using broadband and narrowband x-ray measurements. In *ICLASS 2018, 14th Triennial International Conference on Liquid Atomization and Spray Systems*.

[96] Liu, T., Merat, A., Makhmalbaf, M., Fajardo, C., and Merati, P. (2015). Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images. *Experiments in Fluids*, 56(8):166.

[97] Liu, T. and Shen, L. (2008). Fluid flow and optical flow. *Journal of Fluid Mechanics*, 614:253–291.

[98] Luchini, P. and Bottaro, A. (2014). Adjoint equations in stability analysis. *Annual Review of Fluid Mechanics*, 46.

[99] Lumley, J. L. (2007). *Stochastic tools in turbulence*. Courier Corporation.

[100] Luttman, A., Bollt, E., Basnayake, R., and Kramer, S. (2011). A stream function approach to optical flow with applications to fluid transport dynamics. *PAMM*, 11(1):855–856.

[101] Luttman, A., Bollt, E. M., Basnayake, R., Kramer, S., and Tufillaro, N. B. (2013). A framework for estimating potential fluid flow from digital imagery. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(3):033134.

[102] Maas, H., Gruen, A., and Papantoniou, D. (1993). Particle tracking velocimetry in three-dimensional flows. *Experiments in Fluids*, 15(2):133–146.

[103] Machicoane, N. and Aliseda, A. (2017). Experimental characterization of a canonical coaxial gas-liquid atomizer. In *ILASS-Americas 2017: 29th Annual Conference on Liquid Atomization and Spray Systems*, pages 15–18.

[104] Malik, N., Dracos, T., and Papantoniou, D. (1993). Particle tracking velocimetry in three-dimensional flows. *Experiments in Fluids*, 15(4-5):279–294.

[105] Manohar, K., Brunton, B. W., Kutz, J. N., and Brunton, S. L. (2018a). Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine*, 38(3):63–86.

[106] Manohar, K., Kutz, J. N., and Brunton, S. L. (2018b). Optimal sensor and actuator selection using balanced model reduction. *arXiv preprint arXiv:1812.01574*.

[107] Mauroy, A. and Mezić, I. (2016). Global stability analysis using the eigenfunctions of the Koopman operator. *IEEE Transactions on Automatic Control*, 61(11):3356–3369.

[108] Mezić, I. (2005). Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325.

[109] Mezić, I. (2013). Analysis of fluid flows via spectral properties of the Koopman operator. *Annual Review of Fluid Mechanics*, 45:357–378.

[110] Morgan, T. B., Sashittal, P., Heindel, T. J., and Bodony, D. J. (2020). A stereo back-illuminated imaging system for spray characterization. *ILASS Americas*.

[111] Natarajan, M., Freund, J. B., and Bodony, D. J. (2016). Actuator selection and placement for localized feedback flow control. *Journal of Fluid Mechanics*, 809:775–792.

[112] Natarajan, M., Freund, J. B., and Bodony, D. J. (2018). Global mode-based control of laminar and turbulent high-speed jets. *Comptes Rendus Mécanique*, 346(10):978–996.

[113] Oehler, S. F. and Illingworth, S. J. (2018). Sensor and actuator placement trade-offs for a linear model of spatially developing flows. *Journal of Fluid Mechanics*, 854:34–55.

[114] Optimization, G. (2014). Inc.,"Gurobi optimizer reference manual," 2015.

[115] Osuna-Orozco, R., Machicoane, N., Huck, P. D., and Aliseda, A. (2020). Feedback control of the spray liquid distribution of electrostatically assisted coaxial atomization. *Atomization and Sprays*, 30(1).

[116] Otto, S. E. and Rowley, C. W. (2021). Koopman operators for estimation and control of dynamical systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:59–87.

[117] Owkes, M. and Desjardins, O. (2014). A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (VOF) method. *Journal of Computational Physics*, 270:587–612.

[118] Peitz, S. and Klus, S. (2020). Feedback control of nonlinear PDEs using data-efficient reduced order models based on the Koopman operator. In *The Koopman Operator in Systems and Control*, pages 257–282. Springer.

[119] Proctor, J. L., Brunton, S. L., and Kutz, J. N. (2016). Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161.

[120] Proctor, J. L., Brunton, S. L., and Kutz, J. N. (2018). Generalizing Koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930.

[121] Quénot, G. M., Pakleza, J., and Kowalewski, T. A. (1998). Particle image velocimetry with optical flow. *Experiments in Fluids*, 25(3):177–189.

[122] Rabault, J., Ren, F., Zhang, W., Tang, H., and Xu, H. (2020). Deep reinforcement learning in fluid mechanics: A promising method for both active flow control and shape optimization. *Journal of Hydrodynamics*, 32:234–246.

[123] Raffel, M., Willert, C. E., Scarano, F., Kähler, C. J., Wereley, S. T., and Kompenhans, J. (2018). *Particle image velocimetry: a practical guide*. Springer.

[124] Rowley, C. W. (2005). Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013.

[125] Rowley, C. W. and Dawson, S. T. (2017). Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics*, 49:387–417.

[126] Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P., and Henningson, D. S. (2009). Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127.

[127] Saad, Y. and Schultz, M. H. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869.

[128] Salmasi, A., Shadaram, A., and Taleghani, A. S. (2013). Effect of plasma actuator placement on the airfoil efficiency at poststall angles of attack. *IEEE Transactions on Plasma Science*, 41(10):3079–3085.

[129] Sashittal, P. and Bodony, D. J. (2018). Low-rank dynamic mode decomposition using riemannian manifold optimization. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2265–2270. IEEE.

[130] Sashittal, P. and Bodony, D. J. (2019). Reduced-order control using low-rank dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 33(6):603–623.

[131] Sashittal, P., Sayadi, T., Schmid, P., Jang, I., and Magri, L. (2016). Adjoint-based sensitivity analysis for a reacting jet in crossflow. *Center for Turbulence Research, Summer Program, Stanford, CA, June*, pages 375–384.

[132] Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28.

[133] Schmid, P. J. and Brandt, L. (2014). Analysis of fluid systems: Stability, receptivity, sensitivity lecture notes from the flow – Nordita summer school on advanced instability methods for complex flows, stockholm, sweden, 2013. *Applied Mechanics Reviews*, 66(2):024803.

[134] Schmid, P. J. and Henningson, D. S. (2012). *Stability and transition in shear flows*, volume 142. Springer Science & Business Media.

[135] Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

[136] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

[137] Semeraro, O., Bagheri, S., Brandt, L., and Henningson, D. S. (2011). Feedback control of three-dimensional optimal disturbances using reduced-order models. *Journal of Fluid Mechanics*, 677:63–102.

[138] Semeraro, O., Bagheri, S., Brandt, L., and Henningson, D. S. (2013). Transition delay in a boundary layer flow using active control. *Journal of Fluid Mechanics*, 731:288–311.

[139] Surana, A. (2016). Koopman operator based observer synthesis for control-affine nonlinear systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6492–6499. IEEE.

[140] Surana, A. (2020). Koopman framework for nonlinear estimation. In *The Koopman Operator in Systems and Control*, pages 59–79. Springer.

[141] Surana, A. and Banaszuk, A. (2016). Linear observer synthesis for nonlinear systems using Koopman operator framework. *IFAC-PapersOnLine*, 49(18):716–723.

[142] Susuki, Y. and Mezic, I. (2012). Nonlinear Koopman modes and a precursor to power system swing instabilities. *IEEE Transactions on Power Systems*, 27(3):1182–1191.

[143] Townsend, J., Koep, N., and Weichwald, S. (2016). Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *arXiv preprint arXiv:1603.03236*.

[144] Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., and Kutz, J. N. (2013). On dynamic mode decomposition: theory and applications. *arXiv preprint arXiv:1312.0041*.

[145] Van Overschee, P. and De Moor, B. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93.

[146] Van Overschee, P. and De Moor, B. (2012). *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media.

[147] Vicario, F. (2014). *OKID as a general approach to linear and bilinear system identification*. PhD thesis, Columbia University.

[148] Welch, G., Bishop, G., et al. (1995). An introduction to the Kalman filter.

[149] Westerweel, J. (1997). Fundamentals of digital particle image velocimetry. *Measurement Science and Technology*, 8(12):1379.

[150] Willcox, K. (2006). Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & Fluids*, 35(2):208–226.

[151] Willcox, K. and Peraire, J. (2002). Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330.

[152] Williams, M. O., Kevrekidis, I. G., and Rowley, C. W. (2015). A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346.

[153] Witsenhausen, H. S. (1971). Separation of estimation and control for discrete time systems. *Proceedings of the IEEE*, 59(11):1557–1566.

[154] Wynn, A., Pearson, D., Ganapathisubramani, B., and Goulart, P. J. (2013). Optimal mode decomposition for unsteady flows. *Journal of Fluid Mechanics*, 733:473–503.

[155] Yuan, J., Schnörr, C., and Mémin, E. (2007). Discrete orthogonal decomposition and variational fluid flow estimation. *Journal of Mathematical Imaging and Vision*, 28(1):67–80.

APPENDIX A

SUPPLEMENTARY RESULTS

A.1. Comparison of lrDMD with ERA

ERA [79] is a system identification method proposed for linear systems. Consider a linear system with state variable $\mathbf{q} \in \mathbb{R}^m$ and control input $\mathbf{u} \in \mathbb{R}^p$ governed by the equation

$$\mathbf{q}_{k+1} = A\mathbf{q}_k + B\mathbf{u}_k$$

where subscripts denote the time iteration, $A \in \mathbb{R}^{m\times m}$ is the state transition matrix and $B \in \mathbb{R}^{m\times p}$ captures the effect of control on the state variable. The data matrix $X \in \mathbb{R}^{m\times p(m_c+1)}$ using impulse response of this system will be

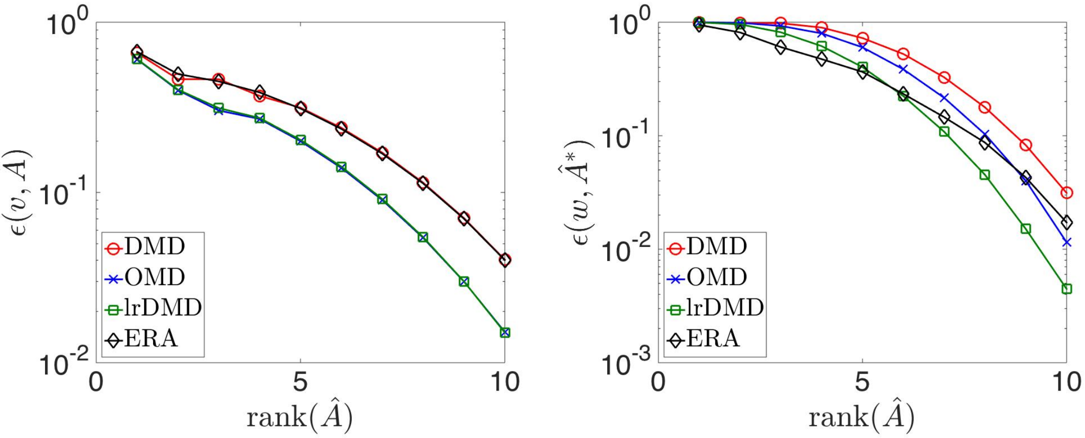
$$X = \begin{bmatrix} B & AB & \cdots & A^{m_c}B \end{bmatrix}$$

where $m_c + 1$ is the number of snapshots.

The first step of the ERA method is to form Hankel matrices from impulse response data of the system. We construct a generalized Hankel matrix $H \in \mathbb{R}^{m(m_o+1)\times p(m_c+1)}$

$$H = \begin{bmatrix} B & AB & \cdots & A^{m_c}B \\ AB & A^2B & \cdots & A^{m_c+1}B \\ \vdots & \vdots & \ddots & \vdots \\ A^{m_o}B & A^{m_o+1}B & \cdots & A^{m_o+m_c}B \end{bmatrix}$$

where m_c and m_o are chosen such that $m_c + m_o \leq m$. We then compute the singular value



(a)

(b)

Figure A.1.: (a) Projection error for the unstable eigenmode and (b)Projection error for the unstable adjoint mode for DMD (red, circle), OMD (blue, cross), lrDMD (green, square) and ERA (black, diamond) for different rank approximations.

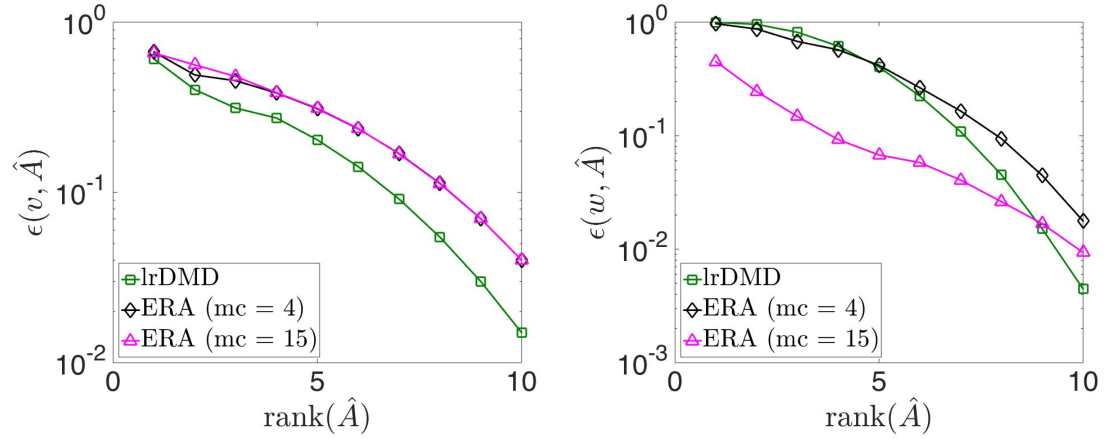
decomposition of H to obtain left and right singular vectors $U \in \mathbb{R}^{m(m_o+1)\times p(m_c+1)}$ and $V \in \mathbb{R}^{p(m_c+1)\times p(m_c+1)}$, and the diagonal matrix with decreasing singular values $\Sigma \in \mathbb{R}^{p(m_c+1)\times p(m_c+1)}$ such that

$$H = U\Sigma V^T.$$

The primal modes $\phi \in \mathbb{R}^{m\times p(m_c+1)}$ and adjoint modes $\psi \in \mathbb{R}^{p(m_c+1)}$ are given by

$$\begin{aligned}\phi &= XV\Sigma^{-1/2}\\ \psi &= \phi(\phi^T\phi)^{-1}.\end{aligned}$$

To compare the projection error for r-ranked reduced order model, we consider the r leading columns ϕ and ψ to get ϕ_r and ψ_r respectively. The projection errors for eigenmode v and adjoint mode w are given by $\epsilon(v,\phi_r)$ and $\epsilon(w,\psi_r)$ where $\epsilon(\cdot)$ is defined in Equation (3.15). For consistency with Section 3.4.1, we use $m_c = 14$ so that we use the same number of snapshots in the data matrix X. m_o is chosen to be 4. The comparison of projection error of lrDMD, OMD and DMD with ERA is shown in Figure A.1. Even though ERA has the same projection error for the unstable eigenmode compared to DMD, it shows significant improvement in the unstable



(a)

(b)

Figure A.2.: (a) Projection error for the unstable eigenmode and (b)Projection error for the unstable adjoint mode for lrDMD (green, square), ERA with $m_o = 4$ (black, diamond) and ERA with $m_o = 15$ for different rank approximations.

adjoint mode projection error. As seen in Figure A.1b, lrDMD outperforms ERA only at ranks higher than 5. However, Figure A.2 shows that increasing m_o improves ERA performance in adjoint projection error and even outperforms lrDMD although having the same higher error in eigenmode projection. This shows that adding delay coordinates by increasing m_o can significantly decrease the adjoint projection error but does not affect the eigenmode projection.

A.2. Model Parameters Optimization for Flow Reconstruction

In this appendix we give the closed form expressions for computing the gradient of the Lagrangian described in Equation 4.7 with respect to the state transition matrix A, the low-dimensional basis L and the observation matrix C. The gradients are as follows,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial A} &= \sum_{k=0}^{n-1} \boldsymbol{\eta}_{k+1|k} \mathbf{a}_{k|k}^H \\ &\quad + (\Lambda + \Lambda^H) A \Sigma (I - L^H C^H S^{-1} C L \Sigma), \\ \frac{\partial \mathcal{L}}{\partial L} &= \sum_{k=1}^{n} C^H K^H \boldsymbol{\eta}_{k|k} \mathbf{a}_{k|k-1}^H + \sum_{k=1}^{n-1} \boldsymbol{\eta}_{k+1|k} \mathbf{u}_k^H B^H \\ &\quad + C^H S^{-1} H^H \Sigma + C^H (T + T^H) C L \Sigma + W L (\Lambda + \Lambda^H) \\ &\quad - C^H S^{-1} C L \Sigma A^H (\Lambda + \Lambda^H) A \Sigma\end{aligned}$$

$$+ \mathbf{q}_0 \boldsymbol{\eta}_{0|0}^H + \sum_{k=1}^{n} (L\mathbf{a}_{k|k} - \mathbf{q}_k)\mathbf{a}_{k|k}^H,$$

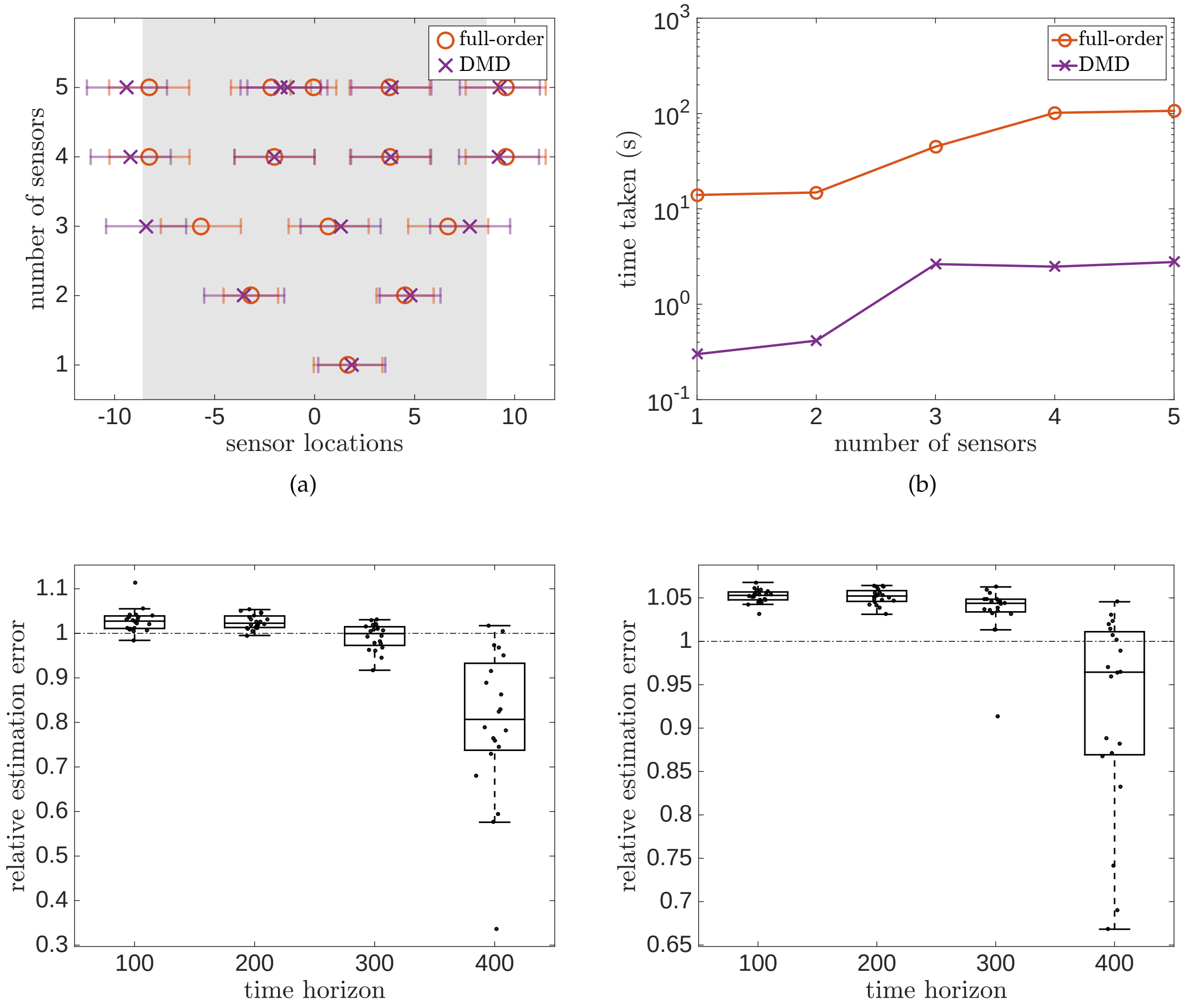
$$\frac{\partial L}{\partial C} = \sum_{k=1}^{n} K^H \boldsymbol{\eta}_{k|k}^H (\mathbf{q}_k - L\mathbf{a}_{k|k-1})^H - S^{-1} C L \Sigma A^H (\Lambda + \Lambda^H) A \Sigma L^H.$$

These gradients can be used to update the reduced-order model A, the low-dimensional subspace basis L and observation matrix C to optimize the objective function J described in Equation 4.4.

A.3. Optimization of Sensor Parameters

In this section we optimize for both the sensor location x_s and the shape parameter σ simultaneously for the Ginzburg-Landau system (Section 4.4.1). We also impose a constraint that the sensor shape parameter can not exceed 2, i.e. $\sigma \leq 2$. The setup of the Ginzburg-Landau system and data collection is the same as described in Section 4.4.1. The number of snapshots used as the training data is $n = 200$ and the rank of the reduced-order operator is $r = 10$. In this section we find optimal sensor locations and shape parameters for number of sensors $p \in \{1, 2, 3, 4, 5\}$ using the DMD reduced-order model and the full-order model. We then compare the performance of the estimation error for varying lengths of time horizons $n_t \in \{100, 200, 300, 400\}$.

Figure A.3 shows the sensor locations and shape parameters obtained using the DMD reduced-order model and the full-order model for number of sensors $p \in \{1, 2, 3, 4, 5\}$. The value of the shape parameter σ is signified as an interval of length σ around the sensor location x_s. In Figure A.3a, we see that the sensor locations and shape parameter values obtained using the reduced-order model closely approximate the solution using the full-order model. Moreover, time taken by by the reduced-order models is significantly lower than the full-order model (Figure A.3b). Finally, Figure A.3c and Figure A.3d show that for $n_t \in \{100, 200, 300\}$, the median relative estimation error of the DMD reduced-order model compared to the full-order model is greater than one. This shows that the reduced-order model outperforms the full-order model in terms of estimation error when the number of sensors $p \in \{1, 2, 3, 4, 5\}$ and the time horizon is short $n_t \leq 300$.



(a) (b)

(c) (d)

Figure A.3.: Simultaneous optimization of sensor location and shape parameter for the Ginzburg-Landau system. (a) The location of sensors and sensor shape parameters (indicated by an interval around the sensor location) obtained using the DMD reduced-order model and the full-order model (baseline) for different number of sensors. (b) Time taken to place different number of sensors using the DMD reduced-order model and the full-order model. Relative estimation error of the reduced-order model compared to the full-order model (baseline) for 20 independent simulation with random initial conditions for number of sensors (c) $p = 1$ and (d) $p = 5$, rank $r = 10$ and noise levels $v = 0.01$ and $w = 0.01$.

A.4. Data-driven Modal Analysis

A.4.1. Modal Decomposition

We have seen in Section 2.1.3, DMD provides a data-driven linear description of the dynamics of a system for a given set of snapshots. In this section we will show how DMD provides a decomposition of the flow as a linear combination of modes, each with a characteristic frequency of oscillation. Consider the spatio-temporal flow field $w(x,t)$ that can be expanded as follows

$$w(x,t) = \sum_{k=1}^{r} a_k(t)\phi_k(x)\alpha_k \tag{A.1}$$

where r is the number of modes, α_k is the magnitude of the contribution of the mode, while $\phi_k(x)$ is the spatial support and $a_k(t)$ is the temporal dynamics of the mode. DMD performs modal decomposition in such a way that the resulting modes have a single frequency dynamics. So if ϕ_k are the DMD modes, the temporal dynamics $a_k(t)$ for the mode can be written as

$$a_k(t) = e^{i\omega_k t}, \tag{A.2}$$

where $\omega_k \in \mathbb{C}$. The real part of ω_k gives the frequency of oscillation for the mode $\phi_k(x)$, while the imaginary part is the decay rate of the mode. Self-sustaining modes have real ω_k value while transient modes have a positive imaginary component in ω_k resulting in a decaying mode.

Let $A \in \mathbb{R}^{r\times r}$ be the r-ranked linear model for the flow provided by the DMD algorithm (see Section 2.1.3). This operator is used to generate the DMD modes and the associated oscillation frequencies. First, the eigenvalues and eigenvectors of A are found such that

$$AW = W\Lambda, \tag{A.3}$$

where the columns of the matrix W are the eigenvectors of A and Λ is the diagonal matrix built with the eigenvalues λ_k. Here λ_k is an approximation of $e^{(i\omega_k\Delta t)}$. Therefore, the frequency of oscillation of the DMD modes is given by $\textsc{Imag}(\ln(\lambda_k))/\Delta t$ while the exponential growth rate is given by $\textsc{Real}(\ln(\lambda_k))/\Delta t$. Finally, the DMD modes are given by

$$\Phi = UW \tag{A.4}$$

where the columns of the matrix Φ are the DMD modes, *i.e.* $\Phi := [\phi_1 \cdots \phi_r]$.

A.4.2. Mode Amplitude Estimation

Let V_i^j denote a matrix with columns given by the flow snapshots such that $V_i^j = [w(x,t_i), \cdots, w(x,t_j)]$. There are several ways of defining the amplitudes $\{\alpha_1, \cdots, \alpha_k\}$ of the DMD modes $\{\phi_1, \cdots, \phi_k\}$. One of the simplest ways is to look at the projection of the first snapshot V_0^0 onto the DMD modes Φ. However, this does not account for the higher contribution of the temporally growing DMD modes and lower contribution of decaying DMD modes in the flow. Instead, we employ the method described in [77] to compute the optimal amplitudes of the DMD modes. Consider the modal decomposition described in Equation A.1 in the matrix form

$$V_0^{n-1} = \Phi \; D_\alpha \; R \tag{A.5}$$

where D_α is the diagonal matrix of the DMD mode amplitudes $\{\alpha_1, \cdots, \alpha_r\}$ and R is the following Vandermonde matrix,

$$R = \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_r & \cdots & \lambda_r^{n-1} \end{bmatrix}. \tag{A.6}$$

Let the vector α be defined as

$$\alpha := \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_r \end{bmatrix}. \tag{A.7}$$

We find the optimal amplitudes α by solving the following optimization problem

$$\min_{\alpha} \left\| V_0^{n-1} - \Phi \; D_\alpha \; R \right\|_F^2 \tag{A.8}$$

where $\|A\|_F$ is the Frobenius norm of A. This optimization problem is convex and can be solved using the least-squares approach. The optimal solution is given by

$$\alpha = P^{-1}q \tag{A.9}$$

where $P = \Phi^*\Phi \circ \overline{(RR^*)}$ and $q = \overline{\text{diag}(RV\Sigma^T)}$. The overline represents complex-conjuate of a matrix or a vector while diag of a matrix is a vector determined by the main diagonal of the given matrix.

Accounting for Experimental Noise

In any experimental data, there is going to be some noise in the estimated eigenvalues of the flow dynamics. To account for the noise, we run the experiment at the same flow conditions $p > 1$ times and the imaging data is collected independently. In this study we choose $p = 5$. The following smoothing technique is then employed to find the dominant frequencies in the flow.

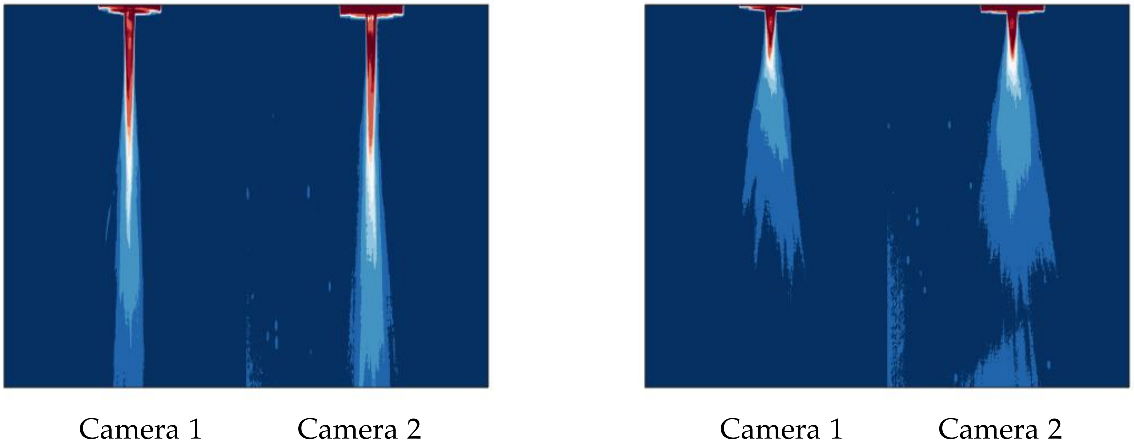
Let $\alpha_{i,j}$ and $f_{i,j}$ be the amplitude and frequency of oscillation respectively for the i-th DMD mode in the j-th run of the experiment at the same flow conditions. We use a Gaussian kernel $\kappa(f, f') = \exp((f - f')/(2\sigma))$ with $\sigma = 0.01$ to filter the DMD mode amplitude profiles where f and f' are frequencies. The estimated amplitude α is a function of the frequency f such that

$$\alpha(f) = \frac{1}{p}\sum_{i=1}^{r}\sum_{j=1}^{p}\kappa(f, f_{i,j})\alpha_{i,j}. \tag{A.10}$$

For our experimental data we observe that the resulting amplitude profile $\alpha(f)$ is more robust to the choice of the number r of DMD modes compared to the frequency spectrum extracted from a single run of the experiment.

A.5. Stationary Modes of Liquid Jet with a Co-axial Airblast Atomizer

Figure A.4 shows the stationary DMD modes, corresponding to $St = 0$, for a liquid core surrounded by a co-axial airblast atomizer without swirling gas-stream and with gas velocity swirl ratio of 0.5. While these modes do not provide information about the temporal dynamics of the flow, they can elucidate physically important parameters of the flow such as spray angle [23].



(a) no swirl

(b) with swirl

Figure A.4.: Stationary DMD mode corresponding to $St = 0$ for the flow (a) case 1 (with no swirling gas) and (b) case 2 (with swirl ratio of 0.5). For both the cases, The liquid Reynolds number $Re_\ell = 1100$ and gas Reynolds number $Re_g = 21300$.

In Figure A.4, we see that the liquid core shape is consistent with the two orthogonally facing cameras for both the flow conditions, with higher spray angle for the flow with gas velocity swirl ratio of 0.5.