Advances in Data-Driven Modeling and Sensing for High-Dimensional Nonlinear Systems

Samuel E. Otto

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Mechanical and Aerospace Engineering
Adviser: Professor Clarence W. Rowley

May 2022

Abstract

Accurate and efficient models of physical processes like fluid flows are crucial for applications ranging from forecasting the weather to controlling autonomous aircraft and suppressing combustion instabilities in liquid fueled rocket engines. These models allow us to predict what the system will do — oftentimes in response to an input or design characteristics that we would like to choose intelligently — as well as to detect what the real system is doing from limited and costly sensor measurements. The main challenge is that the equations governing complex systems like fluid flows that we might derive from first principles are routinely nonlinear and involve too many variables to be simulated by a computer or sensed in real time. Therefore, we aim to construct and leverage simplified models of these complex systems that capture the most important aspects of its behavior for the task at hand, while relying on a much smaller number of variables that can be simulated or sensed in real time. While highly effective and well-studied techniques exist when the system is linear, in many important cases the system is operating too far away from an equilibrium state to employ linearization or other linear approximation techniques. In this thesis, we make use of data collected from the underlying complex system or simulations performed ahead of time in order to identify patterns and construct simplified models based on them. In order to build simplified predictive models, we present a variety of techniques based on projecting the governing equations onto manifolds identified from data. Such manifolds must be nonlinear in order to find models involving the smallest number of variables. We also find that in order to build models of systems with selective sensitivity, such as shear-driven fluid flows, it is important to incorporate information from the linearized adjoint of the governing equations. We also describe an alternative viewpoint for modeling based on converting nonlinear dynamics into linear dynamics in a function space via data-driven approximation of Koopman operators. Finally, we present a constellation of data-driven techniques enabling us to find minimal sets of sensors or measurements to robustly infer what a highly nonlinear system is doing.

Acknowledgements

I am immensely grateful to my family, friends, and mentors who supported me on my academic journey. Most of all, I want to thank my wife Anastasia Bizyaeva who has always been on my team, my mom and dad whose love and support got me here, Ben Kelminson, who might as well be my brother, Alberto Padovan for his friendship and exhilarating conversations, as well my friends at Rojos roastery, where the majority of this thesis was written.

I especially want to thank my advisor Prof. Clarence Rowley for being a supportive mentor and helping me grow as a researcher. I also want to thank Prof. Charles Fefferman for his mentorship and my thesis committee members Prof. Sun-Yuan Kung and Prof. Naomi Leonard.

I am grateful to the following people who made specific contributions to the work presented in this thesis: Prof. Gregory Blaisdell, Dr. Shih-Chieh Lo, Prof. Tasos Lyrintzis, and Dr. Kurt Aikens for providing the code used to compute the shock-mixing layer interaction; Prof. Scott Dawson for providing us with the data from his cylinder wake simulations; Prof. Andres Goza and Prof. Michael Mueller for helpful discussions about nonlinear measurement selection and its applications; and William Eggert for his invaluable help and collaboration on the initial iterations of the LRAN code.

Finally, I would like to thank my thesis readers Prof. Anirudha Majumdar and Prof. Peter Ramadge for their meticulous reading and thoughtful comments. This was quite an undertaking, and I am immensely grateful.

The research presented in this thesis was supported by the Army Research Office under grant number W911NF-17-1-0512, Air Force Office of Scientific Research under grant number FA9550-19-1-0005, and DARPA. S. E. Otto was supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2039656.

This dissertation carries T#3429 in the records of the Department of Mechanical and Aerospace Engineering

To the Fox

Contents

Part I

Background, Summarized Contributions, and Extensions

Chapter 1

Introduction

Accurate and computationally efficient models of physical systems enable a variety of important engineering and scientific tasks to be carried out. Such tasks include

1. making forecasts of what a system will do given its current configuration or "state",
2. tracking or "estimating" the current state in real time using streaming sensor measurements,
3. making decisions or controlling the system via an input to achieve some objective,
4. solving "inverse problems" to determine unknown parameters or initial conditions,
5. deciding what measurements from the system are needed to carry out the preceding tasks,
6. as well as design optimization of the system's parameters to enhance a performance metric.

Several of the above tasks require simulating the model a large number of times, or performing other complex calculations involving the system's definition, sometimes in real time. These demands place limits on the complexity and scale of the model depending on the available computational resources and constraints on the computation time. In many physical systems such as fluid flows, the models we obtain from first principles are too complex to be simulated efficiently enough to perform the desired tasks. The complexity of the physics-based models comes from their nonlinearity and high-dimensional state space.

Fortunately, the behaviors exhibited by many complex physical systems including fluid flows are dominated by coherent spatiotemporal patterns. One of the first observations of coherent structures in turbulence can be found in [32], where they are strikingly visible in experimental shadowgraphs.

Low-dimensional modeling of these coherent structures as in [250, 114, 228] offers a promising approach to capture the dominant emergent behavior of the underlying system. Mathematically, a collection of coherent structures defines a manifold or a subspace of the state space that the system's trajectories remain close to over time. By restricting the dynamics to evolve on this manifold or subspace, one obtains a "reduced-order model" (ROM) with a lower state dimension than the original system or "full-order model" (FOM). While powerful and well-studied techniques such as those discussed in [10] and [20] are available for making low-dimensional approximations of linear systems, the model reduction problem for nonlinear systems operating far from equilibria is more challenging and remains largely unsolved. For reviews of some modern techniques for model reduction of linear and nonlinear dynamical systems, one can consult [219, 14, 20, 228].

The success of linear model reduction approaches stems from the fact that the input-output behavior of finite-dimensional linear systems admits an elegant and essentially complete description. Consider a linear system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx, \end{aligned} \tag{1.1}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^p$, and output $y \in \mathbb{R}^q$. The output of this system is described in the time domain by the "variation of constants" formula

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau \tag{1.2}$$

and in the frequency domain by its Laplace transform

$$\mathcal{L}\{y\}(s) = C(sI - A)^{-1}x(0) + C(sI - A)^{-1}B\mathcal{L}\{u\}(s), \qquad s \in \mathbb{C}. \tag{1.3}$$

In contrast, the behavior of nonlinear systems can be much more complicated. As an illustration, the Lorenz-63 system [161] given by

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1) \\ \dot{x}_2 &= x_1(\rho - x_3) - x_2 \\ \dot{x}_3 &= x_1x_2 - \beta x_3 \end{aligned} \tag{1.4}$$

is a simplified model of a hydrodynamic flow and possesses only three state variables with quadratic

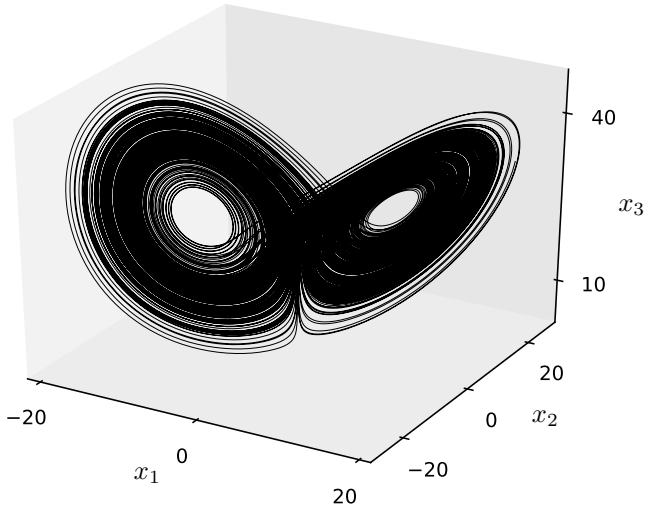


Figure 1.1: A trajectory of the Lorenz system shown after a sufficiently long time so that it closely approximates the Lorenz attractor.

nonlinearities. Over a range of parameter values this system yields chaotic and mixing dynamics that evolve onto the so called "Lorenz attractor" shown in Figure 1.1 for $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$. Embedded in this attractor are an infinite number of unstable periodic orbits as well as a dense orbit that passes infinitesimally close to every point on the attractor an infinite number of times [104]. Such behavior is impossible for a finite-dimensional linear system (see Section 2.7 of [103]).

The complexity arising from nonlinear interactions casts doubt on whether any computationally tractable systematic analysis of a nonlinear system's structure or governing equations can reveal the coherent behaviors that are likely to emerge in simulation or in experiment, except in specialized settings. This motivates the use of data collected from the system together with recent advances in statistics and machine learning to extract the dominant patters and construct reduced-order models governing their behavior. *I shall present several advances in this direction, with the organizing principle being the use of data to confront the inter-related challenges presented by nonlinearity and high-dimensionality.*

This thesis is organized into two parts. The first part summarizes the current contributions by the author and provides detailed extensions of this work, while situating it within the context of existing techniques and challenges. The methods and applications I shall discuss come in three flavors, each getting its own chapter in part I:

1. **Chapter 3:** Projecting high-dimensional dynamics onto low-dimensional nonlinear manifolds

describing coherent structures

2. **Chapter 4:** Turning nonlinear dynamics into linear dynamics of functions on the state space (called observables) by approximating the infinite-dimensional linear operators governing their evolution.

3. **Chapter 5:** Selecting optimal sensors or observables from among a discrete set to reconstruct desired information about the system or to build reduced-order models.

The second part contains selected papers by the author in which the current contributions summarized and contextualized in part I are explained in detail. All papers by the author will be denoted with a star, e.g. [196]*, and papers appearing in Part II will be denoted with two stars, e.g., [194]**.

Chapter 2

Analytical techniques

Before discussing data-driven approaches, it is important to ground our discussion in some special cases where direct analyses of the governing equations are sufficient to shed light on emergent spatio-temporal coherent structures. Such analytical methods have been the dominant approach for studying dynamical systems since the field was pioneered by H. Poincaré [207, 208] since they often produce concrete theoretical results and do not require performing costly experiments or simulations of the system. One of the most popular and successful analytical techniques is the reduction of a system onto a center manifold [104] defined in the neighborhood of an equilibrium point. By expanding the center manifold in a Taylor series, it is possible to capture the structure of local bifurcations. On the other hand, analytical techniques usually rely on perturbation theory, which can only provide local information about a system near an equilibrium, bifurcation point, or orbit. In addition, the cost of carrying out such analyses on large-scale systems such as discretized fluid flows may be comparable to running one or more simulations. Analytical techniques are also intrusive, meaning that they require direct algebraic manipulation of the governing equations. This can be challenging when the equations are embedded in large-scale computer programs intended only to simulate them.

2.1 Modeling dynamics locally

The simplest type of analytical methods stem from analyses of the linearized dynamics about an equilibrium point. If the trajectory of a nonlinear system remains sufficiently close to an equilibrium point, then the dynamics are accurately described by the system's linearization about the equilibrium. For linear systems, a variety of well-studied model reduction techniques such as those

reviewed in [10] and [20] can provide guarantees on the accuracy of the resulting reduced-order model for capturing input-output behavior. Implementation of these techniques rely on numerical linear algebra involving only the matrices defining the linear system. However, in some large-scale applications such as model reduction of linearized fluid flows, the computational cost associated with model reduction based directly on the system matrices can be greater than the cost of running a collection of simulations. In fact, balanced proper orthogonal decomposition (BPOD) [225] reduces the cost of the popular balanced truncation technique for model reduction [182] by computing an efficient data-driven approximation using impulse-response trajectories of the forward and adjoint systems. However, specialized algorithms for computing low-rank solutions of large-scale sparse Lyapunov equations such as those in [205] and [151] may be competitive in terms of efficiency, but more intensive in terms of requiring "intrusive" software implementation.

For model reduction of the original nonlinear governing equations, one approach taken by [13, 6, 118, 120] is to define a Petrov-Galerkin projection using subspaces chosen by model reduction approaches for the linearized dynamics. In a similar spirit, [126, 174] suggest decomposing the dynamics into a linear part and a nonlinear part about a chosen base state, usually the turbulent mean or a fixed point, and viewing the nonlinear terms as broad-spectrum forcing acting on the linear terms. By studying the transfer function associated with the linear terms, [174] found that certain features of the nonlinear forcing are selectively amplified by linear interactions to produce the coherent patterns observed in turbulent pipe flow. In particular, the transfer function for a variety of turbulent flows linearized about their turbulent mean has a large singular value only at select frequencies, and at those frequencies the transfer function is approximately low rank. This observation has lead to successful modeling approaches for a variety of turbulent flows based on the transfer function of the linearized dynamics [297].

However, as trajectories of the system move away from the chosen base state, the nonlinear contributions can become significant. Projection operators defined based on the linearized dynamics about a fixed point may completely ignore the contributions of small-scale features that strongly influence the dynamics at later times due to nonlinear interactions. Similarly, energetic coherent structures can cause large deviations from the turbulent mean, resulting in nonlinear interactions that cannot be adequately modeled as broad-spectrum noise. As A. Padovan, C. W. Rowley, and I show in [197]*, these coherent departures from the mean can cause analyses based on the transfer function to fail due to energetic cross-frequency interactions that are ignored by linear analysis about the mean. We can more accurately predict the structures that dominate perturbed responses of a flow by studying the linearization about an energetic periodic orbit. In doing so, we again

find selective amplification — except now many frequencies may be activated by forcing at a single frequency due to triadic interactions with the frequencies present in the base flow.

Truncated Volterra series expansions provide another local approach for model reduction of bilinear [14, 17, 88] and quadratically bilinear (QB) [18, 19] input-output systems operating near a stable equilibrium point. The Volterra series allows for generalization of the $\mathcal{H}_2$ norm, usually defined only for stable linear systems, to broader classes of nonlinear systems in the neighborhood of an equilibrium point. These promising approaches can extend the region of validity for the projection-based reduced-order model for arbitrary input signals. However, they are limited to a neighborhood of a fixed point and by the number of terms that are retained in the truncated Volterra series. To keep the computational cost manageable, the series is truncated at three terms, which is sufficient to capture the second-order interactions between inputs at different times through the second-order Volterra kernel.

2.2 Modeling dynamics globally

Analytical techniques yielding truly global information about the dynamics are rare and somewhat more difficult to wield for complex high-dimensional systems. Melnikov's method [104] is a perturbative approach which can be used to predict the onset of chaos due to homoclinic bifurcations induced by periodic excitation. However, one must already know that a homoclinic orbit exists and be able to compute with it. Model reduction onto an inertial manifold [92, 90, 124] is another technique that is capable of capturing global nonlinear dynamics such as chaotic attractors. However, the existence of an inertial manifold can only be proved in certain highly specialized settings, such as for dissipative dynamical systems. Furthermore, computing an inertial manifold and projecting the dynamics onto it is extremely difficult in practice. Due to these difficulties, one usually works with an approximate inertial manifold as in [263, 124, 234, 151, 170], which can be a useful for nonlinear model reduction and control of systems for which the method applies. One approach to seek an approximate inertial manifold is to decompose the dynamics into a linear and nonlinear part where the linear part is self-adjoint. One then seeks an approximately invariant manifold expressed as a graph over the subspace spanned by the least dissipative eigenvectors of the self-adjoint, linear part of the dynamics. However, as pointed out in [102], this approach can fail when modes with small spatial length scale become dynamically important. In particular, small scales can exhibit behaviors that do not depend directly on the larger scales, and can even drive the large scale dynamics through non-normal mechanisms [264]. Even when the states can be shown to reside in a small neighbor-

hood of an approximate inertial manifold, the method of proof relies on the dissipative part of the dynamics. When the dissipative length scale is small, as in the case of high Reynolds number fluid flows, the resulting approximate inertial manifold is too high-dimensional to be useful [91].

While the specific approach whereby an approximate inertial manifold is expressed as a graph over the least dissipative eigenspace of the linearized dynamics may not always be the best approach, so called nonlinear Galerkin projection [168] of the dynamics onto a nonlinear manifold capturing the relevant trajectories is very promising for systems exhibiting coherent patterns in general. A main theme of this thesis is that *data collected from the dynamical system can be used to identify such a manifold as well as an associated projection operator for reduced-order modeling.* These data are the most reliable indicator of the coherent patterns in the dynamics that emerge over time, and they do not rely on any kind of perturbative analysis or proximity to a fixed point or other type of orbit. Moreover, data-driven techniques tend to be less intrusive than analytical techniques since they rely only on simulation data, and perhaps also on linearized adjoint simulations that are already commonly used for design optimization.

Chapter 3

Data-driven reduced-order modeling

Reduced-order modeling entails approximating the original system or full-order model (FOM) by another system or "reduced-order model" (ROM) with a smaller state dimension. This is usually accomplished by constraining the reduced-order model to evolve on a subset of the original state space with fewer degrees of freedom, i.e., a low-dimensional submanifold of the state space. In Chapter 2 we discussed several analytical techniques such as reduction onto center and approximate inertial manifolds that may be found in the neighborhoods of equilibrium points and for certain types of dissipative systems. However, as we pointed out at the end of Chapter 2, these techniques have significant limitations, especially for systems operating far away from equilibria or exhibiting sensitive dependence on low-energy features. In the data-driven approach we determine, or learn, the underlying manifold for the reduced-order model based on data collected from the system. This chapter presents background on existing data-driven techniques, analysis of fundamental challenges, and techniques developed by the author to address these challenges.

As we shall review in Section 3.1, the majority of existing techniques for data-driven model reduction follow a two-step recipe:

1. use data from the system to identify the manifold or subspace representing the most salient coherent structures exhibited by the dynamics
2. identify governing equations for the system confined to this manifold either by
   (a) data-driven learning

(b) or by projecting the governing equations into the tangent space.

In many cases, each step is easy to perform on its own, leading to simplicity and efficiency of the overall approach. Data-driven manifold learning techniques are capable of providing excellent low-dimensional representations of the state with very little energy or variance unaccounted for. When the most energetic features described by the learned manifold also have the largest influence on the future states of the system, the model we obtain via the two-step approach will be highly accurate. This resembles the situation described in Section 2.2 when models based on approximate inertial manifolds are accurate.

On the other hand, systems where low-energy features exert a large influence on the way the system evolves over time are extremely challenging to model using existing data-driven techniques. Such systems are abundant in fluid dynamics, especially in shear flows where two layers of fluid move past each other at different speeds [264, 242, 114]. Moreover, we find that the sensitivity mechanisms by which low-energy features drive high-energy dynamics can be highly nonlinear. Capturing this kind of sensitivity is the fundamental challenge motivating the new techniques we develop in this chapter.

There are essentially two reasons for the aforementioned difficulty in modeling sensitivity to low-energy features. First, step 1 in the two-step data-driven model reduction recipe ignores low-energy features by definition. This is because the notion of salience used to find low-dimensional representations is based on reconstruction accuracy and not on how accurately predictions are made in the future. This limitation suggests using criteria that combine reconstruction and prediction accuracy to identify the underlying manifold. In particular, steps 1 and 2 of the data-driven model reduction recipe can be combined and performed simultaneously to accurately model sensitivity to low-energy features. When the amount of data collected from the system exceeds the state dimension, then this approach is essentially sufficient to identify the correct sensitivity mechanisms and build an accurate model.

For high-dimensional systems where the state dimension greatly exceeds the amount of available data, combining steps 1 and 2 of the data-driven model reduction recipe may still not work due to a second fundamental difficulty, which is linear-algebraic in nature. This "curse of dimensionality" poses a serious challenge for constructing reduced-order models of fluid flows where the state dimension routinely exceeds the amount of data collected from simulations by several orders of magnitude. As we discuss later in Section 3.3.3, it is impossible to determine which features a system is sensitive to from a number of input-output pairs smaller than the input dimension. This is in contrast to the

ease with which we find subspaces or manifolds that accurately represent a system's output when it is low-dimensional. Additional constraints or different kinds of data are necessary to determine sensitivity mechanisms, even when the system is sensitive to only a small number features. In particular, we find that data coming from linearized adjoint simulations of the full-order model provides the key information we need to model sensitivity mechanisms. This idea is closely related to the randomized SVD algorithm [108] and to balanced proper orthogonal decomposition (BPOD) [225]. Both techniques rely on data gathered from an operator and its adjoint to build approximations that capture sensitivity. We present reduced-order modeling approaches which extend these ideas to large-scale nonlinear systems.

3.1 Background on data-driven model reduction

3.1.1 Linear projection methods (POD-Galerkin)

One of the easiest and by far the most common approach for step 1 in the standard data-driven model reduction recipe is to identify a subspace using Principal Component Analysis (PCA) [116] or Proper Orthogonal Decomposition (POD) [23, 59], also known as Karhunen-Loève decomposition [131, 160]. The use of POD for model reduction of fluid flows was pioneered by J. L. Lumley [163] and is reviewed in [114], which should be consulted for more details. Both POD and PCA rely on Singular Value Decomposition (SVD) to identify orthonormal bases for subspaces that capture the most energetic state fluctuations with the difference being that POD identifies a genuine subspace containing the origin, while PCA identifies an affine subspace centered about the mean. If the data $\{x_j\}_{j=1}^m$ live in a Hilbert space $\mathcal{X}$, then POD and PCA identify an r-dimensional isometry $U : \mathbb{C}^r \to \mathcal{X}$ so that its range captures the most energy about a center point c. In particular, U solves the optimization problem

$$\underset{\substack{U:\mathbb{C}^r\to\mathcal{X}\\ U^*U=I_r}}{\text{minimize}} \quad \frac{1}{m}\sum_{j=1}^{m} \|(x_j - c) - UU^*(x_j - c)\|^2, \tag{3.1}$$

where the subspace contains $c = 0$ for POD or contains the average $c = \frac{1}{m}\sum_{j=1}^m x_j$ for PCA. The operator $P = UU^*$ is the orthogonal projection onto the r-dimensional subspace $\operatorname{Range} U$. Hence, the above optimization problem is identifying the "best" rank r projection that minimizes the sum of square errors between the original and projected data. Minimizing the average projection error as in Eq. 3.1 is equivalent to maximizing the "energy" contained in the projection subspace $\operatorname{Range} U$

according to

$$\underset{\substack{U:\mathbb{C}^r\to\mathcal{X}\\ U^*U=I_r}}{\text{maximize}} \ \frac{1}{m}\sum_{j=1}^{m}\|U^*(x_j-c)\|^2 = \operatorname{Tr}\left[U^*\left(\frac{1}{m}\sum_{j=1}^{m}(x_j-c)(x_j-c)^*\right)U\right], \tag{3.2}$$

where the matrix $\frac{1}{m}\sum_{j=1}^{m}(x_j-c)(x_j-c)^*$ is referred to as an empirical covariance matrix. Let the centered data be arranged as an operator $X_c:\mathbb{C}^m\to\mathcal{X}$ defined by

$$X_c = \begin{bmatrix}(x_1-c) & \cdots & (x_m-c)\end{bmatrix} : w \ \mapsto \ w_1(x_1-c)+\cdots+w_m(x_m-c). \tag{3.3}$$

Then the solution is given by $U = \begin{bmatrix}u_1 & \cdots & u_r\end{bmatrix}$, where $\{u_j\}_{j=1}^m$ are the left singular vectors of

$$\frac{1}{\sqrt{m}}X_c = \sum_{j=1}^{m}\sigma_j u_j v_j^T, \tag{3.4}$$

arranged in decreasing order $\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_m$. The singular value decomposition of a matrix containing observations of the state is easy to compute and so one readily obtains an orthogonal basis that is optimal in the energetic sense of POD or PCA.

Once the coherent structures in the form of an energetic state subspace have been identified using POD or PCA, there are two main approaches for obtaining a reduced-order model. Suppose the governing equations of the full-order model

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}x &= f(x,u), \qquad x(0)=x_0\\ y &= g(x)\end{aligned} \tag{3.5}$$

with state $x\in\mathcal{X}$, input u, and output y are known. A reduced-order model can be built by constraining the state $\hat{x}$ to lie in the identified subspace $\hat{x}\in c+\operatorname{Range}U$ using a technique referred to as Galerkin projection. While $\hat{x}$ lies in $c+\operatorname{Range}U$, the time derivative $f(\hat{x},u)$ may not lie tangent to $c+\operatorname{Range}U$, i.e. in $T_{\hat{x}}(c+\operatorname{Range}U)=\operatorname{Range}U$, and so one may take the closest approximation of $f(\hat{x},u)$ in $\operatorname{Range}U$ given by its orthogonal projection

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\hat{x} = \underset{v\in\operatorname{Range}U}{\operatorname{argmin}}\|f(\hat{x},u)-v\| = UU^*f(\hat{x},u), \tag{3.6}$$

by the projection theorem for the Hilbert space $\mathcal{X}$. Similarly, the nearest point in $c+\operatorname{Range}U$ to

the initial condition x_0 is given by the shifted orthogonal projection

$$\hat{x}(0) = c + UU^*(x_0 - c). \tag{3.7}$$

In the reduced coordinate system defined by $\hat{x} = c + Uz$, the reduced-order model is given by

$$\boxed{\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} z &= U^* f(c + Uz, u), \qquad z(0) = U^*(x_0 - c) \\ \hat{y} &= g(c + Uz). \end{aligned}} \tag{3.8}$$

One can easily analyze the error incurred by a Galerkin-based reduced-order model of Lipschitz governing equations, i.e., $\|f(x,u) - f(x',u)\| \leq L\|x - x'\|$ for every $x, x' \in \mathcal{X}$, by observing that

$$\begin{aligned} \|x(t) - \hat{x}(t)\| &\leq \|x(0) - \hat{x}(0)\| + \int_0^t \|f(x(\tau), u(\tau)) - UU^* f(\hat{x}(\tau), u(\tau))\| \,\mathrm{d}\,\tau \\ &\leq \min_{\hat{x}_0 \in c + \mathrm{Range}\, U} \|x_0 - \hat{x}_0\| + \int_0^t \left[L\|x(\tau) - \hat{x}(\tau)\| + \min_{v \in \mathrm{Range}\, U} \|f(\hat{x}(\tau), u(\tau)) - v\| \right] \mathrm{d}\,\tau \end{aligned} \tag{3.9}$$

and applying the Grönwall inequality given below by Lemma 3.1.1 to obtain

$$\boxed{\|x(t) - \hat{x}(t)\| \leq \min_{\hat{x}_0 \in c + \mathrm{Range}\, U} \|x_0 - \hat{x}_0\| e^{Lt} + \int_0^t e^{L(t-\tau)} \min_{v \in \mathrm{Range}\, U} \|f(\hat{x}(\tau), u(\tau)) - v\| \,\mathrm{d}\,\tau.} \tag{3.10}$$

The minima appearing in Eq. 3.10 are a consequence of the orthogonal projection of the initial condition and the time derivative into the appropriate subspaces. This local optimality indicates that Galerkin projection onto a POD or PCA subspace is sensible for making an upper bound on the modeling error small. While choosing the orthogonal projection minimizes the reduced-order modeling error for short time horizons, the above bound does not exclude the possibility that the error grows exponentially. This kind of exponential growth takes place when the projection subspace $\mathrm{Range}\, U$ does not capture low-energy fluctuations that are amplified by the dynamics at later times.

Lemma 3.1.1 (inhomogeneous Grönwall inequality)**.** *Suppose that $w : [0,T] \to \mathbb{R}$ and $b : [0,T] \to \mathbb{R}$ are integrable functions satisfying*

$$w(t) \leq a + \int_0^t [Lw(\tau) + b(\tau)] \, d\tau \qquad \forall t \in [0,T] \tag{3.11}$$

for some constants $a, L \in \mathbb{R}$. Then, w is bounded according to

$$w(t) \leq ae^{Lt} + \int_0^t e^{L(t-\tau)} b(\tau)\, d\tau \qquad \forall t \in [0, T]. \tag{3.12}$$

Proof. The proof is given in Appendix 3.A. □

3.1.2 Linear models of projected dynamics (POD-DMD)

On the other hand, the governing equation may not be known, in which case, a model of the dynamics can be built in the subspace identified by PCA or POD using data-driven techniques. It is common to have a data set consisting of snapshot pairs of the state $\{(x_j, x_j^\sharp\}_{j=1}^m$ where $x_j^\sharp = x(t + \Delta t)$ is obtained by evolving the dynamics $\dot{x} = f(x)$ from the initial condition $x(t) = x_j$ over a time Δt. In such a case, there are many techniques that seek to approximate the discrete-time dynamics, i.e., the flow map

$$x(t + \Delta t) = F^{\Delta t}(x(t)) \tag{3.13}$$

from the snapshot pairs $x_j^\sharp = F^{\Delta t}(x_j)$, $j = 1, \ldots, m$. Perhaps the simplest technique is Dynamic Mode Decomposition (DMD), introduced by P. J. Schmid [243, 240], in which a linear approximation of $F^{\Delta t}$ is sought. Given the snapshot data arranged into matrices

$$X = \begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix}, \qquad X^\sharp = \begin{bmatrix} x_1^\sharp & \cdots & x_m^\sharp \end{bmatrix}, \tag{3.14}$$

a least squares approximation of $F^{\Delta t}$ is given by

$$A = X^\sharp X^+ \tag{3.15}$$

where X^+ denotes the Moore-Penrose pseudoinverse of X. However, it is generally impossible to work with A directly since the state dimension, and hence the dimension of A, is enormous for discretized fluid flows of interest. Moreover, the pseudoinverse cannot be stably computed when the matrix X has small singular values as are often encountered in practice. In such cases when the state dimension is large compared to the number of snapshots, m, the operator A is low rank with its nullspace containing the orthogonal complement of the range of X. Consequently, if U is an isometry with $\operatorname{Range} U = \operatorname{Range} X$, then the reduced operator

$$\hat{A} = U^* A U \tag{3.16}$$

has the same nonzero eigenvalues as A, but has dimension at most m. Moreover, it is shown in [266] that if $\hat{v}$ is an eigenvector of $\hat{A}$ with nonzero eigenvalue λ, then the corresponding eigenvector of A with eigenvalue λ is given by

$$v = \frac{1}{\lambda} X^\sharp (U^* X)^+ \hat{v}, \tag{3.17}$$

as can be readily verified via the properties of the Moore-Penrose pseudoinverse:

$$\begin{aligned} Av = \frac{1}{\lambda} X^\sharp X^+ X^\sharp (U^* X)^+ \hat{v} &= \frac{1}{\lambda} X^\sharp (U^* X)^+ U^* X^\sharp X^+ U \hat{v} \\ &= \frac{1}{\lambda} X^\sharp (U^* X)^+ \hat{A} \hat{v} = \lambda v. \end{aligned} \tag{3.18}$$

Obtaining the eigenvector v using Eq. 3.17 is easier than directly computing the eigenvectors of A since U^*X is at most an $m \times m$ matrix. This is referred to as "exact DMD" [266].

The utility of working with the smaller matrix $\hat{A}$ motivates computing a projection of A into a POD subspace obtained from the data X. This subspace closely approximates the range of X in the energetic sense described above. Letting $U_r \Sigma_r V_r^* = \sigma_1 u_1 v_1^* + \cdots + \sigma_r u_r v_r^*$ denote the rank-r truncation of a singular value decomposition $X = U\Sigma V^* = \sigma_1 u_1 v_1^* + \cdots + \sigma_m u_m v_m^*$, then the projection of A into the r-dimensional POD subspace is given in POD coordinates by the matrix

$$\hat{A}_r = U_r^* X^\sharp V_r \Sigma_r^{-1} = U_r^* X^\sharp \left(U_r^* X\right)^+. \tag{3.19}$$

It is important to recognize that $\hat{A}_r$ is precisely the DMD matrix computed using the POD coefficients $z_j = U_r^* x_j$, $z_j^\sharp = U_r^* x_j^\sharp$ of the data arranged into matrices $Z = U_r^* X$ and $Z^\sharp = U_r^* X^\sharp$. The resulting reduced-order model of the original flow map is given by its least-squares approximation with respect to the data in POD coordinates

$$z(t + \Delta t) = U_r^* F^{\Delta t}(U_r z(t)) \approx \hat{A}_r z(t), \qquad \hat{A}_r = Z^\sharp Z^+. \tag{3.20}$$

Essentially the same least squares procedure can be used to build a linear model that incorporates piece-wise constant actuation signals $\{u_j\}_{j=1}^m$ applied during each time interval, except now it is referred to as Dynamic Mode Decomposition with control or DMDc [211]. In DMDc, one uses the same least-squares procedure to obtain a reduced-order model of the form

$$z(t + \Delta t) = U_r^* F^{\Delta t}(U_r z(t), u) \approx \hat{A}_r z(t) + \hat{B}_r u, \qquad \begin{bmatrix} \hat{A}_r & \hat{B}_r \end{bmatrix} = Z^\sharp \begin{bmatrix} z_1 & \cdots & z_m \\ u_1 & \cdots & u_m \end{bmatrix}^+. \tag{3.21}$$

If instead of $x_j^\sharp$, we have approximations of the time derivative $\dot{x}_j = f(x_j, u_j)$, then a similar procedure can be used to find a linear approximation of f in POD coordinates.

3.1.3 Nonlinear models of projected dynamics (POD-kitchen sink)

Of course in many cases, the dynamics are nonlinear, and so a linear approximation as in DMD may be an inadequate description. To construct a data-driven model in the reduced coordinate system, we can select a "dictionary" of real-valued nonlinear functions $\psi_1, \ldots, \psi_N$ to be used in an approximation of the dynamics of each POD coefficient

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}[z]_i = \langle u_i, f(U_r z, u)\rangle \approx c_{i,1}\psi_1(z,u) + \cdots + c_{i,N}\psi_N(z,u) = c_i^T \psi(z,u), \qquad i = 1, \ldots, r. \quad (3.22)$$

Reasonable choices of dictionary elements may be determined by examining the governing equations. For instance, if the dynamics have quadratic nonlinearities, as is true for the incompressible Navier-Stokes equations, then the dynamics of POD coefficients are also described in terms of quadratic nonlinearities. We may then solve a least-squares problem to fit the coefficients $c_{i,j}$ in Eq. 3.22 to a data set made up of inputs u_j, POD coefficients of states $z_j = U_r^* x_j$, and approximations of their time derivatives.

However, the number of terms N that must be considered in such an approximation can grow very large, and can easily exceed the number of data points m. In this case, the least squares problem for the coefficients $c_{i,j}$ will be under-determined and so it will not have a unique solution. The ambiguity in the coefficients may result in wildly different models that all fit the given data exactly, but fail to produce useful predictions for new trajectories — as is the goal for any reduced-order model. Several types of regularization and fitting strategies can be used to remove this ambiguity and obtain models with superior predictive performance. If one assumes that only a few terms from the dictionary contribute to the time derivative of each POD coefficient, then it is possible to employ sparsity promoting techniques such as Sparse Identification of Nonlinear Dynamics (SINDy) [36] to uncover these active coefficients. Sparse coefficients can also improve the physical interpretability of the resulting model. In fact, the POD-SINDy technique has been used to identify an accurate three-dimensional reduced-order model of the periodic shedding dynamics in the wake of a cylinder [36]. A host of other sparse approximation techniques are also available, including ℓ^1 penalization methods [81] including the Least Absolute Shrinkage and Selection Operator (LASSO) [262], iteratively reweighted optimization techniques such as those described in [58, 77, 51], and greedy selection methods like Orthogonal Matching Pursuit (OMP) [198, 61]. The advantage of

these sparsity-promoting techniques is that they all can guarantee exact or near-exact recovery of the active coefficients using fewer data than are needed to uniquely solve the least squares problem. Guidelines for dictionary selection and data sampling that lead to exact recovery of sparse governing equations can be found in [239]. The drawback is that the dynamics of each coefficient must actually have a sparse, or approximately sparse representation in the chosen dictionary. An alternative approach is to approximate the dynamics of POD coefficients using a recurrent neural network as in [275, 272]. However, the resulting models are more difficult to interpret than models based on pre-defined dictionaries.

3.2 The need for reduction onto nonlinear manifolds

In systems like advection-dominated fluid flows, the observed coherent structures may translate through space, making them difficult or impossible to represent using a low-dimensional linear superposition of fixed spatial modes as in POD or PCA [191, 150]. In fact, when the covariance operator for POD is translationally homogeneous, as might be the case for a system with translational symmetry, then the resulting POD modes are simply the Fourier modes along the direction of translation [250, 114]. To see why such modes don't tell us anything useful about the coherent structures, we provide the following:

Example 3.2.1. *Consider a system that generates spatial profiles on the circle $x_t : S^1 \to \mathbb{R}$ that are all phase-shifted copies of the same profile $x_t(e^{i\theta}) = \phi(e^{i(\theta+h(t))})$. Assume that the phases $e^{ih(t_k)}$ at sample times $\{t_k\}_{k=1}^{\infty}$ are distributed uniformly on S^1 in the sense that for any continuous $f : S^1 \to \mathbb{R}$, we have*

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} f(e^{ih(t_k)}) = \int_0^{2\pi} f(e^{i\theta})\, d\theta. \tag{3.23}$$

Then the POD covariance operator $R : L^2(S^1) \to L^2(S^1)$ is given by

$$\begin{aligned}(Rv)(e^{i\theta}) &= \int_0^{2\pi} \left(\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} x_{t_k}(e^{i\theta}) x_{t_k}(e^{i\omega}) \right) v(e^{i\omega})\, d\omega \\ &= \int_0^{2\pi} \underbrace{\left(\int_0^{2\pi} \phi(e^{i(\theta-\omega+\alpha)}) \phi(e^{i(\alpha)})\, d\alpha \right)}_{\rho(e^{i(\theta-\omega)})} v(e^{i\omega})\, d\omega,\end{aligned} \tag{3.24}$$

which has eigenfunctions $u_k(e^{i\theta}) = e^{ik\theta}$ for every integer k since

$$(Ru_k)(e^{i\theta}) = \int_0^{2\pi} \rho(e^{i(\theta-\omega)}) e^{ik\omega}\, d\omega = \left(\int_0^{2\pi} \rho(e^{i\omega}) e^{-ik\omega}\, d\omega \right) e^{ik\theta}. \tag{3.25}$$

The POD modes u_k are simply the Fourier modes on the circle and their variances or energies are given by the Fourier coefficients of ρ. The Fourier coefficients can decay slowly if the profile ϕ has few continuous derivatives, i.e., when ϕ is jagged. Most importantly, the POD modes u_k give us no information whatsoever about the coherent structure ϕ, which simply translates around the circle. While there may be an infinite number of energetic POD modes, the states x_t are distinguished by their phases alone. Hence, these states lie on a one-dimensional loop in an infinite-dimensional state space of functions on the circle.

The fact that an infinite number of energetic POD modes might be needed to represent states that lie on or near low-dimensional underlying manifolds motivates the development of techniques for model reduction that exploit this low-dimensional nonlinear structure.

When the governing equations have continuous symmetries, it is possible to construct reduced-order models on the quotient space formed by the equivalence classes of states that are related by symmetric transformations [229]. In doing so, the original governing equations are decomposed into a "vertical" component aligned with the action of the symmetry group and a complementary "horizontal" component that gives rise to dynamics on the quotient space. For instance, in a system with translational symmetry, one can choose a template profile and model the dynamics on a slice consisting of those states that differ from the template along directions orthogonal to the action of the translation symmetry, effectively sliding solutions back into alignment with the template as they evolve. By removing the translational component of the dynamics, it often becomes possible to represent the reduced state using a low-dimensional superposition of POD modes [230]. However, many systems in engineering are not symmetric due to complicated geometry and boundary conditions. Another approach taken in [80, 8, 200, 203] is to construct different reduced-order models in different regions of state space by employing locallized POD bases.

In general, the coherent structures we wish to model are described by states lying near a low-dimensional underlying manifold. In cases where the underlying manifold is closely approximated by a low-dimensional subspace, as in diffusion-dominated problems, we may employ model reduction methods based on POD. On the other hand, for many advection-dominated phenomena of interest, the low-dimensional underlying manifold is curved in such a way that it departs from any low-dimensional subspace [191]. In such cases, any accurate reduced-order model based on linear projection will require a large number of superfluous state variables when compared to the dimension of the underlying manifold. Instead, we wish to construct a reduced-order model of the dynamics confined to the underlying curved manifold without appealing to a subspace. Advances in machine

learning over the last two decades have provided powerful tools for identifying low-dimensional nonlinear manifolds from data that have promising applications for reduced-order modeling. In turn, model reduction applications for large scale nonlinear dynamical systems like fluid flows place new demands on machine learning techniques, motivating the development of specialized learning methods and architectures.

3.2.1 Parametrizing manifolds using autoencoders

Autoencoders [112, 99] are a type of neural network architecture where mappings into and out of a low-dimensional latent space are learned from data. In particular, an autoencoder consists of an encoder function $\psi_e : \mathbb{R}^n \to \mathbb{R}^r$ that *encodes* states x into a low-dimensional representation $\psi_e(x)$ in the "latent space" $\mathbb{R}^r$ and a decoder function $\psi_d : \mathbb{R}^r \to \mathbb{R}^n$ that reconstructs the states as closely as possible. In general, the encoder and decoder are parametrized nonlinear functions $x \mapsto \psi_e(x;\theta)$ and $z \mapsto \psi_d(z;\theta)$ with defining parameters θ. The encoder and decoder are usually parametrized using neural networks composed of layers as shown in Figure 3.1. Here, the lth layer is a parametrized linear map determined by weight matrices $W^{(l)}(\theta)$ and bias vectors $b^{(l)}(\theta)$ together with a nonlinear activation function σ that acts element-wise as in

$$\psi_d(z) = \psi_d^{(L)} \circ \cdots \circ \psi_d^{(1)}(z), \qquad \text{where} \qquad \psi_d^{(l)}(z^{(l-1)}) = \sigma(W_d^{(l)}(\theta)z^{l-1} + b_d^{(l)}(\theta)). \tag{3.26}$$

The parameters θ defining the encoder and decoder maps are usually optimized according to a reconstruction objective such as mean square reconstruction error

$$\underset{\theta}{\text{minimize}} \;\; \frac{1}{m}\sum_{j=1}^{m} \|x_j - \psi_d(\psi_e(x_j;\theta);\theta)\|^2, \tag{3.27}$$

where $\{x_j\}_{j=1}^m$ is a data set.

We have already encountered a simple type of autoencoder where the encoder and decoder are the linear maps found by POD, that is,

$$\psi_e(x) = U^* x \qquad \psi_d(z) = Uz. \tag{3.28}$$

Here, the decoder provides a coordinate parametrization of the subspace $\mathcal{M} = \operatorname{Range} U$ near which the states lie, while the encoder gives the coordinates of states orthogonally projected onto $\mathcal{M}$.

This suggests how autoencoders might be used for reduced-order modeling, namely in the same

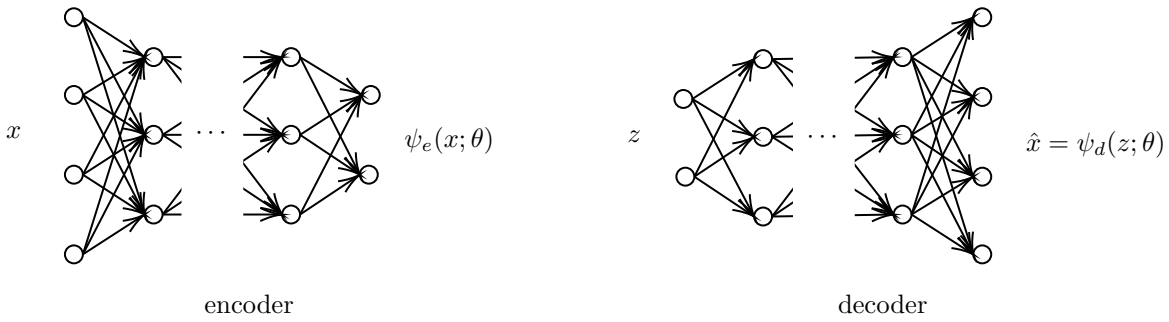


Figure 3.1: An autoencoder neural network is formed by a parametrized encoder function and a parametrized decoder function. These functions are made up of layers where linear maps defined by tunable weights are composed with element-wise nonlinearitities called activation functions. This setup is shown graphically where nodes represent the elements of each layer's output and arrows indicate the tunable linear dependencies on the outputs of the previous layer before applying the activation functions.

ways as POD, except using the possibly nonlinear encoder and decoder functions. The nonlinearity of the decoder function ψ_d, allows it to parametrize a nonlinear state space manifold, $\mathcal{M}_d$, near which the states of the system lie. Consequently, it is possible to project the dynamics onto this manifold as suggested by K. Lee and K. T. Carlberg in [150]. In [150], a nonlinear Galerkin reduced-order model is constructed by approximating the state $\hat{x} \in \mathcal{M}_d$. The time derivative in the tangent space of $\mathcal{M}_d$ is

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\hat{x} = \underset{w \in T_{\hat{x}}\mathcal{M}_d}{\operatorname{argmin}} \; \|f(\hat{x}, u) - w\| = P_{T_{\hat{x}}\mathcal{M}_d} f(\hat{x}, u), \tag{3.29}$$

where $P_{T_{\hat{x}}\mathcal{M}_d}$ is the orthogonal projection onto the tangent space. Assuming that $\mathrm{D}\,\psi_d(z)$ is injective and recalling that $\hat{x} = \psi_d(z)$ for some low-dimensional latent state $z \in \mathbb{R}^r$, the above model yields dynamics for this latent state given by

$$\boxed{\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,z = \underset{v \in \mathbb{R}^r}{\operatorname{argmin}} \; \|f(\psi_d(z), u) - \mathrm{D}\,\psi_d(z)v\| = (\mathrm{D}\,\psi_d(z)^*\,\mathrm{D}\,\psi_d(z))^{-1}\,\mathrm{D}\,\psi_d(z)^* f(\psi_d(z), u),} \tag{3.30}$$

where the orthogonal projection onto the tangent space of $\mathcal{M}$ at $\hat{x} = \psi_d(z)$ is given by

$$P_{T_{\hat{x}}\mathcal{M}_d} = \mathrm{D}\,\psi_d(z)\,(\mathrm{D}\,\psi_d(z)^*\,\mathrm{D}\,\psi_d(z))^{-1}\,\mathrm{D}\,\psi_d(z)^*. \tag{3.31}$$

The error of such a model may be crudely analyzed in the same way as the model based on linear Galerkin projection by employing the Grönwall inequality stated in Lemma 3.1.1.

3.2.2 Models based on low-dimensional embeddings

One may also construct reduced-order models based on the embedding provided by an autoencoder's encoder function or by the analogous function provided by a variety of other manifold learning techniques. Other manifold learning techniques that provide embedding functions analogous to the encoder ψ_e include spectral methods like kernel principal component analysis (KPCA) [244], Isomap [260], Laplacian eigenmaps [15], and diffusion maps [68, 63], as well as locally linear embedding (LLE) [224] and secant-avoidance projection methods [31, 111, 254]. If the states lie near a submanifold $\mathcal{M}$ of the state space $\mathcal{X}$, then ψ_e provides an embedding of $\mathcal{M}$ when the restriction of ψ_e to $\mathcal{M}$ is injective, has injective derivative $\mathrm{D}\,\psi_e(x)$ on $T_x\mathcal{M}$ for every $x \in \mathcal{M}$, and is proper. The technical condition that $\psi_e|_{\mathcal{M}}$ is proper says that the preimage under $\psi_e|_{\mathcal{M}}$ of any compact set is compact. This essentially means that $\psi_e|_{\mathcal{M}}$ does not map arbitrarily far away points to nearby latent states. Put together, these conditions imply that the embedded set $\psi_e(\mathcal{M})$ is a submanifold of the latent space $\mathbb{R}^r$. Moreover, there is a smooth function that reconstructs the original states in $\mathcal{M}$ from those in $\psi_e(\mathcal{M})$, of which the decoder ψ_d is an approximation. Hence, from an intuitive perspective, the training process for an autoencoder seeks to make ψ_e an embedding of an underlying manifold near which the data $\{x_j\}$ lie. When using other manifold learning techniques that do not automatically provide a reconstruction function (e.g., spectral methods and secant-avoidance projection), one may learn an approximate decoder ψ_d by employing a variety of regression techniques such as the one described in [121].

To build a reduced-order model based on an encoder, there are two options: either employ a nonlinear Galerkin method utilizing the reconstruction function, or learn the equations governing the dynamics of the embedded states directly from data. In the nonlinear Galkerin approach described by E. Chiavazzo et al. [63], one approximates the dynamics of the embedded states $z = \psi_e(x)$, $x \in \mathcal{M}$ according to

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, z = D\psi_e(\hat{x}) f(\hat{x}, u), \tag{3.32}$$

where the reconstructed state is provided by the decoder $\hat{x} = \psi_d(z)$. In [63], various extensions of the diffusion maps embedding operator are used to construct an encoder that projects nearby states onto the learned manifold. However, a drawback of this approach is that it may be very computationally expensive to evolve Eq. 3.32 due to the need to reconstruct the full state $\hat{x} = \psi_d(z)$ and evaluate the full-order model dynamics $f(\hat{x}, u)$ at each time step.

The alternative approach suggested by D. S. Broomhead and M. J. Kirby in [31] is to construct a data-driven approximation of the map $\tilde{f}$ defined by $D\psi_e(x) f(x, u) = \tilde{f}(\psi_e(x), u)$ for every $x \in \mathcal{M}$

and input u. Such a map $\tilde{f}$ is guaranteed to exist if ψ_e is an embedding of $\mathcal{M}$. This map can be approximated from data using regression techniques like SINDy [36], neural networks [220, 221], or by radial basis functions as in [31] and [121]. In either case, one uses data consisting of pairs states x_j, inputs u_j, and resulting time derivatives $\dot{x}_j = f(x_j, u_j)$ and constructs the embedded states and time derivatives in the latent space according to $z_j = \psi_e(x_j)$ and $\dot{z}_j = D\psi_e(x_j)\dot{x}_j$. A suitable map $\tilde{f}$ is then found by regression so that each $\dot{z}_j$ is approximated closely by $\tilde{f}(z_j, u_j)$. Using the learned map $\tilde{f}$, we may approximate the dynamics of the latent variables governed by Eq. 3.32 using

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, z = \tilde{f}(z, u). \tag{3.33}$$

The dynamics of the embedded latent states can also be modeled using various types of recurrent neural networks as in [155, 271]. In order to predict the dynamics of the original full-order model we still must reconstruct states on $\mathcal{M}$ from latent variables z using a suitable decoder.

3.2.3 The surprising utility of linear embeddings

While a variety of methods provide nonlinear encoder functions ψ_e, if the goal is merely to embed a compact state space manifold $\mathcal{M}$, one often need not look beyond linear encoders. This is due to Whitney's first embedding theorem [277], which says that any compact k-dimensional submanifold of $\mathbb{R}^n$ may be embedded in $\mathbb{R}^{2k+1}$ by a linear mapping. The proof of Whitney's theorem involves constructing a linear projection with a nullspace that does not contain any secant vector between distinct points of $\mathcal{M}$. If every secant has a nonzero projection then the projection map is one-to-one on $\mathcal{M}$. The degree to which the projection map preserves distances can be measured by the ratios of secant lengths to their lengths after the projection is applied. This idea motivates the Secant-Avoidance Projection (SAP) method introduced by D. S. Broomhead and M. J. Kirby in [31], whereby an orthogonal projection subspace is optimized to avoid zeroing out secants between data points sampled from $\mathcal{M}$.

If one does not care about the quality of the embedding as measured by distance preservation as in SAP, then the use of Sard's theorem in the proof of Whitney's theorem indicates that linear mappings that provide embeddings can be chosen essentially at random [106]. Distance preservation between data points can be achieved with high probability by random projections of sufficiently large dimension thanks to the Johnson-Lindenstrauss lemma [123, 270]. For any collection of m points the dimension of the Johnson-Lindenstrauss embedding with distance distortion factor bounded by $(1 \pm \epsilon)$ grows according to $\mathcal{O}(\varepsilon^{-2} \log m)$ [270]. Moreover, the logarithmic growth in the embedding

dimension for random projections can be removed when the data actually live on a low-dimensional manifold, in which case the dimension depends on the manifold's curvature [12, 65]. The use of random projections for producing cheap embeddings of enormous data sets has become a useful and widespread tool in machine learning and data science [133, 25, 3]. While the Johnson-Lindenstrauss embedding dimension is tight in the sense that random projections provide the minimum possible embedding dimension up to a constant factor, the constant can be large in practice. With increasing embedding dimension, it becomes harder to learn the reduced-order model dynamics $\tilde{f}$ from limited data. Hence, for model reduction purposes with limited data, it is advantageous to employ an optimization approach like SAP [31]. The following example demonstrates how a linear encoder can provide an embedding of a highly nonlinear state space manifold.

Example 3.2.2. *Consider a similar setup as in Example 3.2.1, where states $x_s \in L^2([-2,2])$ are given by shifted copies $x_s(\xi) = \phi(\xi - s)$ of a nonzero, nonnegative, continuously differentiable "bump" function ϕ with support in the interval $[-1,1]$ and $s \in (-1,1)$. Hence, these states live on a one-dimensional manifold*

$$\mathcal{M} = \left\{ x \in L^2([-2,2]) \ : \ x(\xi) = \phi(\xi - s), \ s \in (-1,1) \right\} \tag{3.34}$$

in the infinite-dimensional state space $L^2([-2,2])$. Define the encoder to be any one-dimensional linear map of the form

$$\psi_e(x) = \langle w, \ x \rangle_{L^2([-2,2])} = \int_{-2}^{2} x(\xi) w(\xi) \, d\xi, \tag{3.35}$$

where $w \in C^1([-2,2])$ is a continuously differentiable weight function with positive first derivative. The encoded coordinate for states $x_s \in \mathcal{M}$ are given by

$$z(s) = \psi_e(x_s) = \int_{-2}^{2} \phi(\xi - s) w(\xi) \, d\xi = \int_{-1}^{1} \phi(\xi) w(\xi + s) \, d\xi, \tag{3.36}$$

which is seen to be a continuously differentiable function with positive derivative, bounded above and below by constants,

$$0 < \min_{\eta \in [-2,2]} w'(\eta) \int_{-1}^{1} \phi(\xi) \, d\xi \leq z'(s) = \int_{-1}^{1} \phi(\xi) w'(\xi + s) \, d\xi \leq \max_{\eta \in [-2,2]} w'(\eta) \int_{-1}^{1} \phi(\xi) \, d\xi, \tag{3.37}$$

for $s \in (-1,1)$. Hence, ψ_e and its derivative are injective and the inverse of ψ_e on its range is also differentiable. Therefore, the encoder provides a one-dimensional linear embedding of $\mathcal{M}$ into $\mathbb{R}^1$. This is remarkable because $\mathcal{M}$ may not reside in any finite-dimensional subspace due to the sliding

support of x_s.

3.3 Capturing sensitivity via "dynamics-aware" learning

Unfortunately, reduced-order models of the dynamics on a manifold that closely approximates the observed states of a system may still be inaccurate when the dynamics are sensitive to low-energy features of the state. This problem was observed in [69], where it was found that POD does not always provide an optimal basis for modeling the dynamics. Similarly, [102] finds that an approximate inertial manifold based on the least dissipative modes fails to capture dynamically important small-scale features in a Rayleigh-Bénard convection system.

The manifold learning-based techniques described above in Section 3.2 are capable of identifying manifolds that represent the most energetic coherent structures of the observed states. For instance, autoencoders are usually trained to achieve the highest possible reconstruction accuracy by minimize the energy of the neglected features of the state according to Eq. 3.27. Projecting the states onto these manifolds necessarily removes low-energy features of the state that do not significantly contribute to reconstruction accuracy. The problem with this approach is that low-energy features may still have significant effects on the dynamics observed at later times. Thus, removing all of the low-energy features can result in poor predictive performance, even though states may be very accurately represented on the learned manifold. This problem is not merely theoretical since low-energy features of the state are known to play an important role in driving shear flow instabilities [264, 242, 114]. Shear flows are extremely common, occurring whenever two layers of fluid slide past each other at different speeds, causing shear between the layers. In such situations, a small disturbance introduced at an upstream location can grow while being carried with the flow, resulting in large amplitude structures appearing downstream. Modeling the dynamics on a manifold that represents the most energetic coherent structures appearing downstream is likely to ignore the low-energy features appearing upstream that cause the energetic structures to appear in the first place. We therefore argue that model reduction techniques for these types of systems must be "aware" of the dynamics in order to identify and capture the important low-energy features, while neglecting the unimportant ones.

3.3.1 Non-normality and sensitivity of linear dynamics

In linear dynamics, low-energy states can drive energetic responses due to non-normality of the governing linear operator. An operator $A : \mathcal{X} \to \mathcal{X}$ on a Hilbert space $\mathcal{X}$ is normal when $A^*A = AA^*$.

When A is also compact — as is always true in finite dimensions — the spectral theorem says that $\mathcal{X}$ admits an orthonormal basis $\{e_j\}$ of eigenvectors of A with eigenvalues $\{\lambda_j\}$. Consequently, the operator norm of A, measuring how much A can amplify the length of a vector on which it acts, is given by the eigenvalue with largest magnitude

$$\|A\|_{\text{op}} = \sup_{\substack{x \in \mathcal{X}: \\ \|x\|=1}} \|Ax\| = \max_j |\lambda_j|. \tag{3.38}$$

Moreover, this amplification occurs along the constant direction of the corresponding eigenvector $e_{\max}$ since $Ae_{\max} = \|A\|_{\text{op}} e_{\max}$. In this case, the most energetic direction $e_{\max}$ influences itself through the action of A since the most amplified direction is an eigenvector.

On the other hand, when A is not a normal operator, it may be most sensitive along directions that are not eigenvectors. An input along a sensitive direction can then result in a large output along a different direction. As an illustration, an operator on $\mathbb{R}^3$ defined by a matrix of the form

$$A = \begin{bmatrix} \lambda_1 & 0 & b \\ 0 & \lambda_2 & b \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{3.39}$$

is considered in [225, 114]. This matrix has eigenvalues λ_1, λ_2 and λ_3, while its operator norm is at least as large as its last column

$$\|A\|_{\text{op}} \geq \sqrt{2b^2 + \lambda_3^2} \geq \sqrt{2}|b|. \tag{3.40}$$

Thus, we can make the eigenvalues of A whatever we want, while making the operator norm of A large by choosing $|b|$ large. In this case, A is sensitive along the direction $v = (0, 0, 1)$ while producing arbitrarily large output along a nearly orthogonal direction $Av = (b, b, \lambda_3)$.

When A describes the dynamics of a system, for instance

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} &= \begin{bmatrix} -1 & 0 & 100 \\ 0 & -2 & 100 \\ 0 & 0 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u \\ y &= x_1 + x_2 + x_3 \end{aligned} \tag{3.41}$$

as in [114], the third state x_3 remains small while x_1 and x_2 experience large transient growth with

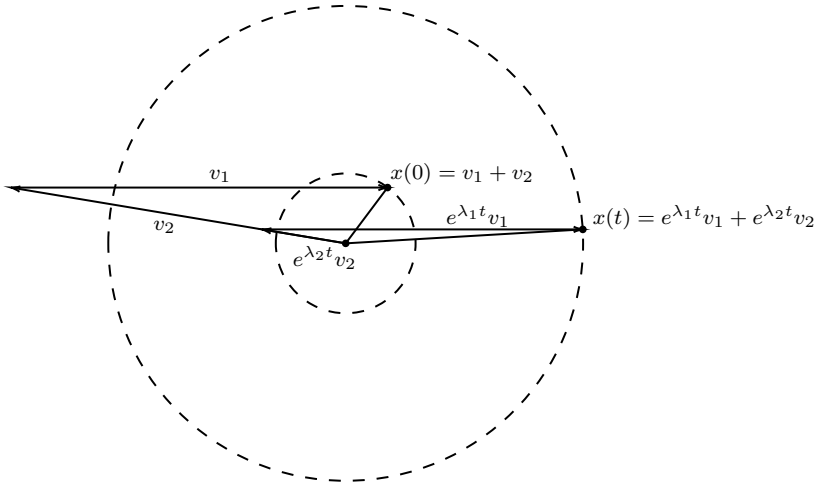


Figure 3.2: Two closely aligned eigenvectors v_1 and v_2 of a stable linear system can give rise to transient growth when their corresponding eigenvalues λ_1 and λ_2 differ. Starting at the initial state $x(0) = v_1 + v_2$, the magnitude of $x(t)$ experiences transient growth because $e^{\lambda_2 t}$ decays more quickly than $e^{\lambda_1 t}$. Moreover, the growth occurs along a direction that is not aligned with the initial state $x(0)$. After a long time, $x(t) \to 0$ because its coefficients decay exponentially.

slow decay and account for most of the state's energy. Hence, one might examine data drawn from impulse responses of this system as in [114] and find that the two-dimensional projection subspace obtained by POD captures most of the system's energy by almost completely neglecting the third state. However, the third state is important for the dynamics of this system because it drives the transient growth of x_1 and x_2. Consequently, two-dimensional POD-based reduced-order models of this system yield predictions that are extremely poor and do not exhibit any transient growth [114]. The geometric mechanism responsible for transient growth in non-normal systems is illustrated in Figure 3.2.

The operators obtained by linearizing shear flows about steady solutions are often non-normal and result in similar transient growth phenomena due to selective amplification of low-energy flow features [264, 242]. In fact, the transient growth due to non-normality can be large enough to drive the state away from the region of validity of the linearization even when the linearized dynamics are stable. This is precisely the mechanism of instability in pipe flows, which always become turbulent at sufficiently high Reynolds numbers, yet paradoxically have linearly stable steady-state velocity profiles [264, 242]. Even in the turbulent regime, the coherent structures one observes are often due to selective amplification of other low-energy features by the linearized operator about the turbulent mean flow [174].

Non-normality does not pose a problem for model reduction of linear systems by more sophisti-

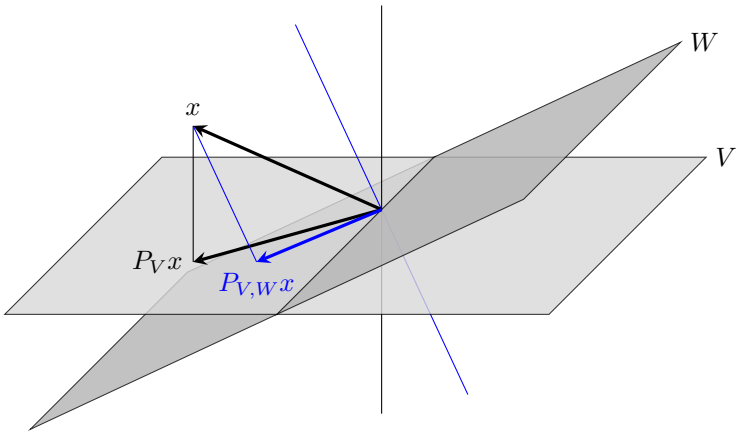


Figure 3.3: We illustrate the geometry of orthogonal and oblique projections of a point $x \in \mathbb{R}^3$ onto a two-dimensional subspace V. The direction $x - P_V x$ of the orthogonal projection is orthogonal to the subspace V, whereas the direction $x - P_{V,W} x$ of the oblique projection is othogonal to another two-dimensional subspace W. This figure is reproduced from C. W. Rowley and S. T.M. Dawson [228] with permission from the authors.

cated techniques like balanced truncation and $\mathcal{H}_2$ optimal model reduction. When the underlying system is non-normal, these techniques yield reduced-order models based on non-orthogonal, i.e., oblique projections of the governing equations. For reference, an orthogonal projection $P_V : \mathcal{X} \to \mathcal{X}$ onto an r-dimensional subspace V is defined by the orthogonality of the projection direction $x - P_V x$ with the subspace V that is, by the relation

$$P_V x \in V \quad \text{such that} \quad \langle v,\ x - P_V x \rangle = 0 \quad \forall v \in V. \tag{3.42}$$

On the other hand, an oblique projection $P_{V,W} : \mathcal{X} \to \mathcal{X}$ onto an r-dimensional subspace V is defined by letting its projection direction $x - P_{V,W} x$ be orthogonal to another r-dimensional subspace W, that is, by

$$P_{V,W} x \in V \quad \text{such that} \quad \langle w,\ x - P_{V,W} x \rangle = 0 \quad \forall w \in W. \tag{3.43}$$

We illustrate the geometry of orthogonal and oblique projections in Figure 3.3. Furthermore, any rank-r idempotent operator $P : \mathcal{X} \to \mathcal{X}$, i.e., $P^2 = P$, is an oblique projection operator described as above by the two r-dimensional subspaces $V = \text{Range}(P)$ and $W = \text{Range}(P^*) = \text{Null}(P)^\perp$. Oblique projections enable the resulting reduced-order models to capture the most energetic components of the system's response in addition to any low-energy features that are important for the dynamics. For instance, a highly accurate two-dimensional reduced-order model of Eq. 3.41 may be obtained by an oblique projection computed using balanced truncation as in [114]. In contrast, a two-dimensional

model based on POD and orthogonal projection performs poorly. Accurate models of linearized channel flow, which exhibits transient energy growth due to non-normality, have also been obtained using oblique projection methods [119]. Here, the oblique projections were identified by Balanced Proper Orthogonal Decomposition (BPOD) [225] — a computationally efficient approximation of balanced truncation.

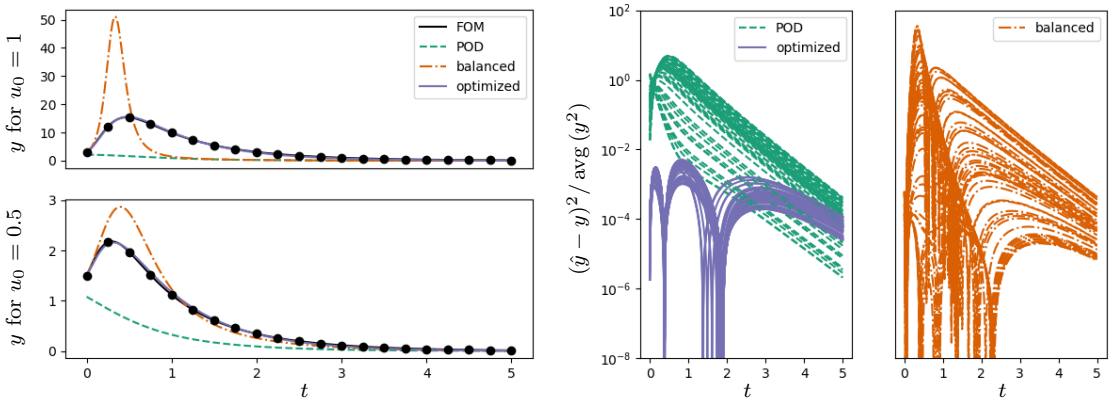
Constructing a reduced-order model by applying an oblique linear projection operator to a nonlinear system is called Petrov-Galerkin projection. In some cases, accurate models of nonlinear fluid flows have been obtained by Petrov-Galerkin projection of the governing equations onto subspaces identified by linear model reduction techniques applied at a nearby equilibrium [13, 6, 118, 120]. However, as we mentioned earlier, non-normality of the linearized dynamics can cause the state to depart from the region of validity where reduced-order models based on linearization or projection subspaces obtained from linearization are valid. In other cases, the states we wish to model may reside near a more complicated attractor that is too far away from any equilibrium to employ projections obtained from linearized dynamics.

3.3.2 Nonlinear sensitivity to low-energy features and optimizing oblique projections using trajectories

When the state is far from an equilibrium point, sensitivity to low-energy features may also be due to nonlinear interactions, which are considerably more difficult to model. For instance, in [192]** we consider a nonlinear system

$$\begin{aligned} \dot{x}_1 &= -x_1 + 15x_1x_3 + u \\ \dot{x}_2 &= -2x_2 + 15x_2x_3 + u \\ \dot{x}_3 &= -5x_3 + u \\ y &= x_1 + x_2 + x_3, \end{aligned} \tag{3.44}$$

that exhibits transient growth due to its quadratic interactions with the low-energy feature x_3. Two trajectories of this system generated from impulse responses with magnitudes $u_0 = 0.5$ and 1.0 are shown in Figure 3.4a along with the predicted trajectories of various two-dimensional projection-based reduced-order models. The normalized prediction errors for 50 impulse response trajectories with magnitudes drawn uniformly at random from the interval $[0, 1]$ are shown in Figure 3.4b. As in the case of the non-normal linear system in Eq. 3.41, the two-dimensional POD-Galerkin reduced-order model of Eq. 3.44 performs poorly since it neglects the low-energy feature x_3. More



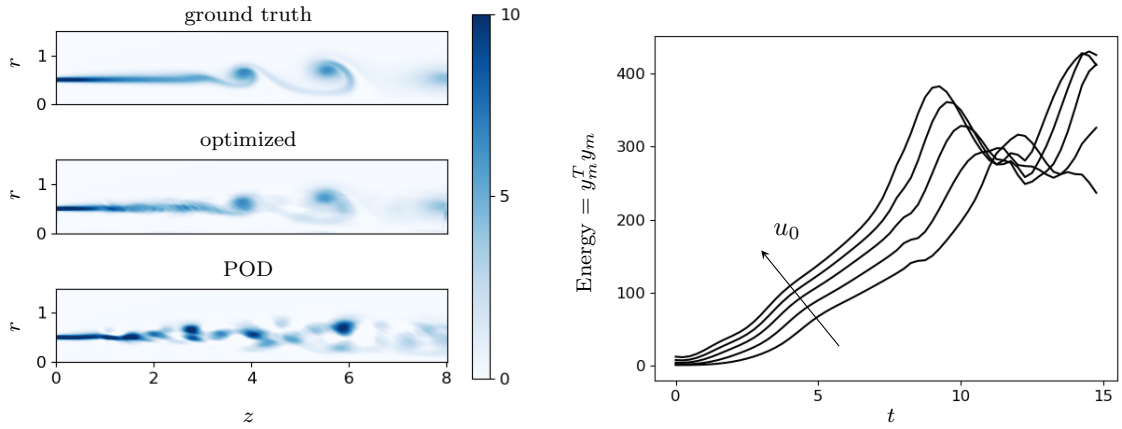
(a) training trajectories

(b) error on testing trajectories

Figure 3.4: In panel (a), we show the outputs generated by the full-order model Eq. 3.44 and the two-dimensional reduced-order models found by POD Galerkin projection, balanced truncation, and our optimization approach in response to impulses with magnitudes $u_0 = 0.5$ and $u_0 = 1$ at $t = 0$. The sample points used to construct the optimization objective objective for the projection operator are shown as black dots. In panel (b), we show the normalized square errors of the reduced-order model predictions in response to 50 impulses at $t = 0$ with magnitudes u_0 were drawn uniformly at random from the interval $[0, 1]$.

interestingly, a two-dimensional reduced-order model obtained by Petrov-Galerkin projection of Eq. 3.44 using subspaces identified by balanced truncation of the linearized dynamics about the equilibrium at the origin also performs poorly. This is because the transient growth exhibited by Eq. 3.44 is due to quadratic nonlinearities, rather than non-normality of the linearized dynamics, which are stable and normal.

In [192]**, we also consider an axisymmetric jet flow governed by the incompressible Navier-Stokes equations, possessing a similar kind of nonlinear sensitivity as our toy model Eq. 3.44. A snapshot of a flowfield formed after an impulse was introduced in a small upstream region centered at axial distance 1.0 and radius 0.5 is shown in Figure 3.5a. In this figure we see energetic vortex structures appearing downstream, while the initial disturbance was tiny and located upstream. These structures are the result of a nonlinear process where small disturbances grow while being convected downstream by the flow, drawing energy from the radial velocity gradient as they travel. Time histories of the disturbance energy along such trajectories with different impulse magnitudes are shown in Figure 3.5b, revealing high amounts of energy growth and nonlinearity. If the system were linear, the time histories of the energy would look like scaled copies increasing with u_0 as a result of linear superposition. As we would expect, the dynamics experience an initial period of exponential growth consistent with linear dynamics. However, after the energy exceeds ~ 50, nonlinear mechanisms begin to have a significant affect.



(a) vorticity snapshot

(b) output energy

Figure 3.5: In panel (a) we show a snapshot of the vorticity in the jet flow at the final time $t = 14.75$ from an impulse response trajectory with input magnitude $u_0 = 0.9$. We also show the prediction of a 50 dimensional POD-Galerkin reduced-order model as well as a 50 dimensional Petrov-Galerkin model where the projection subspaces were optimize to correctly predict trajectories in a separate training data set. In panel (b) we show the output energies along trajectories with varying impulse response magnitudes $u_0 = 0.15, 0.3, 0.5, 0.7$, and 0.9.

As was the case with Eq. 3.44, POD-Galerkin and BPOD-Petrov-Galerkin reduced-order models of this system perform very poorly. In fact, the BPOD-based Petrov-Galerkin models we examined had solutions that blew up in finite time. The prediction of a 50-dimensional POD-based Galerkin reduced-order model shown in Figure 3.5a is very poor despite the fact that the 50-dimensional POD subspaces captures 99.6% of the system's energy on both training and testing data sets. Because the most energetic features are supported downstream, POD fails to capture the low-energy upstream features that drive the response. When examining the BPOD-based model, we found that the subspace V in which the solution was to be represented corresponded to structures supported downstream, while the subspace W defining the projection direction corresponded to structures supported far upstream. Consequently BPOD was correctly predicting the upstream sensitivity of the flow, but ignoring the importance of convection.

Examples like Eq. 3.44 and the jet flow motivate the development of model reduction techniques that are aware of the nonlinear contributions to the dynamics in addition to the linear contributions. In the absence of closed-form expressions for the solutions of nonlinear systems, such an approach can be carried out using data collected from the system. *This leads us to develop an optimization technique for the projection subspaces V and W that aims to minimize the error between reduced-order model predictions and a collection of sampled trajectories of the full-order model.* Using this technique, we can construct accurate Petrov-Galerkin reduced-order models of Eq. 3.44 and of the jet flow. The predictions of the optimized reduced-order models in Figure 3.4 and Figure 3.5a show

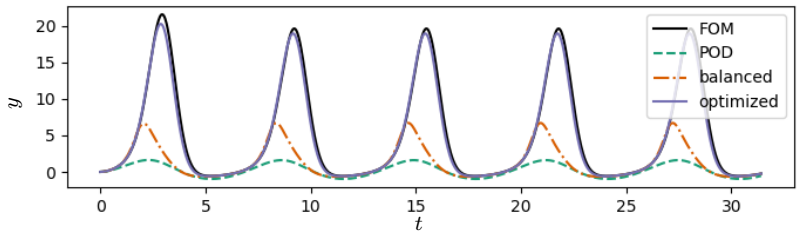


Figure 3.6: We show the responses of Eq. 3.44 and the reduced-order models to input $u(t) = \sin(t)$.

that the approach significantly outperforms other projection-based reduced-order models of the same dimension. The optimized projection-based model we trained on impulse-response trajectories of the toy model Eq. 3.44 can also accurately predict the response to other inputs such as the sinusoidal response shown in Figure 3.6. Again, the models based on projections found using POD and balanced truncation do not correctly capture the response to the sinusoidal input.

To set up the optimization problem for the subspaces V, W determining the projection operator $P_{V,W}$, we assume that the initial condition $x(t_0) = x_0$ and input signal u are known. The resulting Petrov-Galerkin reduced-order model

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\hat{x} &= P_{V,W} f(\hat{x}, u) \qquad \hat{x}(t_0) = P_{V,W} x_0 \\ \hat{y} &= g(\hat{x}) \end{aligned} \tag{3.45}$$

produces an output signal $\hat{y}(t; (V, W))$ that depends only on the choice of subspaces V and W defining the oblique projection operator $P_{V,W}$. In principle, if the initial condition were not known, then the output would also depend on the initial condition, which could be optimized alongside V and W. If we collect samples of the output $y_l = y(t_l)$ at times $t_0 < t_1 < \cdots < t_{L-1}$ along one or more trajectories from the full-order model, then we can compare the output of the reduced-order model to these data. In [192]** we identify the subspaces V, W by minimizing a cost function

$$\boxed{J(V, W) = \frac{1}{L}\sum_{l=0}^{L-1} \|\hat{y}(t_l; (V, W)) - y_l\|^2 + \gamma\rho(V, W),} \tag{3.46}$$

where the function ρ is included with a positive weight γ as a regularization to ensure that the chosen subspaces V, W define a valid oblique projection. Specifically, if $\Phi, \Psi \in \mathbb{R}^{n\times r}$ are matrices such that $V = \mathrm{Range}(\Phi)$ and $W = \mathrm{Range}(\Psi)$ then $\rho(V, W)$ is defined by

$$\rho(V, W) = -\log\left(\frac{\det(\Psi^*\Phi)^2}{\det(\Psi^*\Psi)\det(\Phi^*\Phi)}\right). \tag{3.47}$$

This function has the property that $\rho(V_k, W_k) \to +\infty$ when $\{V_k, W_k\}_{k=1}^{\infty}$ is any sequence of subspaces approaching the set subspaces where oblique projection operators cannot be defined. Furthermore, the minimum value of $\rho(V, W)$ is zero and this value is attained if and only if $V = W$, which corresponds to the case when $P_{V,W}$ is the orthogonal projection onto V. Using the regularization function defined by Eq. 3.47, we show that, under certain mild assumptions (see [192]**) minimizers of Eq. 3.46 always exist and correspond to valid oblique projection operators. The set of all r-dimensional subspaces of the n-dimensional state space $\mathcal{X} = (\mathbb{R}^n, \langle \cdot, \cdot \rangle)$ can be endowed with the structure of an $nr - r^2$ dimensional Riemannian manifold called the Grassmann manifold, denoted $\mathcal{G}_{n,r}$. In [192]**, we optimize the subspaces (V, W) over the product of Grassmann manifolds $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ using a geometric conjugate gradient algorithm [235, 1] with the gradient of Eq. 3.46 computed by an adjoint sensitivity method.

A key feature of the approach we describe in [192]** is the incorporation of dynamics into the optimization process for the projection subspaces. Namely, the projection subspaces are optimized not only to accurately encode and decode individual states, but also to yield models that can forecast trajectories of the system. To make accurate forecasts, the projection operator must learn to pick out features that influence the dynamics at later times, even if the features have low energy. The same principle may be applied to learning reduced-order models of dynamics on nonlinear manifolds from snapshots of the state along trajectories. However, in this case, it becomes computationally expensive to express the dynamics of the latent state since the nonlinear Galerkin projection will still require evaluating the full-order model. Therefore, approaches that seek reduced-order models on nonlinear manifolds tend to also learn the dynamics of the latent variables from data. They do not rely on the governing equations at all once the data is collected. In [194]** we train an autoencoder simultaneously with linearly recurrent dynamics of the latent state in order to make forecasts of sampled system trajectories. Consequently, the encoder and decoder discover a manifold that includes both the energetic features needed for accurate reconstruction, as well as any low-energy features that are needed to make accurate forecasts. A similar approach may also incorporate nonlinear latent state dynamics, as in [98], where a recurrent neural network is used. A variety of other related approaches learn linear [257, 164, 167, 294] and nonlinear [57] latent state dynamics and embedding simultaneously based on time derivative information, or based on delayed state snapshot pairs. Sophisticated techniques based on variational inference can also be used to construct nonlinear models of latent variables from nonlinear, possibly noisy observed quantities [95, 64, 142, 132]. However, as we will see in the next section, the purely data-driven approach requires an amount of data comparable to the state dimension in order to correctly determine the features that the system

is sensitive to.

3.3.3 Inadequacy of methods based solely on input-output data

Another key aspect of our approach in [192]** is that the original governing equations are leveraged to build the model. This not only provides an interpretation for the model in terms of Petrov-Galerkin projection, but also enables us to capture sensitive dependence on low-energy features. We optimize the projection subspaces based on gradient information provided by an adjoint-sensitivity method that incorporates the Jacobian of the governing equations at different points along trajectories. In contrast with purely data-driven approaches, the Jacobian of the full-order model encodes sensitivity information about dynamics, and may be responsible for the method's success in extracting dynamically significant low-energy features.

In particular, when the amount of data is low compared to the state dimension of the full-order model, it becomes difficult to infer from the input-output data alone what low-energy features the system is sensitive to. This is because, in a high-dimensional system where the amount of data is much smaller than the state dimension, there may be an enormous number of low-energy features that appear to be correlated with the energetic outputs. Yet if more data was collected, these correlations would turn out to be spurious. For instance, Example 3.3.1 shows that input-output data can be used to reliably estimate the range of a low-rank linear operator, but not the range of its adjoint. The fact that the range can be reliably estimated from the output of the operator acting on almost any collection of vectors follows from a rather useful technical result stated in Theorem 3.3.2 below. This result characterizes the typical transversal intersecting behavior one can expect for the ranges and null-spaces of two matrices. Here, the range of the operator reflects the energetic components of the output, while the range of the adjoint reflects the inputs that the map is most sensitive to. The same idea also applies to operators that merely have quickly decaying singular values, in which case it is possible to estimate the leading left singular vectors from the output of the operator acting on a small collection of random vectors [270]. If one may also act with the adjoint operator on a collection of random vectors, then it becomes possible to estimate the leading right singular vectors as well; this is precisely what is done in the randomized SVD algorithm described by N. Halko et al. [108]. It is also the basis for using data collected from the adjoint linear system to capture the most observable states via Balanced Proper Orthogonal Decomposition (BPOD) [225].

Example 3.3.1 (Inadequacy of Forward Data for Sensitivity Analysis)**.** *Suppose that* $A : \mathbb{R}^n \to \mathbb{R}^n$ *is a linear map with rank* $r < n$*. Let* $\{x_1, \ldots, x_m\} \subset \mathbb{R}^n$ *be a collection of linearly independent*

vectors arranged as columns of the matrix $X \in \mathbb{R}^{n\times m}$. For almost every choice of these vectors with $m \geq r$, the range of A is given by the span of $\{Ax_1, \ldots, Ax_m\}$. To see this, let $V \in \mathbb{R}^{n\times r}$ have columns spanning the range of A^*. Then Theorem 3.3.2 shows that almost every matrix $X_r \in \mathbb{R}^{n\times r}$ with respect to Lebesgue measure has $\det(V^*X_r) \neq 0$. *Consequently, AX_r has linearly independent columns. For if $AX_rv = 0$, then X_rv is orthogonal to the range of A^*, and so $V^*X_rv = 0$, implying that $v = 0$. Therefore, for almost any collection of vectors $\{x_1, \ldots, x_m\} \subset \mathbb{R}^n$ with $m \geq r$, the first r elements of $\{Ax_1, \ldots, Ax_m\}$ span the range of A.*

On the other hand, there are an infinite number of rank-r matrices $\tilde{A}$ that have the same input-output behavior as A for the inputs X when $m < n$, yet are sensitive to different features of the input due to having different co-ranges $\mathrm{Range}(\tilde{A}^*)$. *The range of any $n \times m$ matrix Y such that* $\det(Y^*X) \neq 0$ *may be selected to contain the range of $\tilde{A}^*$, with* $\mathrm{Range}(Y)$ *sometimes being referred to as a "learning subspace". With a choice for $Y \in \mathbb{R}^{n\times m}$, there is a unique $\tilde{A} \in \mathbb{R}^{n\times n}$ given by*

$$\tilde{A} = AX(Y^*X)^{-1}Y^* \tag{3.48}$$

for which $\tilde{A}X = AX$ and $\mathrm{Range}\,\tilde{A}^* \subset \mathrm{Range}\,Y$. *By Theorem 3.3.2, almost any matrix $Y \in \mathbb{R}^{n\times m}$ with respect to Lebesgue measure may be chosen to span the learning subspace. This leads to ambiguity in those $\tilde{A}$ which agree with A over the data X. In least-squares estimation, one takes $Y = X$, but this choice is arbitrary. We conclude that the action of A on a relatively small collection of vectors tells us a lot about the range of A, but we learn very little about the range of A^* until the number of vectors m is equal to the entire state dimension n.*

Theorem 3.3.2 (Typical intersections of ranges and null-spaces)**.** *If $T \in \mathbb{R}^{n\times r}$ has linearly independent columns, then*

$$\mathcal{S} = \{X \in \mathbb{R}^{n\times r} \ : \ \det(T^*X) \neq 0\} \tag{3.49}$$

is open, dense in $\mathbb{R}^{n\times r}$, and contains almost every $X \in \mathbb{R}^{n\times r}$ with respect to Lebesgue measure. Moreover, $\mathcal{S}$ has two open, connected components

$$\mathcal{S}_+ = \{X \in \mathbb{R}^{n\times r} \ : \ \det(T^*X) > 0\} \quad \textit{and} \quad \mathcal{S}_- = \{X \in \mathbb{R}^{n\times r} \ : \ \det(T^*X) < 0\}. \tag{3.50}$$

Proof. Openness follows from the fact that $\mathcal{S}$ is the pre-image of an open set under a continuous map. We note that the density of $\mathcal{S}$ is implied by the fact that $\mathcal{S}$ contains almost every element of $\mathbb{R}^{n\times r}$. However, we cannot help but provide a direct proof by constructing an arbitrarily small

perturbation of any matrix in $\mathbb{R}^{n\times r} \setminus \mathcal{S}$ that ends up in $\mathcal{S}$. We prove the fact that $\mathbb{R}^{n\times r} \setminus \mathcal{S}$ has Lebesgue measure zero in two very different ways. The most direct way to prove this claim is to observe that $\mathbb{R}^{n\times r} \setminus \mathcal{S}$ is the zero set of a non-constant polynomial on a real Euclidean space and so it has measure zero [53]. We also provide an analytic proof based on Lebesgue's density theorem (see Section 7.2 of W. Rudin [233]), which doesn't rely on special properties of polynomials. Finally, we prove that $\mathcal{S}_+$ and $\mathcal{S}_-$ are connected by constructing paths between arbitrary points of these sets. We give the detailed proofs in Appendix 3.A □

The problem described above also applies to more complicated mappings that one might learn from data using a neural network. If the weights describing the first layer in the network are given by the matrix A, then the inputs that the model is sensitive to are contained in the range of A^*. Yet the range of A^* cannot be reliably estimated from a data set that is smaller than the state dimension, as demonstrated by Example 3.3.1. This casts doubt on whether data driven methods relying only on forward time histories as in [194]** and [98, 257, 164, 167, 294, 95, 64, 142, 132] can properly capture the dynamics of systems with selective sensitivity to low-energy features. This poses a serious problem for making data-driven forecasts, because the learned model might fail to predict how sensitive the system will be to inputs that don't closely resemble the training data.

3.3.4 Building sensitivity into reduced-order models

One way to build the correct sensitivity mechanisms into the model is to incorporate the adjoint Jacobian of the original governing equations into the learning process. The approach we take when optimizing projection subspaces in [192]** presents one possible way to do this. In particular, when the "encoder" and "decoder" are used to construct a projection of governing equations, the adjoint Jacobian of the full-order model appears naturally in the expression for the gradient of the model error with respect to the parameters defining the encoder and decoder. On the other hand, when the encoder and decoder are nonlinear, a latent space model based on projecting the governing equations can still be computationally expensive to evolve in time. As we have seen, these considerations motivate techniques that also learn the latent space dynamics. Yet, by learning the latent space dynamics from data, one removes the direct dependence of the reduced order model on the full order model, and so the adjoint Jacobian of the full-order model disappears. If the model of the latent space dynamics is also to be learned from data, then one approach might be to optimize the model parameters using a cost function that compares both forward and linearized adjoint trajectories of the reduced and full-order models. This approach is analogous to computing a randomized SVD

from small amounts of data obtained by acting with the operator A as well as with its adjoint A^* as described by N. Halko et al. [108].

To be specific, suppose we have a full-order model

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, x &= f(x,u) \\ y &= g(x), \end{aligned} \tag{3.51}$$

and a reduced-order model

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, \hat{x} &= \hat{f}(\hat{x},u;\theta) \\ \hat{y} &= \hat{g}(\hat{x};\theta), \end{aligned} \tag{3.52}$$

with θ defining parameters we seek to optimize. We could begin by collecting forward trajectories $x(t), y(t)$ from the full-order model and $\hat{x}(t), \hat{y}(t)$ from the reduced-order in response to given input signals $u(t)$. Most available techniques seek to optimize the parameters θ defining the reduced-order model by minimizing the error between the output of the reduced-order model $\hat{y}(t)$ and the full-order model $y(t)$. However, as we have seen, doing so may yield a reduced-order model that does not correctly capture the sensitivity of the output to the input.

To capture this sensitivity, we can choose a collection of signals $\xi_j(t)$ in the output space and define functionals of the output signals according to

$$\Xi_j(u) = \langle \xi_j,\ y \rangle := \int_{t_0}^{t_f} \xi_j(t)^T y(t)\,\mathrm{d}\,t \qquad \text{and} \qquad \hat{\Xi}_j(u;\theta) = \langle \xi_j,\ \hat{y} \rangle. \tag{3.53}$$

The signals ξ might be chosen at random, or as the leading principal components of the observed output as in the output projection method for balanced POD described in [225]. Now we can seek parameters θ to match the sensitivity of the output to the input by matching the sensitivity of the reduced-order model $\nabla_u \hat{\Xi}_j(u;\theta)$ to the sensitivity of the full-order model $\nabla_u \Xi_j(u)$. The analogy with randomized SVD is clear, for if we let $\mathrm{D}_u\, y(u)$ and $\mathrm{D}_u\, \hat{y}(u;\theta)$ be the derivatives of the output signals with respect to the input signals, then we have

$$\boxed{\big(\mathrm{D}_u\, y(u)\big)^* \xi_j = \nabla_u\, \Xi_j(u) \qquad \text{and} \qquad \big(\mathrm{D}_u\, \hat{y}(u;\theta)\big)^* \xi_j = \nabla_u\, \hat{\Xi}_j(u;\theta),} \tag{3.54}$$

because $\delta\Xi_j = \langle \nabla_u\, \Xi_j(u),\ \delta u \rangle = \langle \xi_j,\ \mathrm{D}_u\, y(u)\delta u \rangle = \langle (\mathrm{D}_u\, y(u))^* \xi_j,\ \delta u \rangle$ holds for every input perturbation δu. Therefore, the gradients can be viewed as samples allowing us to approximate the

range of $(\mathrm{D}_u\, y(u))^*$, which describes the subspace of input signals that produce the most energetic output responses. The parameters of the reduced order model could then be chosen to minimize a cost function

$$\boxed{J(\theta) = \int_{t_0}^{t_f} \left\| \hat{y}(t;\theta) - y(t) \right\|^2 \mathrm{d}\, t + \sum_j \int_{t_0}^{t_f} \left\| \nabla_u\, \hat{\Xi}_j(u;\theta)(t) - \nabla_u\, \Xi_j(u)(t) \right\|^2 \mathrm{d}\, t} \tag{3.55}$$

that includes both the predictive accuracy of the model as well as its sensitivity.

An adjoint sensitivity method can be used to compute the desired gradients. In particular, if we define adjoint variables for the models satisfying

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}\, \mathrm{t}}\, \lambda_j &= \mathrm{D}_x\, f(x(t), u(t))^* \lambda_j + \mathrm{D}\, g(x(t))^* \xi_j(t), \qquad \lambda_j(t_f) = 0, \\ -\frac{\mathrm{d}}{\mathrm{d}\, \mathrm{t}}\, \hat{\lambda}_j &= \mathrm{D}_x\, \hat{f}(\hat{x}(t), u(t);\theta)^* \hat{\lambda}_j + \mathrm{D}\, \hat{g}(\hat{x}(t))^* \xi_j(t), \qquad \hat{\lambda}_j(t_f) = 0, \end{aligned} \tag{3.56}$$

then the gradients of Ξ and $\hat{\Xi}$ are given explicitly by

$$\nabla_u\, \Xi_j(u)(t) = \mathrm{D}_u\, f(x(t), u(t))^* \lambda_j(t) \quad \text{and} \quad \nabla_u\, \hat{\Xi}_j(u;\theta)(t) = \mathrm{D}_u\, \hat{f}(\hat{x}(t), u(t);\theta)^* \hat{\lambda}_j(t). \tag{3.57}$$

To my knowledge, optimizing the parameters of a reduced-order model based on a cost function that includes sensitivity information in the manner of Eq. 3.55 is new, yet it seems like a natural and promising direction for future work. One drawback will be the need to compute second derivatives of the reduced-order model functions $\hat{f}$ and $\hat{g}$ with respect to the states and inputs. This is not so difficult because it is only necessary to compute second derivatives of the reduced-order model, whose dimension is much smaller than the full-order model.

Another, simpler approach closely resembles balanced truncation [182] and Balanced Proper Orthogonal Decomposition (BPOD) [225]. We may describe the energetic states of the system using the covariance matrix

$$C_x = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} x(t) x(t)^* \,\mathrm{d}\, t, \tag{3.58}$$

which is only written for a single trajectory for the sake of simplicity, but could, in principle, be averaged over many trajectories. On the other hand, we recall that the adjoint variable λ_j defined in Eq. 3.56 provides the gradient Ξ_j with respect to an input v that enters the dynamics according to

$$\frac{\mathrm{d}}{\mathrm{d}\, \mathrm{t}}\, x = f(x, u) + v. \tag{3.59}$$

Therefore, the sensitivity of the output to state perturbations can be quantified using the adjoint covariance matrix

$$C_\lambda = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{t_f - t_0}\int_{t_0}^{t_f}\lambda_j(t)\lambda_j(t)^*\,\mathrm{d}\,t. \tag{3.60}$$

The covariance matrices C_x and C_λ are the analogues of the controlability and observability Grammians used for balanced truncation of linear systems. In particular, these covariance matrices transform in the same way as the Grammians under linear transformations $x = Tx'$ of the state:

$$C_{x'} = T^{-1}C_x T^{-*}, \qquad C_{\lambda'} = T^* C_\lambda T. \tag{3.61}$$

Consequently, it is possible to find a transformation T that simultaneously diagonalizes the covariance matrices such that

$$C_{x'} = C_{\lambda'} = \Sigma^2 = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2), \qquad \sigma_1 \geq \cdots \geq \sigma_n \geq 0. \tag{3.62}$$

In this new coordinate system, the leading state variables $[x']_1, [x']_2, \ldots$ are simultaneously the most energetic and produce the most energetic output responses when perturbed. Consequently, retaining these leading state variables in the transformed space and truncating the rest is likely to lead to an accurate reduced-order model of the system. Letting $C_x = XX^*$ and $C_\lambda = LL^*$ and computing a singular value decomposition

$$X^*L = U\Sigma V^* \tag{3.63}$$

yields the desired transformation

$$T = XU\Sigma^{-1/2}, \qquad T^{-*} = LV\Sigma^{-1/2}. \tag{3.64}$$

Similarly to BPOD [225], matrices X and L as needed above may be constructed from snapshots of the state and adjoint variables with appropriate quadrature weights $\{w_l\}$ according to

$$\begin{aligned} X &= \begin{bmatrix} \sqrt{w_1}x(t_0) & \cdots & \sqrt{w_L}x(t_f) \end{bmatrix}, \\ L &= \frac{1}{\sqrt{m}}\begin{bmatrix} \sqrt{w_1}\lambda_1(t_0) & \cdots & \sqrt{w_L}\lambda_1(t_f) & \cdots & \sqrt{w_1}\lambda_m(t_0) & \cdots & \sqrt{w_L}\lambda_m(t_f) \end{bmatrix}. \end{aligned} \tag{3.65}$$

To build a reduced-order model using the leading r coordinates, we may work with the rank-r

truncation of the above singular value decomposition and let

$$\Phi = XU_r\Sigma_r^{-1/2}, \qquad \Psi = LV_r\Sigma_r^{-1/2}. \tag{3.66}$$

The rank-r oblique projection operator $P = \Phi\Psi^*$ applied to the state x corresponds to transforming x into x', truncating the less energetic and sensitive variables $[x']_{r+1}, \ldots, [x']_n$, and then transforming back. Therefore, P might be a good candidate for building Petrov-Galerkin reduced-order models of the original system. Moreover, the transformed and reduced set of variables

$$z = ([x']_1, \ldots, [x']_r) = \Psi^* x \tag{3.67}$$

might be good candidates for building data-driven reduced-order models that capture the system's sensitivity to low-energy features. One may consider performing such a transformation as a pre-processing step akin to POD to reduced the state dimension before applying a variety of machine learning techniques to build models of the system. By reducing the dimension to include both the most sensitive and energetic states of the system, purely data-driven techniques based on these variables may be capable of robustly predicting the dynamics of the original system.

3.4 Learning nonlinear projections using autoencoders with invertible nonlinearities and biorthogonal weights

It would be highly advantageous to build reduced-order models based on nonlinear projection operators onto underlying curved state space manifolds. So far, in Section 3.2 we have seen why it is important to construct reduced-order models of dynamics confined to nonlinear manifolds. However, our solution to capture the sensitivity of the dynamics to certain low-energy features described in Section 3.3.2 and in [192]** was based on a linear projection. Since the dynamics were represented in a subspace, we needed to retain 50 state variables in the reduced-order model of the jet flow to resolve advecting vortical structures. On the other hand, the states of this flow that we considered actually live on a two-dimensional manifold with a global parametrization by time and the magnitude of the initial impulse. Consequently, it should be possible to find a reduced-order model with a much smaller state dimension by projecting the dynamics onto a nonlinear manifold. In order to find a suitable manifold and project onto it, we define a rich parametric class of nonlinear projection operators using autoencoders. The question remains as to how a projection should be defined in

order to capture the important low-energy features. Here, the encoder selects the features of the reduced-order model and the decoder parametrizes the underlying state space manifold.

The problem with using standard autoencoder architecures to define reduced-order models based on nonlinear projections is that in practice, these autoencoders do not actually yield projections. The defining property of a nonlinear operator $P : \mathcal{X} \to \mathcal{X}$ that makes it a projection is idempotence $P \circ P = P$, that is, repeatedly acting with P does not change states in the image of P. However, if one begins with a state x and uses an autoencoder to repeatedly encode and decode it, the process does not remain constant after the first iteration.

Smooth nonlinear projections are especially useful for reduced-order modeling. Theorem 3.4.1 shows that the image set of a projection with a constant-rank derivative is a smooth submanifold of the state space. Moreover, the preimage sets provides a foliation of the state space. The derivative of a projection P with rank r is an oblique projection onto the tangent space, allowing us to define reduced-order models with states evolving in the manifold $\mathcal{M} = \text{Image}(P)$ according to

$$\boxed{\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\hat{x} = \mathrm{D}\,P(\hat{x})f(\hat{x},u), \qquad \hat{x}(0) = P(x_0).} \tag{3.68}$$

Figure 3.7 illustrates the anatomy of nonlinear projections characterized by Theorem 3.4.1.

Theorem 3.4.1 (Constant rank projections define transversal submanifolds)**.** *Let $\mathcal{N}$ be a smooth, n-dimensional manifold and let $P : \mathcal{N} \to \mathcal{N}$ be a smooth projection with constant rank, that is, $P \circ P = P$ and $\operatorname{rank}(\mathrm{D}\,P(x)) = r$ is constant for every $x \in \mathcal{N}$. Then the image set*

$$\mathcal{M} = P(\mathcal{N}) = \{P(x) \ : \ x \in \mathcal{N}\}$$

is a smooth r-dimensional submanifold of $\mathcal{N}$. Furthermore, for every $x_0 \in \mathcal{M}$, the preimage set

$$P^{-1}(x_0) = \{x \in \mathcal{N} \ : \ P(x) = x_0\}$$

is a codimension-r submanifold of $\mathcal{N}$ that transversally intersects $\mathcal{M}$ at x_0, i.e., we have the direct-sum decomposition $T_{x_0}\mathcal{M} \oplus T_{x_0}P^{-1}(x_0) = T_{x_0}\mathcal{N}$. At any $x_0 \in \mathcal{M}$, the derivative $\mathrm{D}\,P(x_0) : T_{x_0}\mathcal{N} \to T_{x_0}\mathcal{N}$ is the linear oblique projection with range and nullspace

$$\text{Range}(\mathrm{D}\,P(x_0)) = T_{x_0}\mathcal{M} \qquad \textit{and} \qquad \text{Null}(\mathrm{D}\,P(x_0)) = T_{x_0}P^{-1}(x_0).$$

Proof. We prove that $\mathcal{M}$ is a smooth submanifold of $\mathcal{N}$ by applying the rank theorem (Theorem 4.12

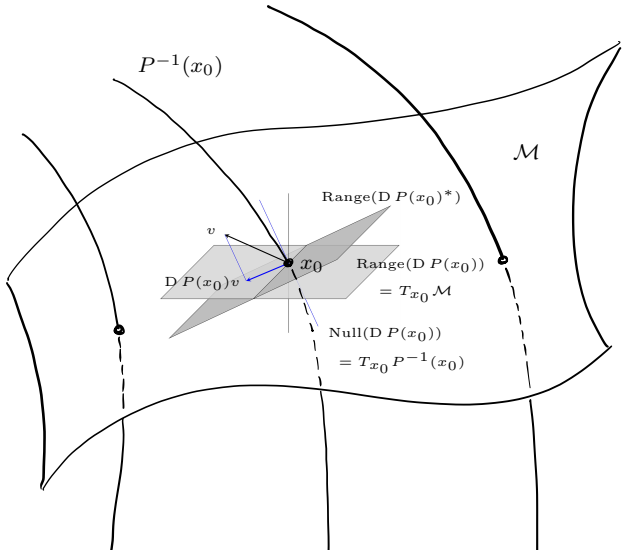


Figure 3.7: We show the anatomy of a constant rank nonlinear projection. The image set of such a projection is a smooth manifold $\mathcal{M}$. The pre-image set of any $x_0 \in \mathcal{M}$ is a manifold of complementary dimension transversal to $\mathcal{M}$. At any point point $x_0 \in \mathcal{M}$ the derivative of the nonlinear projection is an oblique linear projection onto the tangent space of $\mathcal{M}$ with null space tangent to $P^{-1}(x_0)$.

on p.81 in J. M. Lee [149]) to obtain a local parametrization of $\mathcal{M}$ in a neighborhood of any $x_0 \in \mathcal{M}$. We use the preimage theorem (see Section 1.4 of V. Guillemin and A. Pollack [106]) and the fact that $\mathrm{D}\,P(x)$ is a surjective map onto $T_{x_0}\mathcal{M}$ for every $x \in P^{-1}(x_0)$ to prove that $P^{-1}(x_0)$ is a codimension-r submanifold of $\mathcal{N}$. The derivative $\mathrm{D}\,P(x_0)$ is a projection for every $x_0 \in \mathcal{M}$ by the chain rule. Transversality follows from the fact that $T_{x_0}\mathcal{M} = \mathrm{Range}\big(\mathrm{D}\,P(x_0)\big)$ $T_{x_0}P^{-1}(x_0) = \mathrm{Null}\big(\mathrm{D}\,P(x_0)\big)$. □

The rank of any smooth idempotent map on a connected manifold is automatically constant in a sufficiently small neighborhood of the image set, as shown by Theorem 1.15 in P. W. Michor [179]. Consequently, the image set of such a nonlinear projection is still a smooth manifold even if the rank of $\mathrm{D}\,P(x)$ is not constant away from $\mathcal{M}$ (in fact, it can only decrease or remain constant). In such cases, Theorem 3.4.1 describes the behavior of P in a small neighborhood $\mathcal{N}$ of the image manifold.

The converse of Theorem 3.4.1 is also partially true in that any submanifold admits a nonlinear projection of constant rank, as long as we are allowed to restrict our attention to a possibly small neighborhood of the submanifold. Fortunately, this is precisely the case of interest for model reduction since we assume that the states lie near a submanifold that we hope to find. A result known as the tubular neighborhood theorem [106] states that the normal bundle to any smooth submanifold $\mathcal{M}$ in $\mathcal{N}$ can be identified diffeomorphically with a neighborhood of $\mathcal{M}$ in $\mathcal{N}$. It follows immediately that there always exists a constant rank nonlinear projection defined in a neighborhood of $\mathcal{M}$ in $\mathcal{N}$

with image set $\mathcal{M}$. Such projections can be oblique when different Riemannian metrics on $\mathcal{N}$ are used to define the normal bundle to $\mathcal{M}$.

3.4.1 Autoencoder architecture defining constant rank projections

It is possible to constrain the weights in an autoencoder in such a way that it always defines a constant rank projection onto a smooth submanifold of the state space. An autoencoder consisting of an encoder function $\psi_e : \mathcal{X} \to \mathbb{R}^r$ and a decoder function $\psi_d : \mathbb{R}^r \to \mathcal{X}$ defines a projection $P = \psi_d \circ \psi_e$ when the encoder is a left inverse of the decoder, that is when encoding after decoding yields the identity $\psi_e \circ \psi_d = I_r$. We may construct the encoder and decoder by enforcing this property layer-wise. When the activation functions used in the encoder and decoder are inverses of each other then this leads to a biorthogonality constraint for the weights defining corresponding layers of the encoder and decoder. The dynamics of the resulting nonlinear projection-based reduced-order model can be described in the latent space of the autoencoder according to

$$\boxed{\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, z = \mathrm{D}\,\psi_e(\psi_d(z)) f(\psi_d(z), u) \qquad z(0) = \psi_e(x_0),} \tag{3.69}$$

where $\hat{x} = \psi_d(z)$ evolves on the manifold $\mathcal{M} = \mathrm{Image}(P)$ according to Eq. 3.68.

The autoencoder's weights defining such a model may be optimized in a similar way as a the projection subspaces defining the model in Section 3.3.2 and in [192]**. That is, we can simulate Eq. 3.69, generate predicted output time histories $\hat{y}(t) = g(\psi_d(z(t)))$, and minimize the error between these output signals and corresponding output signals of the full-order model. A variety of other cost functions may also be used, one of which is successfully employed in Section 3.4.3, below, to find a nonlinear projection onto the slow manifold in a simple three-dimensional system. As we discussed in Section 3.2.2, an alternative approach is to learn a parametrized approximation of the latent space dynamics (in place of Eq. 3.69) simultaneously with the weights defining the autoencoder using a similar optimization objective. However, as we pointed out in Section 3.3.4, this approach may fail to capture sensitivity mechanisms due to the lack of adjoint sensitivity information coming from the full-order model. To avoid this problem, we suggest pre-projecting the dynamics into a fixed lower-dimensional coordinate system obtained by truncating balanced empirical forward and adjoint covariance matrices as described at the end of Section 3.3.4. As long as amount of available training data exceeds the dimension of this pre-projected system, it should be possible reduce the dimension further without neglecting import sensitivity mechanisms by training an autoencoder and parametrized latent space dynamics.

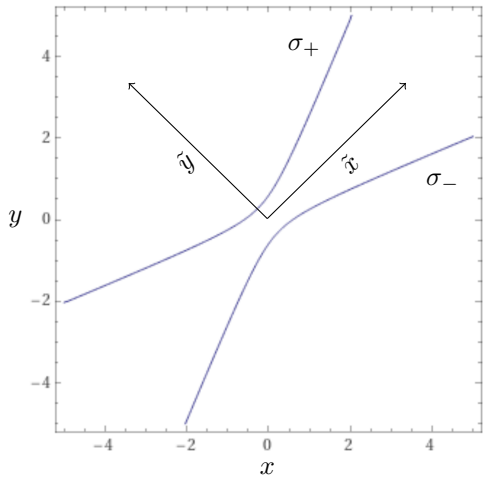


Figure 3.8: We show the smooth, invertible activation functions described by Eq. 3.71 with asymptotes at an angle $\alpha = \pi/8$ from the diagonal. The rotated coordinate system used to define the graphs of the activation functions as the hyperbola given by Eq. 3.70 is also shown.

Let us begin by defining activation functions $\sigma_+, \sigma_- : \mathbb{R} \to \mathbb{R}$ for the encoder and decoder that are smooth and inverses of each other. Geometrically, this means that the graphs of σ_+ and σ_- are related by reflection symmetry about the diagonal line $y = x$ in $\mathbb{R}^2$. Rotating the diagonal to horizontal by working in coordinates $(\tilde{x}, \tilde{y}) = \frac{\sqrt{2}}{2}(x+y, x-y)$, one good choice is to take the graphs of σ_+ and σ_- to be the upper and lower parts of a hyperbola defined by

$$\frac{\tilde{y}^2}{\sin^2\alpha} - \frac{\tilde{x}^2}{\cos^2\alpha} = 1, \tag{3.70}$$

where $0 < \alpha < \pi/4$ gives the angle of the asymptotes with respect to horizontal in the $(\tilde{x}, \tilde{y})$ plane. A plot of this hyperbola with asymptotes at angle $\alpha = \pi/8$ is show in Figure 3.8. Rotating back to (x, y) coordinates, the resulting functions σ_+ and σ_- automatically have the desired reflection symmetry. The condition $0 < \alpha < \pi/4$ ensures that σ_+ and σ_- are well-defined and monotone increasing. These activation functions are given explicitly by

$$\boxed{\sigma_\pm(x) = \frac{bx \pm \sqrt{(b^2-a^2)x^2 + 2a}}{a}, \quad \text{where} \quad \begin{cases} a = \csc^2\alpha - \sec^2\alpha \\ b = \csc^2\alpha + \sec^2\alpha \end{cases}} \tag{3.71}$$

and are defined for all real numbers x since $0 < a < b$. Examining the graphs of these activation functions in Figure 3.8 shows that they resemble smoothed out versions of the commonly used "leaky" rectified linear unit (ReLU).

The encoder and decoder shall have the same number of layers L. We find that the desired

condition $\psi_e \circ \psi_d = I$ can be met when the weight matrix in each layer of the encoder is a left inverse for the weight matrix in the corresponding layer of the decoder. With the convention that the activation functions act element-wise on vectors, suppose that the decoder is defined by

$$\boxed{\psi_d = \psi_d^{(L)} \circ \cdots \circ \psi_d^{(1)}, \qquad \text{where} \qquad \psi_d^{(l)}(z^{(l-1)}) = \Phi_l \sigma_+(z^{(l-1)}) + b_l,} \tag{3.72}$$

and the weight matrices Φ_l are injective. If $\{\Psi_1, \ldots, \Psi_L\}$ are another collection of matrices such that $\Psi_l^* \Phi_l = I$ for each $l = 1, \ldots, L$, then the encoder defined by

$$\boxed{\psi_e = \psi_e^{(1)} \circ \cdots \circ \psi_e^{(L)}, \qquad \text{where} \qquad \psi_e^{(l)}(x^{(l+1)}) = \sigma_- \left(\Psi_l^*(x^{(l+1)} - b_l) \right),} \tag{3.73}$$

is a left inverse of the decoder. In other words, since $\psi_e^{(l)} \circ \psi_d^{(l)} = I$ by construction, it follows that

$$\psi_e \circ \psi_d = \psi_e^{(1)} \circ \cdots \circ \psi_e^{(L)} \circ \psi_d^{(L)} \circ \cdots \circ \psi_d^{(1)} = I, \tag{3.74}$$

and so the autoencoder $P = \psi_d \circ \psi_e$ satisfies $P \circ P = P$ and is indeed a projection.

Autoencoders defined according to Eq. 3.72 and Eq. 3.73 with the biorthogonality constraint $\Psi_l^* \Phi_l = I$ provide a rich parametric class of nonlinear projections onto smooth submanifolds of the state space. The encoder and decoder may be applied to map between points on this manifold and their coordinate representation in the latent space without the possibility of drifting after repeated encodings and decodings. This projection $P = \psi_d \circ \psi_e$ is smooth because the activation functions are infinitely continuously differentiable. Moreover, the projection has constant rank, and so is characterized by Theorem 3.4.1. To see this, we observe that the derivatives of the activation functions in each layer

$$D_d^{(l)} = \mathrm{D}\, \sigma_+(z^{(l-1)}) \qquad D_e^{(l)} = \mathrm{D}\, \sigma_- \left(\Psi_l^*(x^{(l+1)} - b_l) \right) \tag{3.75}$$

are diagonal matrices with strictly positive entries on the diagonal because the activation functions $\sigma_\pm$ always have positive slope. Consequently the derivative of the encoder

$$\mathrm{D}\, \psi_e(x) = D_e^{(1)} \Psi_1^* \cdots D_e^{(L)} \Psi_L^* \tag{3.76}$$

is always surjective. This makes the rank of the projection $\mathrm{D}\, P(x) = \mathrm{D}\, \psi_d(\psi_e(x))\, \mathrm{D}\, \psi_e(x)$ equal to the rank of the decoder $\mathrm{D}\, \psi_d(\psi_e(x))$. The rank of the decoder is always equal to the latent state

dimension since the derivative of the decoder

$$\mathrm{D}\,\psi_d(z) = \Phi_L D_d^{(L)} \cdots \Phi_1 D_d^{(1)} \tag{3.77}$$

is always injective. Finally, we observe that the encoder is a diffeomorphism when it is restricted to the manifold $\mathcal{M}$, and the decoder provides its inverse. When the manifold we wish to learn $\mathcal{M}'$ is not diffeomorphic to $\mathbb{R}^r$, then the approach described above can be used to find a manifold $\mathcal{M}$ that contains $\mathcal{M}'$ and is diffeomorphic to a real space. In this situation, the encoder provides an embedding of the manifold $\mathcal{M}'$ into $\mathbb{R}^r$. This is not so bad because every smooth k-dimensional manifold can be embedded in a real space with dimension $r = 2k$ thanks to Whitney's embedding theorem [278].

3.4.2 Optimization on the manifold of biorthogonal matrices

The question arises as to how the nonlinear, non-convex constraint $\Psi_l^* \Phi_l = I$ should be imposed during the training process, which usually entails gradient descent. Fortunately, the set of matrices having the desired biorthogonality property,

$$\mathcal{B}_{n,r} = \left\{ (\Phi, \Psi) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r} \;:\; \Psi^*\Phi = I_r \right\}, \tag{3.78}$$

is a smooth $2nr - r^2$ dimensional submanifold of the Euclidean space $\mathcal{E} = \mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r}$ thanks to the preimage theorem (see Section 1.4 of V. Guillemin and A. Pollack [106]). Optimization on this biorthogonal manifold in the case when $r = n$ is considered in [96]; however, we are much more interested in the case of dimension reduction when $r < n$. In addition to being a smooth manifold, the inner product

$$\big\langle (X_1, Y_1),\ (X_2, Y_2) \big\rangle_{\mathcal{E}} = \mathrm{Tr}(X_1^* X_2) + \mathrm{Tr}(Y_1^* Y_2) \tag{3.79}$$

on $\mathcal{E}$ induces a Riemannian metric on $\mathcal{B}_{n,r}$. The metric allows us to define the gradients of functions on $\mathcal{B}_{n,r}$ such as the cost function used to optimize the autoencoder.

If we know how to compute the gradient of a function $J : \mathcal{E} \to \mathbb{R}$ defined for all matrices, then it is easy to compute the gradient of its restriction $J|_{\mathcal{B}_{n,r}}$ to biorthogonal matrices. If $p = (\Phi, \Psi) \in \mathcal{B}_{n,r}$ then $\nabla\, J|_{\mathcal{B}_{n,r}}(p)$ is equal to the orthogonal projection of $\nabla\, J(p)$ onto the tangent space $T_p\mathcal{B}_{n,r}$. This key fact means that we can compute the gradient on the biorthogonal manifold by first computing the gradient with respect to the matrices $(\Phi, \Psi) \in \mathcal{E}$ as we would in the case without any constraints. Then we orthogonally project the unconstrained gradient onto the tangent space of the biorthogonal

manifold. To see why this key property is true, we recall that the gradient is defined as the Riesz representative of the derivative. In particular, $\nabla J|_{\mathcal{B}_{n,r}}(p) \in T_p\mathcal{B}_{n,r}$ satisfies

$$\mathrm{D}\, J(p)\xi = \left\langle \nabla J(p),\ \xi \right\rangle = \left\langle \nabla J|_{\mathcal{B}_{n,r}}(p),\ \xi \right\rangle \qquad \forall \xi \in T_p\mathcal{B}_{n,r}, \tag{3.80}$$

and so $\nabla J(p) - \nabla J|_{\mathcal{B}_{n,r}}(p) \perp T_p\mathcal{B}_{n,r}$. There is only one such element — the orthogonal projection of $\nabla J(p)$ onto $T_p\mathcal{B}_{n,r}$. The following Theorem 3.4.2 provides a descrition of the tangent space to the biorthogonal manifold as well as an explicit expression for the orthogonal projection onto it.

Theorem 3.4.2 (The biorthogonal manifold)**.** *The set of matrices $\mathcal{B}_{n,r}$ defined by Eq. 3.78 is a smooth $2nr - r^2$ dimensional submanifold of $\mathcal{E}$, with tangent and normal spaces at a point $(\Phi, \Psi) \in \mathcal{B}_{n,r}$ given by*

$$\begin{aligned} T_{(\Phi,\Psi)}\mathcal{B}_{n,r} &= \{(X, Y) \in \mathcal{E} \ :\ Y^*\Phi + \Psi^* X = 0\} \\ \left(T_{(\Phi,\Psi)}\mathcal{B}_{n,r}\right)^{\perp} &= \left\{(\Psi A,\ \Phi A^T) \in \mathcal{E} \ :\ A \in \mathbb{R}^{r\times r}\right\}. \end{aligned} \tag{3.81}$$

The orthogonal projection of any $(X, Y) \in \mathcal{E}$ onto $T_{(\Phi,\Psi)}\mathcal{B}_{n,r}$ is given by

$$\boxed{P_{(\Phi,\Psi)}(X, Y) = \left(X - \Psi A,\ Y - \Phi A^T\right),} \tag{3.82}$$

where $A \in \mathbb{R}^{r\times r}$ is the unique solution of the Sylvester equation

$$A(\Phi^*\Phi) + (\Psi^*\Psi)A = Y^*\Phi + \Psi^* X. \tag{3.83}$$

The Sylvester equation is equivalent to the symmetric, positive-definite linear system

$$\left[(\Phi^*\Phi) \otimes I_r + I_r \otimes (\Psi^*\Psi)\right] \mathrm{vec}(A) = \mathrm{vec}\left(Y^*\Phi + \Psi^* X\right), \quad \mathrm{vec}(A) = \begin{bmatrix} \mathrm{col}_1(A) \\ \vdots \\ \mathrm{col}_r(A) \end{bmatrix}. \tag{3.84}$$

Proof. We construct $\mathcal{B}_{n,r}$ as the preimage of the regular value 0 under the map $F : X, Y \mapsto Y^*X - I_r$ using the preimage theorem [106]. The tangent space to $\mathcal{B}_{n,r}$ at (Φ, Ψ) is given by the null-space of $\mathrm{D}\, F(\Phi, \Psi)$ according to the local submersion theorem [106]. We give the details of the proof in Appendix 3.A. □

Due to the curvature of the manifold $\mathcal{B}_{n,r}$, taking finite-sized steps along tangent directions com-

puted based on the gradient may produce new points that do not lie in $\mathcal{B}_{n,r}$. To perform optimization we must introduce a correction to bring the new point back into $\mathcal{B}_{n,r}$ without undoing any progress we make by moving in the search direction. A correction mapping that remains asymptotically small in comparison with the displacement in the tangent space is called a "retraction." Retractions allow us to parametrize a neighborhood of our current iterate p in the curved manifold $\mathcal{B}_{n,r}$ using tangent vectors in the Euclidean space $T_p\mathcal{B}_{n,r}$. Below, we give the formal definition of a retraction presented by P. A. Absil in [1].

Definition 3.4.3 (Retraction [1])**.** *A retraction on a manifold $\mathcal{M}$ is a smooth mapping R from the tangent bundle $T\mathcal{M}$ onto $\mathcal{M}$ with the following properties. Let R_p denote the restriction of R to $T_p\mathcal{M}$.*

1. *(Base point preservation) $R_p(0) = p$.*
2. *(Local rigidity) R_p satisfies*

$$\mathrm{D}\, R_p(0)\xi = \xi \qquad \forall \xi \in T_p\mathcal{M}. \tag{3.85}$$

The retraction converts small vectors in the tangent space into small displacements on the underlying manifold while preserving the first derivative so that

$$\mathrm{D}\, J(p)\xi = \mathrm{D}\left(J \circ R_p\right)(0)\xi, \qquad \forall \xi \in T_p\mathcal{M}. \tag{3.86}$$

A simple retraction on $\mathcal{B}_{n,r}$ is given below by Theorem 3.4.4.

Theorem 3.4.4 (a retraction on the biorthogonal manifold)**.** *Let $(\Phi, \Psi) \in \mathcal{B}_{n,r}$ and $(X, Y) \in T_{(\Phi,\Psi)}\mathcal{B}_{n,r}$. Then the map defined by*

$$\boxed{R_{(\Phi,\Psi)}(X,Y) = \left((\Phi + X)\left[(\Psi + Y)^*(\Phi + X)\right]^{-1},\ \Psi + Y\right)} \tag{3.87}$$

is a retraction.

Proof. It is clear from the definition that our map preserves base points, that is,

$$R_{(\Phi,\Psi)}(0,0) = (\Phi, \Psi), \qquad \forall (\Phi, \Psi) \in \mathcal{B}_{n,r}. \tag{3.88}$$

Differentiating the map at $(\Phi, \Psi) \in \mathcal{B}_{n,r}$ yields

$$\mathrm{D}\, R_{(\Phi,\Psi)}(0,0)(V,W) = \left(V\left[\Psi^*\Phi\right]^{-1} - \Phi\left[\Psi^*\Phi\right]^{-1}\left[W^*\Phi + \Psi^*V\right]\left[\Psi^*\Phi\right]^{-1},\ W\right), \tag{3.89}$$

for every tangent vector $(V, W) \in T_{(\Phi,\Psi)}\mathcal{B}_{n,r}$. Using the fact that $\Psi^*\Phi = I_r$ and that $W^*\Phi + \Psi^*V = 0$ (see Theorem 3.4.2), we conclude that our map R satisfies local rigidity, i.e.,

$$\mathrm{D}\, R_{(\Phi,\Psi)}(0,0)(V,W) = (V,W)\,, \qquad \forall((\Phi,\Psi),(V,W)) \in T\mathcal{B}_{n,r}. \tag{3.90}$$

□

Using this retraction, we could implement the Riemannian stochastic gradient descent method of S. Bonnabel [28] to optimize the weights of our autoencoder constructed in Section 3.4.1 over a product of L biorthogonal manifolds corresponding to matching layers of the encoder and decoder.

In general, a retraction agrees with the exponential map on a Riemannian manifold up to first order. A retraction that agrees with the exponential map to second order is called a second-order retraction [2]. When a second-order retraction is used, the second derivative information about a smooth cost function J on the manifold is preserved when J is pulled back to the tangent space by the retraction according to $J \circ R_p$. The following definition given in [2] is a necessary and sufficient condition for a retraction to agree with the exponential map to second order.

Definition 3.4.5 (Second-Order Retraction [2])**.** *A second-order retraction is a retraction $R : T\mathcal{M} \to \mathcal{M}$ that satisfies the additional condition*

$$\frac{d^2}{dt^2} R_p(t\xi)\bigg|_{t=0} \in (T_p\mathcal{M})^{\perp}\,, \qquad \forall \xi \in T_p\mathcal{M}. \tag{3.91}$$

This condition says that the curve $t \mapsto R_p(t\xi)$ cannot experience any acceleration tangent to $\mathcal{M}$ at the base point p when $t = 0$. In general, the curves generated by the exponential map (called geodesics) satisfy such a condition everywhere, and not just at a given point.

A second-order retraction on $\mathcal{B}_{n,r}$ is given below in Theorem 3.4.6 by introducing correction terms to the retraction described earlier in Theorem 3.4.4.

Theorem 3.4.6 (A Second-Order Retraction)**.** *The following map defines a second-order retraction on the biorthogonal manifold:*

$$\boxed{R_{(\Phi,\Psi)}(X,Y) = \left((\Phi + X + \Psi A)H,\ \Psi + Y + \Phi A^T\right), \qquad ((\Phi,\Psi),(X,Y)) \in T\mathcal{B}_{n,r},} \tag{3.92}$$

where $H = \left[(\Psi + Y + \Phi A^T)^*(\Phi + X + \Psi A)\right]^{-1}$ and $A \in \mathbb{R}^{r\times r}$ is the unique solution of the Sylvester equation

$$A\,(\Phi^*\Phi) + (\Psi^*\Psi)\,A + Y^*X = 0. \tag{3.93}$$

The push-forward map of this retraction is given by

$$\boxed{\mathrm{D}\, R_{(\Phi,\Psi)}(X,Y)(V,W) = \left((V+\Psi A')H + (\Phi + X + \Psi A)H',\ W + \Phi(A')^T\right),} \tag{3.94}$$

where $H' = -H\left[(W+\Phi(A')^T)^*(\Phi+X+\Psi A) + (\Psi + Y + \Phi A^T)^*(V+\Psi A')\right] H$ *and* $A' \in \mathbb{R}^{r\times r}$ *solves the Sylvester equation*

$$A'\left(\Phi^*\Phi\right) + \left(\Psi^*\Psi\right)A' + W^*X + Y^*V = 0. \tag{3.95}$$

Proof. The proof is tedious, so we provide it in Appendix 3.A. □

Many optimization techniques such as quasi-Newton methods [117, 222], conjugate gradient algorithms [222, 235], and variance-reduced stochastic gradient descent [238], rely on linear combinations of vectors from different tangent spaces to compute the next search direction. Consequently, a way of carrying a tangent vector at one point to a tangent vector at another point is needed. The most natural way to do this on a Riemannian manifold is parallel translation of vectors along geodesics. Such calculations can be computationally expensive to carry out, so [1] provides a more general notion of "vector transport", stated below in Definition 3.4.7, which retains only the properties that are needed in the context of optimization.

Definition 3.4.7 (Vector Transport [1])**.** *Let the "Whitney sum"*

$$T\mathcal{M} \oplus T\mathcal{M} = \{(\eta_p, \xi_p)\ :\ \eta_p, \xi_p \in T_p\mathcal{M},\ p \in \mathcal{M}\} \tag{3.96}$$

denote pairs of tangent vectors sharing the same root points. A vector transport on the manifold $\mathcal{M}$ *is a smooth mapping*

$$T\mathcal{M} \oplus T\mathcal{M} \to T\mathcal{M}\ :\ (\eta_p, \xi_p) \mapsto \mathcal{T}_{\eta_p}(\xi_p) \tag{3.97}$$

satisfying the following properties:

1. *(Associated retraction) There exists a retraction* R*, called the retraction associated with* $\mathcal{T}$*, such that* $\mathcal{T}_{\eta_p}(\xi_p) \in T_{R_p(\eta_p)}\mathcal{M}$ *for every* $(\eta_p, \xi_p) \in T\mathcal{M} \oplus T\mathcal{M}$*.*
2. *(Consistency)* $\mathcal{T}_{0_p}(\xi_p) = \xi_p$ *for all* $\xi_p \in T\mathcal{M}$
3. *(Linearity)* $\mathcal{T}_{\eta_p}(a\xi_p + b\zeta_p) = a\mathcal{T}_{\eta_p}(\xi_p) + b\mathcal{T}_{\eta_p}(\zeta_p)$ *for all* $a, b \in \mathbb{R}$*,* $\eta_p, \xi_p, \zeta_p \in T_p\mathcal{M}$*, and every* $p \in \mathcal{M}$*.*

According to [1], vector transports can be obtained by differentiating retractions, that is,

$$\mathcal{T}_{\eta_p}(\xi_p) := \mathrm{D}\, R_p(\eta_p)\xi_p = \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, R_p(\eta_p + t\xi_p)\bigg|_{t=0}. \tag{3.98}$$

Consequently, the derivative map given by Eq. 3.94 for the second-order retraction defined in Theorem 3.4.6 provides us with a vector transport on the biorthogonal manifold. Another vector transport is provided by orthogonal projection onto the tangent space using Eq. 3.82 at the point provided by a given retraction.

Given the retraction and transport defined in Theorem 3.4.6, or by orthogonal projection, we now have all of the ingredients we need to implement a variety of optimization algorithms [1] on the biorthogonal manifold. For an example of how retractions and vector transports can be used for conjugate gradient-based optimization, see [192]**. Such algorithms can be applied to train the weights in our autoencoder constructed in Section 3.4.1 by optimizing the weights of the encoder and decoder on a product of L biorthogonal manifolds.

3.4.3 Results for a simple system with a slow manifold

In this section we apply the autoencoder developed above to construct a reduced-order model of a system with a known slow manifold. We consider the system

$$\begin{aligned} \dot{x}_1 &= \mu x_1 - \omega x_2 - x_1 x_3 \\ \dot{x}_2 &= \omega x_1 + \mu x_2 - x_2 x_3 \\ \varepsilon \dot{x}_3 &= x_1^2 + x_2^2 - x_3, \end{aligned} \tag{3.99}$$

originally proposed by B. R. Noack et al. [188] as a mean-field model for the formation of vortices in the wake of a cylinder through a supercritical Hopf bifurcation arising from the quadratic nonlinearities of the incompressible Navier-Stokes equations. For small $\varepsilon > 0$, Eq. 3.99 is a slow-fast system and standard Fenichel theory (see C. Kuehn [143]) can be used to show that the state is attracted to a slow invariant manifold lying near the critical manifold $x_3 = x_1^2 + x_2^2$. By Theorem 11.1.1 in [143], we may express the slow manifold as an asymptotic series $x_3 = h(x_1, x_2) = \sum_{k=0}^{\infty} h_k(x_1, x_2)\varepsilon^k$ with the invariance condition in polar coordinates $x_1 = r\cos\theta$, $x_2 = r\sin\theta$ yielding

$$h_{k+1} = r\sum_{l=0}^{k} h_l \frac{\partial h_{k-l}}{\partial r} - \mu r \frac{\partial h_k}{\partial r}, \qquad h_0 = r^2. \tag{3.100}$$

The first three terms of the series provide an approximation,

$$x_3 = r^2 + 2r^2(r^2 - \mu)\varepsilon + 4r^2(r^2 - \mu)(3r^2 - \mu)\varepsilon^2 + \mathcal{O}(\varepsilon^3). \tag{3.101}$$

We note that only even powers of r appear in each h_k, reflecting the axisymmetry of the slow manifold. When $\omega \neq 0$, the system undergoes a supercritical Hopf bifurcation with a stable limit cycle $x_3 = r^2 = \mu$ at frequency ω appearing as μ passes through 0 [104]. The center manifold for this bifurcation (with $\dot{\mu} = 0$ treated as an extra state equation) coincides with the slow manifold computed above. We choose $\varepsilon = 0.05$, $\mu = 1$, $\omega = 10$, and we draw the initial conditions at random from a Guassian distribution centered about the origin with covariance matrix $0.01 I_3$.

We consider $m = 10$ such trajectories of Eq. 3.99 over the time interval $t \in [0, 2\pi]$ and we optimize a nonlinear projection operator using an autoencoder described in Section 3.4.1. Letting $x^{(i)}(t)$ denote the ith trajectory, $P = \psi_d \circ \psi_e$, and $\tilde{x}^{(i)}(t) = P\big(x^{(i)}(t)\big)$, we minimize the cost function

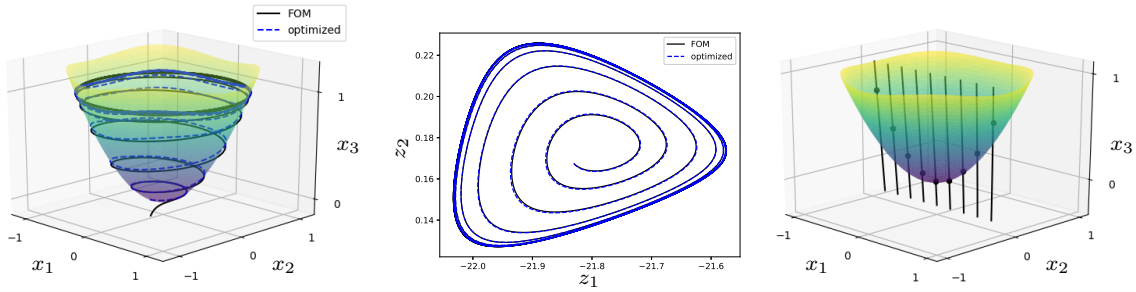
$$J = \frac{1}{m}\sum_{i=1}^{m}\left\{\frac{1}{E_i}\int_0^{2\pi}\left\|x^{(i)}(t) - P\big(x^{(i)}(t)\big)\right\|_2^2 \mathrm{d}\,t \right. \\ \left. + \frac{1}{F_i}\int_0^{2\pi}\left\|\mathrm{D}\,P\big(x^{(i)}(t)\big) f\big(x^{(i)}(t)\big) - \mathrm{D}\,P\big(\tilde{x}^{(i)}(t)\big) f\big(\tilde{x}^{(i)}(t)\big)\right\|_2^2 \mathrm{d}\,t\right\}, \tag{3.102}$$

where each term is normalized by an appropriate total energy

$$E_i = \int_0^{2\pi}\left\|x^{(i)}(t)\right\|_2^2 \mathrm{d}\,t, \qquad F_i = \int_0^{2\pi}\left\|\mathrm{D}\,P\big(x^{(i)}(t)\big) f\big(x^{(i)}(t)\big)\right\|_2^2 \mathrm{d}\,t. \tag{3.103}$$

The first term of Eq. 3.102 penalizes the error between the points along the trajectories and their projections onto the learned manifold. The second term of Eq. 3.102 penalizes the difference between the true time derivative $\dot{x}^{(i)}(t) = f\big(x^{(i)}(t)\big)$ projected onto the manifold and the time derivative of the reduced-order model Eq. 3.68 evaluated at the projected point $P\big(x^{(i)}(t)\big)$. Optimization was carried out using the Riemannian Dai-Yuan conjugate gradient algorithm of H. Sato [235] using the simple first-order retraction and vector transport on the biorthogonal manifold provided by Theorem 3.4.4. Line search was performed using the bisection method described in [39] to satisfy the weak Wolfe conditions.

To accurately approximate the underlying manifold in such a low-dimensional space, we needed to use a deep neural network with $L = 50$ layers in the encoder and the same number of layers in the decoder. The dimension of the latent space was $r = 2$ and the last 49 layers of the decoder (and the first 49 layers of the encoder) had dimension 3, so the corresponding weight matrices Φ_l and Ψ_l

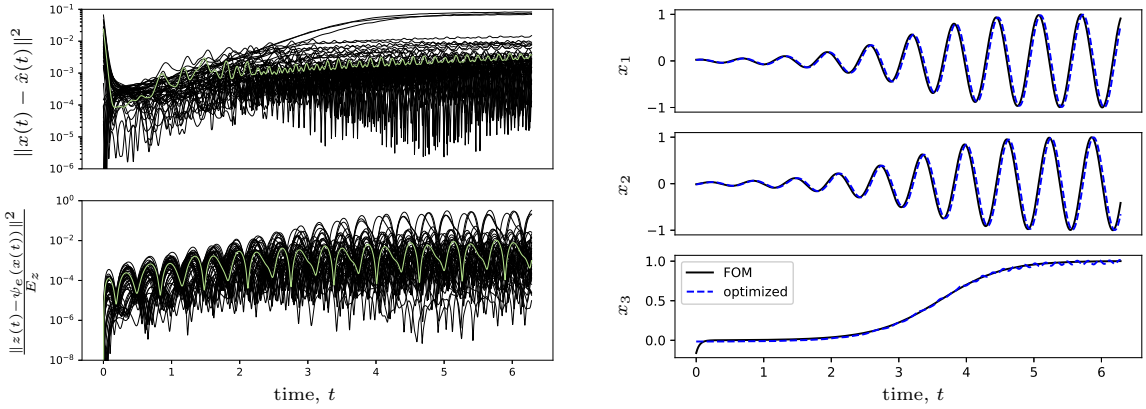


(a) example trajectory (b) latent space trajectory (c) projection fibers

Figure 3.9: In panel (a), we show a trajectory of the full-order model Eq. 3.99 together with the predicted trajectory of the reduced-order model Eq. 3.68. The corresponding trajectories in the latent space are shown in panel (b). The learned manifold together with several fibers of the optimized projection are shown in panel (c).

were constrained to be inverses of one another for $l = 2, 3, ..., 50$. We used the hyperbolic activation functions described by Eq. 3.71 with an asymptote angle $\alpha = \pi/8$. We initialized $\Phi_2 = \cdots = \Phi_{50} = \tan(\pi/4 + \alpha)I_3$, $\Psi_2 = \cdots = \Psi_{50} = \cot(\pi/4 + \alpha)I_3$, with zero biases $b_2 = \cdots = b_{50} = 0$. We used POD on the data after being fed through $\psi_e^{(2)} \circ \cdots \psi_e^{(50)}$ to initialize the 3×2 weight matrices Φ_1 and Ψ_1 with $b_1 = 0$. The factor $\cot(\pi/4 + \alpha)$ was used to prevent the scale of the latent space variables from growing enormous as the layers were compounded.

The manifold associated with the learned nonlinear projection operator is shown in Figure 3.9a along with a trajectory of the full-order model Eq. 3.99 and a predicted trajectory. The prediction was obtained by projecting the initial condition onto the learned manifold and evolving the reduced-order model Eq. 3.68. The trajectories in the latent space of the autoencoder are shown in Figure 3.9b. The error between the trajectories of the two models on 50 such unseen initial conditions drawn at random is plotted in Figure 3.10a with the specific trajectory shown in Figures 3.9a and 3.9b highlighted in green. We observe that the learned manifold is extremely close to the true parabolic slow manifold $x_3 = x_1^2 + x_2^2$. The predictions made by the reduced-order model evolving on the learned manifold also agree closely with the trajectories of the original system. In Figure 3.9c we show the fibers associated with the learned projection (see Theorem 3.4.1 and Figure 3.7) and we observe that the projection is very nearly vertical. This means that the optimized nonlinear projection has learned to project points vertically onto the parabolic slow manifold, which is essentially what happens to the dynamics of the original system when $\varepsilon \to 0$ in Eq. 3.99. The largest errors along predicted trajectories occurred when the initial conditions were very close to the fiber passing through the origin, resulting in phase errors as shown in Figure 3.10b. However, even the worst-case phase error that we observed among the testing trajectories was very small.



(a) error on testing trajectories

(b) worst predicted trajectory

Figure 3.10: In panel (a) we show the error of the nonlinear-projection-based reduced-order model Eq. 3.68 for the system Eq. 3.99 on 50 unseen testing trajectories selected at random. Here, the normalization factor E_z is the mean square fluctuation of the latent space trajectories $\psi_e(x(t))$ about their average. The error along the example trajectory shown in Figure 3.9a is highlighted in green. In panel (b) we plot the trajectory corresponding to the highest error at the final time.

Appendix

3.A Chapter 3 Proofs

Proof of Lemma 3.1.1 (inhomogeneous Grönwall inequality). Let us define the function

$$v(t) = e^{-Lt} \int_0^t [Lw(\tau) + b(\tau)] \, \mathrm{d}\, \tau \tag{3.104}$$

and observe that

$$v'(t) = e^{-Lt} \left\{ -L \int_0^t [Lw(\tau) + b(\tau)] \, \mathrm{d}\, \tau + Lw(t) + b(t) \right\} \leq e^{-Lt} \{La + b(t)\} . \tag{3.105}$$

Integrating, and noting that $v(0) = 0$ we find

$$v(t) \leq a - ae^{-Lt} + \int_0^t e^{-L\tau} b(\tau) \, \mathrm{d}\, \tau \tag{3.106}$$

and so we obtain

$$w(t) \leq a + e^{Lt} v(t) \leq ae^{Lt} + \int_0^t e^{L(t-\tau)} b(\tau) \, \mathrm{d}\, \tau. \tag{3.107}$$

□

Proof of Theorem 3.3.2 (Typical intersections of ranges and null-spaces). The set $\mathcal{S}$ is open because

the function $\phi : \mathbb{R}^{n\times r} \to \mathbb{R}$ defined by $\phi(X) = \det(T^*X)$ is continuous, and $\mathcal{S} = \phi^{-1}(\mathbb{R} \setminus \{0\})$ is the pre-image of an open set. The sets $\mathcal{S}_+ = \phi^{-1}((0,\infty))$ and $\mathcal{S}_- = \phi^{-1}((-\infty,0))$ are open for the same reason.

To show that $\mathcal{S}$ is dense in $\mathbb{R}^{n\times r}$, choose any $X \in \mathbb{R}^{n\times r}$ for which $\det(T^*X) = 0$. Consider a full-sized singular value decomposition $T^*X = U\Sigma V^*$ and let

$$X_t = X + tT(T^*T)^{-1}UV^*. \tag{3.108}$$

Taking the determinant, we find

$$\det(T^*X_t) = \det(U\Sigma V^* + tUV^*) = \det(U)\det(\Sigma + tI)\det(V^*) > 0 \qquad \forall t > 0, \tag{3.109}$$

and so $X_t \in \mathcal{S}$ for every $t > 0$ even though $X_0 = X \notin \mathcal{S}$. Since the map $t \mapsto X_t$ is continuous, it follows that every open neighborhood of X contains an element of $\mathcal{S}$, proving that $\mathcal{S}$ is dense in $\mathbb{R}^{n\times r}$.

To show that $\mathcal{S}^c$, the complement of $\mathcal{S}$ in $\mathbb{R}^{n\times r}$, has Lebesgue measure zero, we observe that $X \mapsto \det(T^*X)$ is a non-constant polynomial function on $\mathbb{R}^{n\times r}$ and $\mathcal{S}^c$ is the zero set of this polynomial. Since the zero set of any non-constant polynomial has Lebesgue measure zero [53], it follows that $\mathcal{S}^c$ has measure zero.

We can also provide an analytic proof that $\mathcal{S}^c$ has measure zero by relying on Lebesgue's density theorem (see Section 7.2 of W. Rudin [233]). If μ denotes the Lebesgue measure, then the metric density of a set E at a point x is defined to be

$$\rho_E(x) = \lim_{\epsilon\to 0^+} \frac{\mu\left(E \cap B(x,\epsilon)\right)}{\mu(B(x,\epsilon))}, \tag{3.110}$$

where $B(x,\epsilon)$ denotes the open ball of radius r centered at x. Lebesgue's density theorem states that the metric density is unity, $\rho_E(x) = 1$, for almost every point x of E. By showing that $\rho_{\mathcal{S}^c}(X) < 1$ for every $X \in \mathcal{S}^c$ where $\rho_{\mathcal{S}^c}(X)$ is defined, it follows from this theorem that $\mathcal{S}^c$ has Lebesgue measure zero. Choosing any $X \in \mathcal{S}^c$ and any $\epsilon > 0$, we use Eq. 3.108 to construct a point

$$X_0 = X + \frac{\epsilon}{2\|T(T^*T)^{-1}UV^*\|} T(T^*T)^{-1}UV^* \in \mathcal{S}, \tag{3.111}$$

which lies in $\mathcal{S}$ at a distance $\|X_0 - X\| = \epsilon/2$ from X. For convenience, we take $\|\cdot\|$ to be the operator norm, but this choice is not important because all norms are equivalent on finite-dimensional vector

spaces. We take $\delta_\epsilon = \alpha\epsilon$ with the constant α given by

$$\alpha = \min\left\{\frac{1}{2},\ \frac{1}{2\|T\|\|T(T^*T)^{-1}UV^*\|}\right\}, \tag{3.112}$$

and observe that $B(X_0, \delta_\epsilon) \subset B(X, \epsilon)$ and for any $Y \in B(X_0, \delta_\epsilon)$, the smallest singular value, σ_r, of T^*Y is bounded below by

$$\begin{aligned}\sigma_r(T^*Y) &= \sigma_r\left[T^*X_0 + T^*(Y - X_0)\right] \\ &\geq \sigma_r(T^*X_0) - \|T\|\|Y - X_0\| \\ &> \frac{\epsilon}{2\|T(T^*T)^{-1}UV^*\|} - \|T\|\delta_\epsilon \geq 0.\end{aligned} \tag{3.113}$$

Consequently, T^*Y is invertible for every $Y \in B(X_0, \delta_\epsilon)$ and so $B(X_0, \delta_\epsilon) \subset \mathcal{S}$. Therefore the metric density of $\mathcal{S}^c$ at $X \in \mathcal{S}^c$ is bounded above by

$$\rho_{\mathcal{S}^c}(X) \leq \lim_{\epsilon \to 0^+} \frac{\mu\left(B(X, \epsilon) \setminus B(X_0, \delta_\epsilon)\right)}{\mu(B(X, \epsilon))} = 1 - \lim_{\epsilon \to 0^+} \frac{\mu(B(X_0, \alpha\epsilon))}{\mu(B(X, \epsilon))} = 1 - \alpha^{nr} < 1, \tag{3.114}$$

when it is defined. Therefore, the Lebesgue measure of $\mathcal{S}^c$ is equal to the measure of points in $\mathcal{S}^c$ with metric density undefined or less than unity. By Lebesgue's density theorem, the Lebesgue measure of $\mathcal{S}^c$ is equal to zero.

To show that $\mathcal{S}_+$ is connected, we choose any $X, Y \in \mathcal{S}_+$. Recall that the general linear group GL_r of invertible $r \times r$ matrices has two connected component corresponding to matrices with positive and negative determinants [109]. Since $\det((T^*X)^{-1}(T^*Y)) > 0$, it follows that there is a continuous path $t \mapsto G_t \in GL_r$ such that $G_0 = I$ and $G_1 = (T^*X)^{-1}(T^*Y)$. Consequently, the path $t \mapsto X_t = XG_t$ has $X_0 = X$, $T^*X_1 = T^*XG_1 = T^*Y$, and X_t remains in $\mathcal{S}_+$ because $\det(T^*X_t) = \det(T^*X)\det(G_t) > 0$ for every $t \in [0, 1]$. Now, we construct a path from X_1 to Y by letting $Z_t = tY + (1 - t)X_1$ and observing that

$$\det(T^*Z_t) = \det(tT^*Y + (1 - t)T^*X_1) = \det(T^*Y) > 0 \qquad \forall t \in [0, 1], \tag{3.115}$$

and so Z_t remains in $\mathcal{S}_+$. Connecting the two paths $X = X_0 \rightsquigarrow X_1 = Z_0 \rightsquigarrow Z_1 = Y$, we have constructed a path from X to Y that remains in $\mathcal{S}_+$. The same proof may be repeated verbatim, modulo sign flips, for $X, Y \in \mathcal{S}_-$. □

Proof of Theorem 3.4.1 (Constant-Rank Projections). We begin by showing that the image set $\mathcal{M} =$

Image(P) is a submanifold of $\mathcal{N}$. To do this, it suffices to show that each $x_0 \in \mathcal{M}$ is contained in a local chart for $\mathcal{M}$. Choose any $x_0 \in \mathcal{M}$ and observe that $P(x_0) = x_0$. By the rank theorem (Theorem 4.12 on p.81 in J. M. Lee [149]), there are two open neighborhoods $\mathcal{U}$ and $\mathcal{V}$ of x_0 in $\mathcal{N}$ and diffeomorphisms $\phi : \mathbb{R}^n \to \mathcal{U}$ and $\psi : \mathbb{R}^n \to \mathcal{V}$ such that $P(\mathcal{U}) \subset \mathcal{V}$ and the coordinate representation $\hat{P} = \psi^{-1} \circ P \circ \phi : \mathbb{R}^n \to \mathbb{R}^n$ is given by

$$\hat{P}(z^1, \ldots, z^n) = (z^1, \ldots, z^r, 0, \ldots, 0). \tag{3.116}$$

If $H = \{(z^1, \ldots, z^n) \in \mathbb{R}^n \ : \ z^{r+1} = \cdots = z^n = 0\}$ then it is clear that $P(\mathcal{U}) = \psi(H)$. Moreover, since $P \circ P = P$, it is clear that $\mathcal{M} \cap \mathcal{U} = P(\mathcal{U}) \cap \mathcal{U} = P(\mathcal{U}) \cap (\mathcal{U} \cap \mathcal{V})$. Letting $W = \psi^{-1}(\mathcal{U} \cap \mathcal{V})$, which is open in $\mathbb{R}^n$, we obtain $\mathcal{M} \cap \mathcal{U} = \psi(H) \cap \psi(W) = \psi(H \cap W)$. Thus, the restriction $\psi|_H : W \cap H \to \mathcal{M} \cap \mathcal{U}$ provides a local parametrization of $\mathcal{M} \cap \mathcal{U}$. Since $x_0 \in \mathcal{M}$ was arbitrary, we conclude that $\mathcal{M}$ is a smooth r-dimensional manifold.

Because the derivative of $P : \mathcal{N} \to \mathcal{M}$ has constant rank equal to the dimension of $\mathcal{M}$, it follows that $\mathrm{D}\, P(x) : T_x\mathcal{N} \to T_{x_0}\mathcal{M}$ is surjective for each $x \in P^{-1}(x_0)$. Consequently each $x_0 \in \mathcal{M}$ is a regular value of P, and so $P^{-1}(x_0)$ is a smooth codimension-r submanifold of $\mathcal{N}$. The tangent space of $P^{-1}(x_0)$ at x is equal to the null space of $\mathrm{D}\, P(x)$ by the preimage theorem (see Section 1.4 of V. Guillemin and A. Pollack [106]).

Finally, for any $x_0 \in \mathcal{M}$ the map $\mathrm{D}\, P(x_0) : T_{x_0}\mathcal{N} \to T_{x_0}\mathcal{N}$ is a projection due to the chain rule

$$\mathrm{D}\, P(x_0) = \mathrm{D}(P \circ P)(x_0) = \mathrm{D}\, P(x_0)\, \mathrm{D}\, P(x_0). \tag{3.117}$$

Choosing any element $v \in (T_{x_0}\mathcal{M}) \cap (T_{x_0}P^{-1}(x_0)) = \mathrm{Range}(\mathrm{D}\, P(x_0)) \cap \mathrm{Null}(\mathrm{D}\, P(x_0))$, it follows from Eq. 3.117 that $v = \mathrm{D}\, P(x_0)v = 0$. Since $T_{x_0}\mathcal{M}$ and $T_{x_0}P^{-1}(x_0)$ have complementary dimensions in $T_{x_0}\mathcal{N}$ and $(T_{x_0}\mathcal{M}) \cap (T_{x_0}P^{-1}(x_0)) = \{0\}$, we have

$$T_{x_0}\mathcal{N} = T_{x_0}\mathcal{M} \oplus T_{x_0}P^{-1}(x_0), \tag{3.118}$$

and so $\mathcal{M}$ and $T^{-1}(x_0)$ intersect transversally at x_0. □

Proof of Theorem 3.4.2 (Biorthogonal Manifold). Consider the smooth map $F : \mathcal{E} \to \mathbb{R}^{r \times r}$ defined by

$$F(X, Y) = Y^*X - I_r. \tag{3.119}$$

We observe that $\mathcal{B}_{n,r} = F^{-1}(0)$ is the preimage of the zero matrix under the map F. If $(\Phi, \Psi) \in \mathcal{B}_{n,r}$,

then the derivative of F at this point is given by the map

$$\mathrm{D}\,F(\Phi,\Psi):(X,Y)\mapsto Y^*\Phi+\Psi^*X,\qquad (X,Y)\in\mathcal{E}. \tag{3.120}$$

It is easy to see that the derivative is surjective at every $(\Phi,\Psi)\in\mathcal{B}_{n,r}$ because

$$\mathrm{D}\,F(\Phi,\Psi)(\Phi A,0)=A,\qquad \forall A\in\mathbb{R}^{r\times r}. \tag{3.121}$$

By the preimage theorem [106], it follows that $\mathcal{B}_{n,r}$ is a smooth sub-manifold of codimension r^2 in $\mathcal{E}$. Since $\mathcal{E}$ is $2nr$-dimensional, the dimension of $\mathcal{B}_{n,r}$ is $2nr-r^2$.

By the local submersion theorem [106], the tangent space at $(\Phi,\Psi)\in\mathcal{B}_{n,r}$ is characterized by the null space of the derivative, that is,

$$T_{(\Phi,\Psi)}\mathcal{B}_{n,r}=\ker\mathrm{D}\,F(\Phi,\Psi)=\{(X,Y)\in\mathcal{E}\ :\ Y^*\Phi+\Psi^*X=0\}. \tag{3.122}$$

Since $\mathrm{D}\,F(\Phi,\Psi)$ is a finite-dimensional linear map between $\mathcal{E}$ and the Euclidean space $\mathbb{R}^{r\times r}$, we know that

$$\left(T_{(\Phi,\Psi)}\mathcal{B}_{n,r}\right)^{\perp}=(\ker\mathrm{D}\,F(\Phi,\Psi))^{\perp}=\mathrm{Range}\left(\mathrm{D}\,F(\Phi,\Psi)^*\right), \tag{3.123}$$

where $\mathrm{D}\,F(\Phi,\Psi)^*:\mathbb{R}^{r\times r}\to\mathcal{E}$ is the adjoint of $\mathrm{D}\,F(\Phi,\Psi)$. We claim that the adjoint operator is given by

$$\mathrm{D}\,F(\Phi,\Psi)^*:A\mapsto\left(\Psi A,\Phi A^T\right). \tag{3.124}$$

To verify this, choose any $A\in\mathbb{R}^{r\times r}$ and $(X,Y)\in\mathcal{E}$. Then

$$\begin{aligned}\left\langle A,\ \mathrm{D}\,F(\Phi,\Psi)(X,Y)\right\rangle_{\mathbb{R}^{r\times r}}&=\mathrm{Tr}\left[A^T\left(Y^*\Phi+\Psi^*X\right)\right] &(3.125)\\ &=\mathrm{Tr}\left(\Phi^*YA\right)+\mathrm{Tr}\left(A^T\Psi^*X\right) &(3.126)\\ &=\mathrm{Tr}\left(A\Phi^*Y\right)+\mathrm{Tr}\left(A^T\Psi^*X\right), &(3.127)\end{aligned}$$

where we have used the invariance of the trace under transposition and cyclic permutation as well as the symmetry of the state space's inner product $\langle\cdot,\ \cdot\rangle$ which implies $(Y^*\Phi)^T=\Phi^*Y$. It remains to verify that $A\Phi^*=(\Phi A^T)^*$ and $A^T\Psi^*=(\Psi A)^*$, for if this is the case then we obtain

$$\left\langle A,\ \mathrm{D}\,F(\Phi,\Psi)(X,Y)\right\rangle_{\mathbb{R}^{r\times r}}=\left\langle\mathrm{D}\,F(\Phi,\Psi)^*A,\ (X,Y)\right\rangle_{\mathcal{E}}=\left\langle\left(\Psi A,\Phi A^T\right),\ (X,Y)\right\rangle_{\mathcal{E}}. \tag{3.128}$$

Choosing any $u \in (\mathbb{R}^n,\ \langle\cdot,\ \cdot\rangle)$ and $v \in \mathbb{R}^r$, we find

$$\langle u,\ \Psi A v\rangle = \langle \Psi^* u,\ Av\rangle_{\mathbb{R}^r} = \langle A^T \Psi^* u,\ v\rangle_{\mathbb{R}^r}. \tag{3.129}$$

Therefore, we have verified that $A^T\Psi^* = (\Psi A)^*$. The verification of $A\Phi^* = (\Phi A^T)^*$ proceeds in the same way. Finally, we can conclude that

$$\left(T_{(\Phi,\Psi)}\mathcal{B}_{n,r}\right)^{\perp} = \text{Range}\left(\mathrm{D}\, F(\Phi,\Psi)^*\right) = \left\{\left(\Psi A, \Phi A^T\right) \in \mathcal{E}\ :\ A \in \mathbb{R}^{r\times r}\right\}. \tag{3.130}$$

The orthogonal projection $(\hat{X},\hat{Y}) = P_{(\Phi,\Psi)}(X,Y)$ is the unique element in $T_{(\Phi,\Psi)}\mathcal{B}_{n,r}$ such that $(X-\hat{X}, Y-\hat{Y}) \in \left(T_{(\Phi,\Psi)}\mathcal{B}_{n,r}\right)^{\perp}$. By the characterization of the orthogonal complement of the tangent space shown above, we know that

$$(X-\hat{X},\ Y-\hat{Y}) = (\Psi A,\ \Phi A^T) \tag{3.131}$$

for some matrix $A \in \mathbb{R}^{r\times r}$. Using the characterization of the tangent space shown above, the condition that $(\hat{X},\ \hat{Y}) = (X-\Psi A,\ Y-\Phi A^T) \in T_{(\Phi,\Psi)}\mathcal{B}_{n,r}$ means that A must satisfy

$$0 = \left(Y-\Phi A^T\right)^* \Phi + \Psi^*\left(X-\Psi A\right). \tag{3.132}$$

Rearranging, yields the Sylvester equation

$$A\left(\Phi^*\Phi\right) + \left(\Psi^*\Psi\right) A = Y^*\Phi + \Psi^* X. \tag{3.133}$$

To vectorize the Sylvester equation, we recall some facts about vectorized matrix products (see [146], [269]). If A and B are $r\times r$ matrices, then

$$\text{vec}(AB) = \begin{bmatrix} A\,\text{col}_1(B) \\ \vdots \\ A\,\text{col}_r(B) \end{bmatrix} = \begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix} \begin{bmatrix} \text{col}_1(B) \\ \vdots \\ \text{col}_r(B) \end{bmatrix} = (I_r \otimes A)\,\text{vec}(B). \tag{3.134}$$

The same product can be written in terms of the vectorization of B. We also observe that

$$\operatorname{vec}(AB) = \begin{bmatrix} \begin{pmatrix} B_{1,1}A_{1,1} + \cdots + B_{r,1}A_{1,r} \\ B_{1,1}A_{2,1} + \cdots + B_{r,1}A_{2,r} \\ \vdots \\ B_{1,1}A_{r,1} + \cdots + B_{r,1}A_{r,r} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} B_{1,r}A_{1,1} + \cdots + B_{r,r}A_{1,r} \\ B_{1,r}A_{2,1} + \cdots + B_{r,r}A_{2,r} \\ \vdots \\ B_{1,r}A_{r,1} + \cdots + B_{r,r}A_{r,r} \end{pmatrix} \end{bmatrix} = \begin{bmatrix} B_{1,1}I & \cdots & B_{r,1}I_r \\ \vdots & \ddots & \vdots \\ B_{1,r}I & \cdots & B_{r,r}I_r \end{bmatrix} \begin{bmatrix} \operatorname{col}_1(A) \\ \vdots \\ \operatorname{col}_r(A) \end{bmatrix}$$

$$= \left(B^T \otimes I_r\right) \operatorname{vec}(A). \quad (3.135)$$

Using these facts, we can vectorize the Sylvester equation, yielding the symmetric linear system

$$\left[(\Phi^*\Phi) \otimes I_r + I_r \otimes (\Psi^*\Psi)\right] \operatorname{vec}(A) = \operatorname{vec}\left(Y^*\Phi + \Psi^*X\right). \quad (3.136)$$

By Theorem. 13.12 in [146], the eigenvalues of $(\Phi^*\Phi) \otimes I_r$ are given by the eigenvalues of $\Phi^*\Phi$ with multiplicities increased by a factor of r. Likewise, the eigenvalues of $I_r \otimes (\Psi^*\Psi)$ are given by the eigenvalues of $\Psi^*\Psi$ with multiplicities increased by a factor of r. One can see this for general matrices A and B by taking an eigenvector u of A with eigenvalue λ and an eigenvector v of B with eigenvalue μ and observing that $u \otimes v$ is an eigenvector of $A \otimes B$ with eigenvalue $\lambda\mu$:

$$(A \otimes B)(u \otimes v) = (Au) \otimes (Bv) = \lambda\mu(u \otimes v). \quad (3.137)$$

If A and B are diagonalizable, then all eigenvectors of $A \otimes B$ are formed in this way. Therefore, taking the products of eigenvalues of A with the eigenvalues of B produces all of the eigenvalues of $A \otimes B$.

Since Φ and Ψ each have r linearly independent columns, we know that $\Phi^*\Phi$ and $\Psi^*\Psi$ are symmetric, positive-definite matrices. It then follows that $(\Phi^*\Phi) \otimes I_r$ and $I_r \otimes (\Psi^*\Psi)$ are symmetric, positive-definite matrices, and so must be their sum. Therefore, the solution of the Sylvester equation exists and gives the unique orthogonal projection of (X, Y) onto $T_{(\Phi,\Psi)}\mathcal{B}_{n,r}$. □

Proof of Theorem 3.4.6 (Second-order retraction). We begin by observing that the base point is pre-

served, i.e.,

$$R_{(\Phi,\Psi)}(0,0) = (\Phi,\ \Psi), \qquad \forall (\Phi,\Psi) \in \mathcal{B}_{n,r}. \tag{3.138}$$

Let us define the map $L : M \mapsto A$ by the unique solution of the Sylvester equation

$$A\left(\Phi^*\Phi\right) + \left(\Psi^*\Psi\right) A + M = 0 \tag{3.139}$$

and the map $g : (X,Y) \mapsto Y^*X$. We observe that L is a linear map and $\mathrm{D}\, g(0,0) = 0$, so

$$\mathrm{D}(L \circ g)(0,0)(V,W) = L\, \mathrm{D}\, g(0,0)(V,W) = 0, \qquad \forall (V,W) \in \mathcal{E}. \tag{3.140}$$

It is also clear that $L(0) = 0$. Therefore, the derivative of the retraction is given by

$$\mathrm{D}\, R_{(\Phi,\Psi)}(0,0)(V,W) = \left(V - \Phi\left[W^*\Phi + \Psi^*V\right],\ W\right) \tag{3.141}$$

Since $(V,W) \in T\mathcal{B}_{n,r}$, Theorem 3.4.2 tells us that $W^*\Phi + \Psi^*V = 0$ and so we find that R satisfies local rigidity

$$\mathrm{D}\, R_{(\Phi,\Psi)}(0,0)(V,W) = (V,W), \qquad \forall (V,W) \in T\mathcal{B}_{n,r}. \tag{3.142}$$

Hence, we have shown that R is a retraction and it remains to show that

$$\frac{d^2}{dt^2} R_{(\Phi,\Psi)}\left(t(X,Y)\right)\Big|_{t=0} = \left(\Psi B,\ \Phi B^T\right) \tag{3.143}$$

for some matrix $B \in \mathbb{R}^{r\times r}$ by Theorem 3.4.2. We observe that $L \circ g\left(t(X,Y)\right) = t^2 L(Y^*X)$ and define

$$H(t) = \left[\left(\Psi + tY + t^2\Phi L(Y^*X)^T\right)^* \left(\Phi + tX + t^2\Psi L(Y^*X)\right)\right]^{-1}. \tag{3.144}$$

Differentiating, we obtain

$$\begin{aligned}\frac{d}{dt}H(t) = -H(t)\Big[&\left(Y + 2t\Phi L(Y^*X)^T\right)^* \left(\Phi + tX + t^2\Psi L(Y^*X)\right) \\ &+ \left(\Psi + tY + t^2\Phi L(Y^*X)^T\right)^* \left(X + 2t\Psi L(Y^*X)\right)\Big] H(t)\end{aligned} \tag{3.145}$$

$$\frac{d}{dt}H(t)\Big|_{t=0} = -\left[Y^*\Phi + \Psi^*X\right] = 0, \qquad \forall (X,Y) \in T_{(\Phi,\Psi)}\mathcal{B}_{n,r} \tag{3.146}$$

and

$$\frac{d^2}{dt^2}H(t)\bigg|_{t=0} = -\left[2L(Y^*X)\Phi^*\Phi + Y^*X + Y^*X + 2\Psi^*\Psi L(Y^*X)\right] = 0, \tag{3.147}$$

by definition of L for every $(X,Y) \in T_{(\Phi,\Psi)}\mathcal{B}_{n,r}$. Using the above expressions, the retraction may be written as

$$R_{(\Phi,\Psi)}\left(t(X,Y)\right) = \left((\Phi + tX + t^2\Psi L(Y^*X))H(t),\ \Psi + tY + t^2\Phi L(Y^*X)^T\right) \tag{3.148}$$

and differentiated, giving

$$\frac{d}{dt}R_{(\Phi,\Psi)}\left(t(X,Y)\right) = \left((X + 2t\Psi L(Y^*X))\,H(t) + \left(\Phi + tX + t^2\Psi L(Y^*X)\right)\frac{d}{dt}H(t),\ Y + 2t\Phi L(Y^*X)^T\right). \tag{3.149}$$

Differentiating again and applying Theorem 3.4.2, we obtain

$$\frac{d^2}{dt^2}R_{(\Phi,\Psi)}\left(t(X,Y)\right)\bigg|_{t=0} = \left(2\Psi L(Y^*X),\ 2\Phi L(Y^*X)^T\right) \in \left(T_{(\Phi,\Psi)}\mathcal{B}_{n,r}\right)^{\perp}. \tag{3.150}$$

for every $(\Phi,\Psi) \in \mathcal{B}_{n,r}$ and $(X,Y) \in T_{(\Phi,\Psi)}\mathcal{B}_{n,r}$. This proves that R is a second-order retraction according to Definition 3.4.5.

Finally, the push-forward map is found by straightforward differentiation. □

Chapter 4

Models based on approximating Koopman operators

The Koopman operator provides an alternative viewpoint for dynamical systems by studying the evolution of functions on the state space. The most attractive feature of this theory is that the evolution of these functions is governed by a linear operator — the Koopman operator — even when the system has nonlinear state-space dynamics. This linearity provides us with a variety of tools, such as spectral analysis, for studying the system's behavior and building simplified models of the dynamics. In general, one seeks a finite dimensional subspace of functions that are rich enough to describe macroscopic quantities of interest about the system, while being invariant or nearly invariant under the action of the Koopman operator. The eigenfunctions spanning such an invariant subspace provide coherent observables of the system, which are in some sense dual to the coherent structures we discussed in Chapter 3. By placing the focus on functions with coherent time-evolution, we can study the aggregate behavior of trajectories that exhibit both chaotic and organized dynamics. However, because the Koopman operator is infinite-dimensional and may not have any non-trivial finite-dimensional invariant subspaces, a great deal of attention has been given to making meaningful finite-dimensional approximations of the Koopman operator. For more details, one can consult our review paper [196]* as well as [177] and [34]. This chapter describes some theoretical foundations for the Koopman operator as well as practical approximation techniques developed by S. E. Otto et al. that are applicable in a variety of engineering settings, including to actuated systems.

4.1 Koopman operators, generators, and function spaces

The trajectories of an autonomous dynamical system on a forward-invariant set $\mathcal{X}$ are described by the "flow map" $F^t : \mathcal{X} \to \mathcal{X}$ according to

$$x(s+t) = F^t(x(s)) \tag{4.1}$$

for any $x(s) \in \mathcal{X}$ and $t \in \mathbb{R}_+$ for continuous-time systems or $t \in \mathbb{N}$ for discrete-time systems. While the flow map may be nonlinear, B. O. Koopman [136] noticed that if we look instead at what happens to complex-valued functions of the state $\psi : \mathcal{X} \to \mathbb{C}$ called "observables" then composition with the flow map produces a new observable $\psi^\sharp = \psi \circ F^t : \mathcal{X} \to \mathbb{C}$ and this composition operation is linear. That is,

$$(\alpha_1\psi_1 + \alpha_2\psi_2) \circ F^t = \alpha_1\psi_1 \circ F^t + \alpha_2\psi_2 \circ F^t, \tag{4.2}$$

for every $\alpha_1, \alpha_2 \in \mathbb{C}$ and pair of observables ψ_1, ψ_2. Therefore, if we have a linear space of functions $\mathcal{F}$ that is closed under composition with the flow map $\psi \circ F^t \in \mathcal{F}$ for every $\psi \in \mathcal{F}$, then it is possible to define the "Koopman operator" on $\mathcal{F}$ according to

$$\boxed{U^t\psi := \psi \circ F^t, \qquad \forall \psi \in \mathcal{F}.} \tag{4.3}$$

As we have just shown, the Koopman operator associated with a possibly nonlinear flow F^t is a linear operator on a space of functions $\mathcal{F}$.

If the function space $\mathcal{F}$ contains observables $\psi_1, \psi_2, \ldots$ for which the state x can always be recovered as a function $x = h(\psi_1(x), \psi_2(x), \ldots)$ for all $x \in \mathcal{X}$ then the Koopman operator contains the same information as the original flow map because

$$F^t = h\left(\psi_1 \circ F^t, \psi_2 \circ F^t, \ldots\right) = h\left(U^t\psi_1, U^t\psi_2, \ldots\right). \tag{4.4}$$

In exchange for linearity in working with U^t instead of the nonlinear flow F^t, the function space $\mathcal{F}$ might have to be infinite-dimensional in order to satisfy Koopman invariance $U\mathcal{F} \subseteq \mathcal{F}$ and state reconstructability.

When the underlying system is measure-preserving, then a natural choice for the function space $\mathcal{F}$ are the square integrable functions with respect to the preserved measure. The flow map F^t preserves a measure μ on $\mathcal{X}$ if the pre-image of a measurable subset $A \subset \mathcal{X}$ has the same measure as A, that is, $\mu\left((F^t)^{-1}(A)\right) = \mu(A)$. In such a case, the Koopman operator defined on the Hilbert

space of square μ-integrable functions $\mathcal{F} = L^2((\mathcal{X}, \mu); \mathbb{C})$ is an isometry because

$$\|U^t f\|^2 = \int_{\mathcal{X}} |f \circ F^t|^2 \, \mathrm{d}\, \mu = \int_{\mathcal{X}} |f|^2 \, \mathrm{d}(\mu \circ (F^t)^{-1}) = \int_{\mathcal{X}} |f|^2 \, \mathrm{d}\, \mu = \|f\|^2. \tag{4.5}$$

Moreover, when F^t is invertible, then U^t is unitary because $f \circ (F^t)^{-1} \in L^2((\mathcal{X}, \mu); \mathbb{C})$ for every $f \in L^2((\mathcal{X}, \mu); \mathbb{C})$. This follows from essentially the same argument above and the fact that $\mu\,(F^t A) = \mu(A)$ for every measurable $A \in \mathcal{X}$ when F^t is invertible. By the spectral resolution of unitary operators (see Chapter 31 of P. D. Lax [147]), the Koopman operator may then be expressed as an integral over the unit circle in the complex plane

$$U^t = \int_0^{2\pi} e^{i\theta} \, \mathrm{d}\, E(\theta) \tag{4.6}$$

with respect to an orthogonal projection-valued measure E.

When the underlying system is not measure-preserving, then a natural choice for the function space on which to define the Koopman operator is less clear. When the factor by which the flow map F^t shrinks measurable sets according to a measure μ is bounded, then U^t may still be defined as a bounded operator on $\mathcal{F} = L^2((\mathcal{X}, \mu); \mathbb{C})$. In particular, if $\mu\left((F^t)^{-1}(A)\right) \leq c\mu(A)$ for every measurable subset $A \subset \mathcal{X}$ then

$$\|U^t f\|^2 = \int_{\mathcal{X}} |f \circ F^t|^2 \, \mathrm{d}\, \mu = \int_{F^t(\mathcal{X})} |f|^2 \, \mathrm{d}(\mu \circ (F^t)^{-1}) \leq c \int_{\mathcal{X}} |f|^2 \, \mathrm{d}\, \mu = c\|f\|^2. \tag{4.7}$$

However, the choice of μ is ambiguous and we have lost the normality of U^t, which enabled a spectral resolution of the Koopman operator in the case of a measure preserving flow map.

When the flow map is continuous, another option is to define the Koopman operator over a Banach space of continuous functions such as $\mathcal{F} = C(\mathcal{X})$ or $\mathcal{F} = C_0(\mathcal{X})$, the closure of compactly supported functions in $C(\mathcal{X})$. An advantage of this choice for $\mathcal{F}$ is that functions in this space can be evaluated point-wise — which is generally an essential requirement for any data-driven approximation method. When the set $\mathcal{X}$ is merely forward-invariant, then

$$\|U^t f\| = \sup_{x \in \mathcal{X}} f(F^t(x)) = \sup_{x \in F^t(\mathcal{X})} f(x) \leq \sup_{x \in \mathcal{X}} f(x) = \|f\|, \tag{4.8}$$

where the inequality becomes equality if $F^t(\mathcal{X}) = \mathcal{X}$. Hence, the Koopman operator on $C(\mathcal{X})$ or on $C_0(\mathcal{X})$ is an isometry when F^t is surjective; and if F^t is invertible, then the Koopman operator is also invertible.

When U^t is defined on the space of continuous functions $C(\mathcal{X})$, then the adjoint operator $(U^t)^*$ is defined on the dual space of $C(\mathcal{X})$. The dual Banach space of $C(\mathcal{X})$ is the Banach space of Radon measures, denoted $\mathcal{M}(\mathcal{X})$, which contains the probability measures on Borel subsets of the state space. If $\mu \in \mathcal{M}(\mathcal{X})$ is a Radon measure, then for every $f \in C(\mathcal{X})$ we have

$$\langle (U^t)^* \mu, f \rangle = \langle \mu, U^t f \rangle = \int_{\mathcal{X}} f \circ F^t \, \mathrm{d}\, \mu = \int_{F^t(\mathcal{X})} f \, \mathrm{d}(\mu \circ (F^t)^{-1}) = \langle \mu \circ (F^t)^{-1}, f \rangle. \tag{4.9}$$

Therefore, the adjoint operator $(U^t)^*$ acts on a Radon measure μ to produce the measure of pre-image sets defined by

$$(U^t)^* \mu(A) = \mu \left((F^t)^{-1}(A) \right) \tag{4.10}$$

for every Borel measurable subset $A \subset \mathcal{X}$. Probability measures are the set of positive Radon measures with unit norm, and the adjoint operator $(U^t)^*$ acts to transport probability measures under the flow F^t of the system. That is, if the measure μ describes the probability of finding the initial state of the system in Borel subsets, then the measure $(U^t)^* \mu$ describes the probability of finding the state in Borel subsets after evolving the system according to F^t. The operator $(U^t)^*$, which transports probability measures under the flow of the system, is called the Perron-Frobenius operator. Finally, if F^t is invertible, then $(U^t)^*$ is an invertible isometry because U^t is an invertible isometry.

So far we have not considered systems with actuation or control. Preservation of a fixed measure is often destroyed when control is applied to the system, and our understanding of the Koopman operator outside of the measure-preserving setting is limited. Efforts over the last five years including [211, 279, 255, 101, 138, 201] have sought to introduce actuation and control in the Koopman framework. The Koopman generator, discussed next, has emerged as one of the primary tools for doing this in a systematic way.

4.1.1 Koopman Generator

In this section, we consider a state space defined by a smooth manifold $\mathcal{X}$ and continuous time flow maps $F : \mathcal{X} \times [0, \infty) \to \mathcal{X}$. We shall often refer to the flow map at time t as $F^t : x \mapsto F(x, t)$. The family of Koopman operators $\{U^t\}_{t \geq 0}$ inherits the properties of a semigroup from the flow map,

namely

$$F^0(x) = x,\ \forall x \in \mathcal{X} \quad \Rightarrow \quad U^0\psi = \psi,\ \forall \psi \in \mathcal{F} \tag{4.11}$$

$$F^{t+s} = F^t \circ F^s,\ \forall s,t \geq 0 \quad \Rightarrow \quad U^{t+s} = U^t U^s,\ \forall s,t \geq 0. \tag{4.12}$$

If the flow map F^t is inverible, then the Koopman operator U^t is also invertible, making $\{U_t\}_{t\in\mathbb{R}}$ into a group. This is not always the case, for instance if the Koopman operator is being defined for a flow map on $\mathbb{R}^n$ restricted to a trapping region $\mathcal{X} \subset \mathbb{R}^n$. Here, $F^t(\mathcal{X})$ may be a proper subset of $\mathcal{X}$ and so $F^t : \mathcal{X} \to \mathcal{X}$ will fail to be surjective.

If the Koopman semigroup $\{U^t\}_{t\geq 0}$ is strongly continuous, i.e.,

$$U^t f \to f \quad \text{as} \quad t \to 0 \qquad \forall f \in \mathcal{F}, \tag{4.13}$$

then it is uniquely determined by its infinitesimal generator $V : \mathrm{Dom}(V) \to \mathcal{F}$ defined by

$$\boxed{Vf = \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, U^t f\Big|_{t=0} = \lim_{t\to 0} \frac{U^t f - f}{t}.} \tag{4.14}$$

The domain of the generator $\mathrm{Dom}\, V$ consists of those $f \in \mathcal{F}$ for which the above limit converges strongly, and it is dense in $\mathcal{F}$. Moreover, V is closed and commutes with every U^t in the sense that if $f \in \mathrm{Dom}\, V$ then $U^t f \in \mathrm{Dom}(V)$ and $VU^t f = U^t V f$. Consequently, U^t can be recovered from V by solving

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, U^t f = VU^t f \qquad f \in \mathrm{Dom}(V) \tag{4.15}$$

and extending U^t to all of $\mathcal{F}$ using the density of $\mathrm{Dom}(V)$ in $\mathcal{F}$ and boundedness of U^t. The solution is given explicitly by Hille's exponential formula (Theorem 8.3 in Section 1.8 of A. Pazy [199])

$$\boxed{U^t = \exp(tV) := \lim_{n\to\infty} \left(I - \frac{t}{n}V\right)^{-n} \qquad t \geq 0,} \tag{4.16}$$

which converges strongly and corresponds to an implicit time stepping scheme for solving Eq. 4.15. For more details on strongly continuous semigroups of operators, see Chapter 34 in P. D. Lax [147] and the books by A. Pazy [199] and K.-J. Engel and R. Nagel [86].

Strong continuity is not a strong requirement, as the following Theorem 4.1.1 shows that any continuous, proper flow F produces a strongly continuous Koopman semigroup on the function space $\mathcal{F} = C_0(\mathcal{X})$. Recall that $C_0(\mathcal{X})$ is the closure of the compactly supported continuous functions $C_c(\mathcal{X})$

in $C(\mathcal{X})$, that is, with respect to the sup-norm

$$\|f\| = \sup_{x \in \mathcal{X}} |f(x)|. \tag{4.17}$$

The requirement that $\tilde{F} : (x,t) \mapsto (F(x,t),t)$ is proper says that the pre-image under $\tilde{F}$ of any compact subset of $\mathcal{X} \times [0,\infty)$ is compact. This is automatically true when $\mathcal{X}$ is compact. Anther case when $\tilde{F}$ is proper is when each F^t has a continuous inverse since this implies that $\tilde{F}$ has a continuous inverse. The preimage of a compact set under $\tilde{F}$ is given by its image under $\tilde{F}^{-1}$, which is compact because $\tilde{F}^{-1}$ is continuous.

Theorem 4.1.1. *If $F : \mathcal{X} \times [0,\infty) \to \mathcal{X}$ is continuous and the map $\tilde{F} : (x,t) \mapsto (F(x,t),t)$ is proper, then the corresponding Koopman semigroup $\{U^t\}_{t \geq 0}$ is well-defined and strongly continuous on $C_0(\mathcal{X})$. If, in addition, the flow map F is continuously differentiable, then the infinitesimal generator $V : \mathrm{Dom}(V) \to C_0(\mathcal{X})$ of the Koopman semigroup is given by the closure of*

$$\begin{aligned} \tilde{V} : C_c^1(\mathcal{X}) &\to C_0(\mathcal{X}) \\ f &\mapsto \left.\frac{\partial F}{\partial t}\right|_{t=0} \cdot \nabla f \end{aligned} \tag{4.18}$$

in the graph norm $\|f\|_{\tilde{V}} = \|f\| + \|\tilde{V} f\|$.

Proof. We provide the proof in Appendix 4.A. □

Remark 4.1.2. *Essentially the same argument used in the proof of Theorem 4.1.1 can be used to prove the analogous result when U^t are bounded operators on $L^2(\mathcal{X})$, the key fact being that $C_c(\mathcal{X})$ is dense in $L^2(\mathcal{X})$.*

The following Theorem 4.1.3 characterizes the generators of contraction semigroups, that is, semigroups that do not increase the norm on $\mathcal{F}$. As we saw earlier, when the Koopman operator is defined over a Banach space of continuous functions, it is automatically a contraction semigroup. Therefore, Theorem 4.1.3 will provide useful information about the spectrum of the generator when U^t is strongly continuous.

Theorem 4.1.3 (Hille-Yosida, theorem 2.3.5 in [86])**.** *Let $V : \mathrm{Dom}(V) \to \mathcal{F}$ be a linear operator on a Banach space $\mathcal{F}$. Then the following properties are equivalent:*

1. *V generates a strongly continuous semigroup $U^t : \mathcal{F} \to \mathcal{F}$ of contractions, i.e., $\|U^t\| \leq 1$ for every $t > 0$*

2. *V is closed,* $\mathrm{Dom}(V)$ *is dense in* $\mathcal{F}$*, and for every* $\lambda \in \mathbb{R}$ *with* $\lambda > 0$*, the operator* $(\lambda I - V) : \mathrm{Dom}(V) \to \mathcal{F}$ *is invertible and*

$$\left\|\lambda(\lambda I - V)^{-1}\right\| \leq 1 \tag{4.19}$$

3. *V is closed,* $\mathrm{Dom}(V)$ *is dense in* $\mathcal{F}$*, and for every* $\lambda \in \mathbb{C}$ *with* $\mathrm{Re}(\lambda) > 0$*, the operator* $(\lambda I - V) : \mathrm{Dom}(V) \to \mathcal{F}$ *is invertible and*

$$\left\|(\lambda I - V)^{-1}\right\| \leq \frac{1}{\mathrm{Re}(\lambda)}. \tag{4.20}$$

Below, we state a corollary of Theorem 4.1.3 that characterizes the generators of groups of invertible isometries. Recall that when the flow map F^t is continuous and invertible, then U^t defined on a Banach space of continuous functions is a group of isometries. Consequently, if it can be shown that U^t is strongly continuous, then Corollary 4.1.4 says that the Koopman generator V has purely imaginary spectrum.

Corollary 4.1.4 (Corollary 2.3.7 in [86])**.** *Let* $V : \mathrm{Dom}(V) \to \mathcal{F}$ *be a linear operator on a Banach space* $\mathcal{F}$*. Then the following properties are equivalent:*

1. *V generates a strongly continuous group* $U^t : \mathcal{F} \to \mathcal{F}$ *of invertible isometries*

2. *V is closed,* $\mathrm{Dom}(V)$ *is dense in* $\mathcal{F}$*, and for every* $\lambda \in \mathbb{R} \setminus \{0\}$*, the operator* $(\lambda I - V) : \mathrm{Dom}(V) \to \mathcal{F}$ *is invertible and*

$$\left\|\lambda(\lambda I - V)^{-1}\right\| \leq 1 \tag{4.21}$$

3. *V is closed,* $\mathrm{Dom}(V)$ *is dense in* $\mathcal{F}$*, and for every* $\lambda \in \mathbb{C} \setminus i\mathbb{R}$*, the operator* $(\lambda I - V) : \mathrm{Dom}(V) \to \mathcal{F}$ *is invertible and*

$$\left\|(\lambda I - V)^{-1}\right\| \leq \frac{1}{|\,\mathrm{Re}(\lambda)|}. \tag{4.22}$$

Finally, Stone's theorem, stated below, provides a complete description of unitary semigroups on Hilbert spaces in terms of skew-adjoint generators. This result is especially important in the setting where F^t is invertible and measure-preserving — giving rise to a unitary Koopman group U^t on the Hilbert space $L^2((\mathcal{X}, \mu); \mathbb{C})$.

Theorem 4.1.5 (Stone, theorem 2.3.24 in [86])**.** *Let* $V : \mathrm{Dom}(V) \to \mathcal{F}$ *be a densely defined operator on a Hilbert space* $\mathcal{F}$*. Then* V *generates a strongly continuous group of unitary operators* $U^t : \mathcal{F} \to \mathcal{F}$ *if and only if* V *is skew-adjoint, i.e.,* $V^* = -V$*.*

The form of the Koopman generator expressed in Theorem 4.1.1 is especially useful for flows generated by ordinary differential equations with control input. In particular, suppose that the dynamics of the state $x \in \mathcal{X}$ are governed by

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, x = f(x, u). \tag{4.23}$$

Then, when the input is held constant, these dynamics generate a family of flow maps F_u^t and Koopman semigroups $\{U_u^t\}_{t\geq 0}$ on $\mathcal{F} = C_0(\mathcal{X})$ parametrized by the input level u. The Koopman generators are given by the closures of

$$\tilde{V}_u \psi = f(\cdot, u) \cdot \nabla\, \psi, \qquad \psi \in C_c^1(\mathcal{X}), \tag{4.24}$$

which have similar input dependence to f. For instance, when the dynamics are input-affine

$$f(x, u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x), \tag{4.25}$$

then the Koopman generator is also input affine

$$V_u = V_0 + \sum_{i=1}^{m} u_i V_i \tag{4.26}$$

over $\bigcap_{i=1}^m \mathrm{Dom}(V_i) \subset \mathrm{Dom}(V_u)$, where V_i is the Koopman generator associated with each component vector field f_i.

Remark 4.1.6. *In general, the domain of V_u may be strictly larger than the intersection of the component generators $\bigcap_{i=1}^m \mathrm{Dom}(V_i)$. For instance, consider $f_1, f_2 : \mathbb{R}^2 \to \mathbb{R}^2$ defined by $f_1(x) = (1, 0)$ and $f_2(x) = (0, 1)$. The function $\psi \in C_0(\mathbb{R}^2)$ defined by*

$$\psi(x_1, x_2) = |x_2 - x_1| e^{-x_1^2 - x_2^2} \tag{4.27}$$

is not in $\mathrm{Dom}(V_1)$, nor in $\mathrm{Dom}(V_2)$, yet ψ is in the domain of the Koopman generator $V_{(1,1)}$ of the vector field $f = f_1 + f_2 = (1, 1)$ because

$$\begin{aligned} V_{(1,1)}\psi(x_1, x_2) &= \lim_{t\to 0} \frac{1}{t} \left[|(x_2 + t) - (x_1 + t)| e^{-(x_1+t)^2 - (x_2+t)^2} - |x_2 - x_1| e^{-x_1^2 - x_2^2} \right] \\ &= |x_2 - x_1| e^{-x_1^2 - x_2^2} \left(-2x_1 - 2x_2 \right), \end{aligned} \tag{4.28}$$

where the limit converges uniformly over $\mathbb{R}^2$*. In such cases, the graph of* V_u *is the closure of the graph of* $V_0 + \sum_{i=1}^m u_i V_i$*. As we will see next, this distinction does not matter from the perspective of the exponential map.*

On the other hand, the Koopman operators U_u^t depend on the input in a much more complicated way than the Koopman generators. In particular, this dependence is captured by the exponenial map

$$U_u^t = \exp(tV_u) = \exp\left[t\left(V_0 + \sum_{i=1}^m u_i V_i\right)\right], \tag{4.29}$$

where the term on the right is understood using extension by continuity. In particular, the exponential map defined by Eq. 4.16 still makes sense for the sum of generators defined over $\bigcap_{i=1}^m \mathrm{Dom}(V_i)$ because the resolvent operators appearing in Eq. 4.16 can be understood using the extension by continuity provided by Lemma 4.1.7 below. Recall that a "core" of an operator is a subset of its domain that is dense in the graph norm and that Theorem 4.1.1 provides us with a core, $C_c^1(\mathcal{X})$, that is shared by every Koopman generator associated with a smooth vector field on $\mathcal{X}$.

Lemma 4.1.7 (Resolvents of sums with a common core)**.** *Let* $\mathcal{C} \subset \mathcal{F}$ *be a core for* V_1*,* V_2*, and* V*. If* $Vf = V_1 f + V_2 f$ *for every* $f \in \mathcal{C}$ *and* λ *is in the resolvent set of* V*, i.e.,* $\lambda I - V : \mathrm{Dom}(V) \to \mathcal{F}$ *is bijective, then*

$$(\lambda I - V)^{-1} = (\lambda I - (V_1 + V_2))^{-1}, \tag{4.30}$$

where the term on the right is understood as the extension by continuity of $(\lambda I - (V_1 + V_2))^{-1}$ *on the dense subset* $(\lambda I - (V_1 + V_2))(\mathcal{C})$ *of* $\mathcal{F}$*.*

Proof. We have $(\lambda I - V)^{-1} g = (\lambda I - (V_1 + V_2))^{-1} g$ for every $g \in (\lambda I - (V_1 + V_2))(\mathcal{C})$. By the open mapping theorem, the resolvent operator $(\lambda I - V)^{-1}$ is bounded, and so it suffices to prove that $(\lambda I - (V_1 + V_2))(\mathcal{C})$ is dense in $\mathcal{F}$. Choose any $f \in \mathcal{F}$ and take $x_0 \in \mathrm{Dom}(V)$ such that $(\lambda I - V)x_0 = f$. Since $\mathcal{C}$ is a core for V, there is a sequence $\{x_n\}_{n=1}^\infty \subset \mathcal{C}$ such that $x_n \to x_0$ and $Vx_n \to Vx_0$. Therefore, we have

$$(\lambda I - (V_1 + V_2))x_n = \lambda x_n - (V_1 + V_2)x_n = \lambda x_n - Vx_n \to \lambda x_0 - Vx_0 = f. \tag{4.31}$$

□

4.1.2 An augmented unitary Koopman group

Principled approximation techniques yielding powerful convergence guarantees for the spectra of unitary groups have been developed recently by S. Das et al. [75]. However, these techniques cannot be applied directly to Koopman operators in the non-measure preserving setting since the resulting operators are not unitary or even normal. One way to skirt around this problem might be to study an augmented Koopman operator that is unitary on $\mathcal{F} = L^2((\mathcal{X}, \mu); \mathbb{C})$. Such an operator would then admit convergent finite-dimensional approximations using the RKHS compactification technique developed by S. Das et al. [75] even when the system is not measure-preserving, for instance, when there is actuation. We construct such an augmented Koopman operator here.

Suppose that μ is a σ-finite measure, $\mu \circ (F^t)^{-1}$ is absolutely continuous with respect to μ. Let $w_t = \frac{\mathrm{d}\,\mu \circ (F^t)^{-1}}{\mathrm{d}\,\mu}$ be the Radon-Nikodym derivative of $\mu \circ (F^t)^{-1}$ with respect to μ, i.e.,

$$\mu\left((F^t)^{-1}(A)\right) = \int_A w_t \,\mathrm{d}\,\mu \tag{4.32}$$

for every measurable subset A. For instance, if μ has density $\rho > 0$ with respect to the Lebesgue measure on an open subset $\mathcal{X} \subset \mathbb{R}^n$ and F^t is a diffeomorphism of $\mathcal{X}$, then

$$w_t(x) = \frac{\rho\left((F^t)^{-1}(x)\right)}{\rho(x)} \left|\det\left(\mathrm{D}(F^t)^{-1}(x)\right)\right|. \tag{4.33}$$

If $w_t > 0$ μ-almost everywhere, then we can define an augmented Koopman operator $\tilde{U}^t$ on $\mathcal{F} = L^2((\mathcal{X}, \mu); \mathbb{C})$ according to

$$\boxed{\tilde{U}^t f = \left(\frac{f}{\sqrt{w_t}}\right) \circ F^t = \frac{f \circ F^t}{\sqrt{w_t \circ F^t}}.} \tag{4.34}$$

We observe that this augemented Koopman operator agrees with the original Koopman operator whenever the flow map preserves μ.

The key property of $\tilde{U}^t$ is that it is unitary when F^t is invertible, even when F^t is not measure-preserving. To see this, we observe that $\tilde{U}^t$ is an isometry when F^t is invertible because

$$\|\tilde{U}^t f\|^2 = \int_{\mathcal{X}} \frac{|f \circ F^t|^2}{w_t \circ F^t} \,\mathrm{d}\,\mu = \int_{\mathcal{X}} \frac{|f|^2}{w_t} \,\mathrm{d}\,\mu \circ F^{-1} = \int_{\mathcal{X}} \frac{|f|^2}{w_t} w_t \,\mathrm{d}\,\mu = \|f\|^2. \tag{4.35}$$

Finally, the inverse operator $(\tilde{U}^t)^{-1} f = \sqrt{w_t}(f \circ (F^t)^{-1}) = (\tilde{U}^t)^*$ is an isometry because

$$\|(U^t)^{-1} f\| = \int_{\mathcal{X}} |f \circ (F^t)^{-1}|^2 w_t \,\mathrm{d}\,\mu = \int_{\mathcal{X}} |f \circ (F^t)^{-1}|^2 \,\mathrm{d}\,\mu \circ (F^t)^{-1} = \int_{\mathcal{X}} |f|^2 \,\mathrm{d}\,\mu = \|f\|^2 \tag{4.36}$$

for every $f \in L^2((\mathcal{X}, \mu); \mathbb{C})$.

When the flow F^t is a diffeomorphism produced by integrating an ordinary differential equation

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, x = f(x) \tag{4.37}$$

on an open subset $\mathcal{X} \subset \mathbb{R}^n$, then the generator $\tilde{V} : \mathrm{Dom}(\tilde{V}) \to L^2(\mathcal{X})$ of the augmented Koopman group $\{\tilde{U}^t\}_{t\in\mathbb{R}}$ can be written explicitly over a core. Here, we assume that the measure μ on $\mathcal{X}$ has continuously differentiable density $\rho > 0$ with respect to the Lebesgue measure. A simple calculation shows that when f is continuously differentiable, then for each $\psi \in C_c^1(\mathcal{X})$, we have

$$\boxed{\tilde{V}\psi = \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} \left(\tilde{U}^t \psi\right)\Big|_{t=0} = \frac{1}{2}\left[\rho^{-1} f \cdot \nabla\, \rho + \mathrm{Tr}\,(\mathrm{D}\, f)\right] \psi + f \cdot \nabla\, \psi,} \tag{4.38}$$

where the second term is the action of the usual Koopman generator V on ψ. The space $C_c^1(\mathcal{X})$ is a core for $\tilde{V}$ because it is dense in $\mathcal{F} = L^2(\mathcal{X})$ and invariant under $\tilde{U}^t$ (see Proposition 1.7 in Chapter 2 of [86]). Moreover, by Stone's theorem (Theorem 4.1.5), we know that $\tilde{V} = -\tilde{V}^*$ is skew-adjoint.

Proof of Eq. 4.38. Since F^t is a flow map, it preserves orientation, and by Eq. 4.33 we have

$$\frac{1}{w_t(F^t(x))} = \frac{\rho\,(F^t(x))}{\rho(x)} \det\left(\mathrm{D}\, F^t(x)\right). \tag{4.39}$$

Differentiating with respect to time and recalling that $w_0 \equiv 1$, we find

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} \left(\frac{1}{\sqrt{w_t(F^t(x))}}\right)\Big|_{t=0} = \frac{1}{2}\, \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} \left(\frac{1}{w_t(F^t(x))}\right)\Big|_{t=0} = \frac{1}{2}\left[\frac{f(x) \cdot \nabla\, \rho(x)}{\rho(x)} + \mathrm{Tr}\,(\mathrm{D}\, f(x))\right]. \tag{4.40}$$

We now obtain Eq. 4.38 by differentiating Eq. 4.34 with respect to time and substituting the above expression. □

Studying the augmented Koopman semigroup may be useful when the system of interest depends on a parameter or input signal. Even if the system is measure-preserving at each parameter value or input level, the same measure might not be preserved across parameter values or input levels. When the flow maps are diffeomorphisms, then the input paramterized augmented Koopman operators form a family of unitary groups. Moreover, when the flow is generated by an input-affine vector field

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, x = f(x) + \sum_{i=1}^{m} u_i f_i(x) \tag{4.41}$$

then the skew-adjoint generators of the augmented Koopman groups $\{\tilde{U}_u^t\}_{t\in\mathbb{R}}$ are also input-affine

$$\tilde{V}_u\psi = \tilde{V}_0\psi + \sum_{i=1}^{m} u_i\tilde{V}_i\psi, \qquad \psi \in \bigcap_{i=0}^{m} \mathrm{Dom}(\tilde{V}_i), \tag{4.42}$$

where $\tilde{V}_i$ is the augmented Koopman generator associated with each component vector field f_i. It may now be possible to apply the RKHS compactification technique developed by S. Das et al. [75] to make convergent finite-dimensional approximations of the augmented Koopman groups for actuated systems.

If the constant function 1 is an element of $\mathcal{F} = L^2((\mathcal{X},\mu);\mathbb{C})$, then it is possible recover the original Koopman group and its generator from the augmented unitary Koopman group $\tilde{U}^t$ and its skew-adjoint generator $\tilde{V}$. In particular, we can recover the action of the original Koopman operator on an observable f according to

$$U^t f = \left(\frac{1}{\tilde{U}^t 1}\right)\tilde{U}^t f. \tag{4.43}$$

The action of the Koopman generator is recovered according to

$$V\psi = \tilde{V}\psi - (\tilde{V}1)\psi. \tag{4.44}$$

This makes it possible to approximate the Koopman operator and generator indirectly by first approximating the augmented operators using principled techniques developed in the unitary setting, such as the one described in [75].

4.2 Coherent observables and structures

In this section, we review some important connections between the Koopman operator and the state space dynamics. Here, we are primarily concerned with invariant subspaces of the Koopman operator and generator, as well as expansions of relevant observables in an eigenfunction basis. Whether or not the Koopman operator has any eigenfunctions depends both on the dynamics and on the choice of function space. In many cases, such as chaotic dynamics, there are no non-trivial finite-dimensional Koopman-invariant subspaces. On the other hand, the Koopman operator may have eigenfunctions when defined over one function space, but not over another. Our discussion in this section does not require the Koopman operator to be defined on a normed space, so it will be convenient to work with various spaces of functions that are defined point-wise and may not be bounded. The Koopman generator can then be defined with respect to the topology of point-wise

convergence in this function space. This flexibility in the definition of the Koopman operator enables the global linearization of dissipative systems discussed below in Section 4.2.1.

4.2.1 Global Linearization

An important area of research concerns identifying the class of systems that are known to be described by the evolution of observables in a finite-dimensional Koopman-invariant subspace. While a complete description of this class of systems in terms of state space geometry is not yet known, the work of Y. Lan and I. Mezić [145] makes progress in this direction by extending the Hartman-Grobman theorem to the entire basins of attraction of linearly stable fixed points and limit cycles.

To sketch the main idea of Y. Lan and I. Mezić [145], consider a twice continuously differentiable system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x})$ on $\mathbb{R}^n$ where $\boldsymbol{g}(\boldsymbol{0}) = \boldsymbol{0}$, $\mathrm{D}\,\boldsymbol{g}(\boldsymbol{0}) = \boldsymbol{0}$, and all eigenvalues of $\boldsymbol{A}$ have strictly negative real part. Then the Hartman-Grobman theorem says that there is a C^1 diffeomorphism $\tilde{\boldsymbol{\psi}}$ with $\mathrm{D}\,\tilde{\boldsymbol{\psi}}(\boldsymbol{0}) = \boldsymbol{I}$ on a neighborhood of the origin so that the dynamics are conjugate to $\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z}$ with $\boldsymbol{z} = \tilde{\boldsymbol{\psi}}(\boldsymbol{x})$ in this neighborhood. Taking Σ to be a level surface of a strictly decreasing quadratic Lyapunov function contained within the neighborhood, Y. Lan and I. Mezić [145] observe that for every $\boldsymbol{x}$ in the basin of attraction, there are unique and smoothly varying $t_\Sigma(\boldsymbol{x}) \in \mathbb{R}$ and $\boldsymbol{x}_\Sigma(\boldsymbol{x}) \in \Sigma$ so that $\boldsymbol{x} = \boldsymbol{F}^{t_\Sigma(\boldsymbol{x})}(\boldsymbol{x}_\Sigma(\boldsymbol{x}))$. Therefore, they conclude that the dynamics are conjugate to

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\boldsymbol{\psi}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{\psi}(\boldsymbol{x}), \qquad \boldsymbol{\psi}(\boldsymbol{x}) := e^{\boldsymbol{A}t_\Sigma(\boldsymbol{x})}\tilde{\boldsymbol{\psi}}\left(\boldsymbol{F}^{-t_\Sigma(\boldsymbol{x})}(\boldsymbol{x})\right), \tag{4.45}$$

over the entire basin of attraction, where $\boldsymbol{\psi}$ is a diffeomorphism that extends $\tilde{\boldsymbol{\psi}}$.

The observables $\psi_i := [\boldsymbol{\psi}]_i$, $i = 1, \ldots, n$ provided by each component of $\boldsymbol{\psi}$ span a Koopman-invariant subspace. This holds because any observable $\psi_{\boldsymbol{v}} := v_1\psi_1 + \cdots + v_n\psi_n = \boldsymbol{v}^T\boldsymbol{h}$ evolves according to

$$V\left(v_1\psi_1 + \cdots + v_n\psi_n\right) = \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\left(\boldsymbol{v}^T\boldsymbol{\psi}\right) = \boldsymbol{v}^T\boldsymbol{A}\boldsymbol{\psi} = [\boldsymbol{A}^T\boldsymbol{v}]_1\psi_1 + \cdots + [\boldsymbol{A}^T\boldsymbol{v}]_n\psi_n, \tag{4.46}$$

and so it stays in the subspace. Analogous results are also developed in [145] for discrete-time maps and limit cycle dynamics, in which case the conjugate linear system is the one provided by the Floquet transformation of the linearized dynamics around the limit cycle [145]. These results have also been extended to globally linearize dynamics in the basins of attracting quasi-periodic tori [178].

We have just seen how finite-dimensional Koopman-invariant subspaces can always be found to globally linearize dynamics in the basins of attractive, hyperbolic fixed points or limit cycles.

Moreover, the eigenvalues of the resulting linear systems agree with the eigenvalues of the linearized dynamics at the fixed point or with the Floquet exponents in the case of a limit cycle. The work of A. Mauroy and co-authors [173, 171] shows that important information about the state space geometry can be obtained by considering the level sets of the associated (generalized) Koopman eigenfunctions to be discussed in the next section.

4.2.2 Koopman Eigenfunctions and Modes

Koopman Eigenfunctions: Coordinate Systems, Spectral Lattice, and Stability

An eigenfunction of the Koopman generator $V : \mathrm{Dom}(V) \to \mathcal{F}$ with eigenvalue λ is a nonzero observable $\varphi \in \mathrm{Dom}(V)$ for which $V\varphi = \lambda\varphi$, that is an observable in the null space of $(V - \lambda I)$. If φ is an eigenfunction of V with eigenvalue λ, then φ is an eigenfunction of the Koopman operator U^t for every $t \geq 0$ with eigenvalue $e^{\lambda t}$. Following Section 34.5 of [147], this holds because

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\left(e^{-\lambda t}U^t\varphi\right) = e^{-\lambda t}U^t(V\varphi - \lambda\varphi) = 0, \qquad e^0U^0\varphi = \varphi, \tag{4.47}$$

which implies that $e^{-\lambda t}U^t\varphi = \varphi$ for all $t \geq 0$. Conversely, if $\varphi \in \mathcal{F}$ is an eigenfunction of U^t with eigenvalue $e^{\lambda t}$ for every $t \geq 0$, then $\varphi \in \mathrm{Dom}(V)$ is an eigenfunction V. Note that it is not enough for φ to be an eigenfunction of U^t for a single $t > 0$. For instance, consider the dynamics $F^t(x) = x + t$ on $\mathbb{R}^1$ and the function $\varphi(x) = e^{ix} + e^{2ix}$, which is an eigenfunction of $U^{2\pi}$ with eigenvalue 1. However φ is not an eigenfunction of V because $(V\varphi)(x) = ie^{ix} + 2ie^{2ix} \neq \lambda\varphi(x)$ for any constant λ. There may also be eigenfunctions of U^t with eigenvalue zero. These eigenfunctions cannot be eigenfunctions of the Koopman generator because $e^{\lambda t}$ can never be zero.

When $\mathcal{F}$ is closed under point-wise products of functions, then the eigenvalues and eigenfunctions of the Koopman operator and generator form a lattice in the complex plane. In particular, if φ_1 and φ_2 are eigenfunctions of the Koopman operator U^t with eigenvalues μ_1 and μ_2, then $\varphi_1\varphi_2$ is also an eigenfunction of U^t with eigenvalue $\mu_1\mu_2$. This is verified using the definition of the Koopman operator

$$U^t(\varphi_1\varphi_2) = (\varphi_1 \circ F^t)(\varphi_2 \circ F^t) = (U^t\varphi_1)(U^t\varphi_2) = \mu_1\mu_2(\varphi_1\varphi_2). \tag{4.48}$$

A similar property also holds for eigenfunctions of the Koopman generator thanks to the following "product rule" for Koopman generators:

Lemma 4.2.1 (Product rule for Koopman generators)**.** *Suppose that $\mathcal{F}$ is closed under point-wise multiplication of functions and suppose that the Koopman generator is defined with respect to point-*

wise convergence. For any $\psi_1, \psi_2 \in \mathrm{Dom}(V)$*, we have* $\psi_1\psi_2 \in \mathrm{Dom}(V)$ *and*

$$V(\psi_1\psi_2) = \psi_2 V\psi_1 + \psi_1 V\psi_2. \tag{4.49}$$

Proof. The proof is essentially the same as the proof of the product rule in calculus. For completeness, we give the details in Appendix 4.A. □

As a consequence of Lemma 4.2.1, if we have two eigenfunctions $\varphi_1, \varphi_2 \in \mathrm{Dom}(V)$ of the Koopman generator with eigenvalues λ_1, λ_2 respectively, then $\varphi_1\varphi_2 \in \mathrm{Dom}(V)$ is an eigenfunction of the Koopman generator with eigenvalue $\lambda_1 + \lambda_2$. To see this, we compute

$$V(\varphi_1\varphi_2) = \varphi_1 V\varphi_2 + \varphi_2 V\varphi_2 = \lambda_2\varphi_1\varphi_2 + \lambda_1\varphi_1\varphi_2 = (\lambda_1 + \lambda_2)\varphi_1\varphi_2. \tag{4.50}$$

Suppose we have a collection of observables $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_d)$ that span a Koopman-invariant subspace with dynamics given by $\dot{\boldsymbol{\psi}} = \boldsymbol{A}\boldsymbol{\psi}$, where $\boldsymbol{A}$ is diagonalizable. Then each eigenvector $\boldsymbol{v}_i$ of $\boldsymbol{A}^T$ with eigenvalue λ_i gives rise to a Koopman eigenfunction $\varphi_i = \boldsymbol{v}_i^T\boldsymbol{\psi}$ with the same eigenvalue λ. In the particular case of a stable, hyperbolic fixed point or limit cycle with conjugate linear dynamics described by Eq. 4.45 in Section 4.2.1, the Koopman eigenfunctions provide a global coordinate system in the basin of attraction where the dynamics are decoupled. For example, Mauroy et al. [173] notice that the magnitude and phase of any eigenfunction φ with eigenvalue $\lambda = \sigma + \imath\omega$ having nonzero imaginary part provides a pair of action-angle coordinates in the basin that evolve according to

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,|\varphi_{\lambda,1}| = \sigma|\varphi_{\lambda,1}| \qquad \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\angle\varphi_{\lambda,1} = \omega. \tag{4.51}$$

In the case of stable hyperbolic fixed points and limit cycles the eigenfunctions whose eigenvalues have the largest real part describe the asymptotic behavior of the system [173, 171]. When there is a limit cycle, the periodic coordinate gives rise to Koopman eigenfunctions with purely imaginary eigenvalues $\imath k\omega_0$. The points lying on the same level set of $\angle\varphi_{\imath k\omega_0,1}$ have asymptotically the same phase around the limit cycle and are called "isochrons" [171]. Similarly, isochrons can be defined as points having asymptotically the same phase in their plane of approach to the origin. Isochrons are given by the level sets of $\angle\varphi_{\lambda,1}$, where λ is the (assumed to be unique) eigenvalue with largest real part. The level sets of this eigenfunction's magnitude provide complementary collections of points called "isostables" [173] with the same asymptotic approach to the origin up to a difference in phase.

Finally, the work of Mauroy and Mezić [172] shows how Koopman eigenfunctions and their level

sets can be used to study the stability of arbitrary sets in state space. They show that the zero level sets of continuous eigenfunctions of the Koopman generator with negative real-part eigenvalues are forward-invariant and globally asymptotically stable. Consequently this is true of any intersection of zero level sets associated with multiple such eigenfunctions, allowing the geometric study of such level sets to reveal asymptotically stable sets. A Lyapunov function for the globally stable set can be constructed by summing the absolute squares of each of these eigenfunctions. Moreover, if there is a globally stable set, then all continuous eigenfunctions supported on its complement have eigenvalues with negative real part and those continuous eigenfunctions with support overlapping the globally stable set have purely imaginary eigenvalues. These results are interesting because they indicate that the Koopman operator and its eigenfunctions may be useful tools for studying the transient dynamics off of attractors.

Koopman Mode Analysis

A consequence of the product rule for Koopman eigenfunctions allows us to express certain observables as linear combinations of eigenfunctions. Suppose that $\boldsymbol{\psi} : \mathcal{X} \to \mathbb{C}^q$ is a vector-valued observable that can be expressed by composing a continuous function $\boldsymbol{g} : \mathbb{C}^m \to \mathbb{C}^d$ with a finite collection of eigenfunctions $\varphi_1, \ldots, \varphi_m$ of the Koopman generator, that is,

$$\boldsymbol{\psi}(x) = \boldsymbol{g}(\varphi_1(x), \ldots, \varphi_m(x)). \tag{4.52}$$

If $\boldsymbol{g}$ is expressible as a convergent power series on the image of $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_m) : \mathcal{X} \to \mathbb{C}^m$, then we may express $\boldsymbol{\psi}$ as a power series of eigenfunctions

$$\boldsymbol{\varphi} = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^m} \boldsymbol{\xi}_{\boldsymbol{\alpha}} \boldsymbol{\varphi}^{\boldsymbol{\alpha}}, \qquad \boldsymbol{\varphi}^{\boldsymbol{\alpha}} = \varphi_1^{\alpha_1} \cdots \varphi_m^{\alpha_m}, \tag{4.53}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ is a multi-index with sum $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_m$. If each eigenfunction φ_i of the Koopman generator has eigenvalue λ_i, then each function $\boldsymbol{\varphi}^{\boldsymbol{\alpha}} = \varphi_1^{\alpha_1} \cdots \varphi_m^{\alpha_m}$ is also an eigenfunction of the Koopman generator with eigenvalue $\alpha_1\lambda_1 + \cdots + \alpha_m\lambda_m$. Such eigenvalues form a lattice in the complex plane. If the reconstruction function $\boldsymbol{g}$ is merely continuous, then by the Stone-Weierstrass theorem, there is a sequence of polynomials

$$\boldsymbol{g}_n(\boldsymbol{z}) = \sum_{\substack{\boldsymbol{\alpha} \in \mathbb{N}^m : \\ |\boldsymbol{\alpha}| \leq d_n}} \boldsymbol{\xi}_{n,\boldsymbol{\alpha}} \boldsymbol{z}^{\boldsymbol{\alpha}}, \tag{4.54}$$

such that $\boldsymbol{g}_n \to \boldsymbol{g}$ uniformly on every compact subset of $\mathbb{C}^m$. Consequently, the observable $\boldsymbol{\psi}$ may be expressed as a limit

$$\boldsymbol{\psi}(x) = \lim_{n\to\infty} \boldsymbol{\psi}_n(x), \qquad \boldsymbol{\psi}_n(x) = \sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}^m:\\ |\boldsymbol{\alpha}|\le d_n}} \boldsymbol{\xi}_{n,\boldsymbol{\alpha}} \boldsymbol{\varphi}^{\boldsymbol{\alpha}}(x), \tag{4.55}$$

where the convergence is uniform on every compact subset of $\mathcal{X}$ if $\boldsymbol{\varphi}$ is continuous.

The coefficients on expansions of observables in terms of Koopman eigenfunctions evolve in a very simple way under the dynamics. The coefficients are called "Koopman modes", defined below.

Definition 4.2.2 (Koopman mode)**.** *The coefficients $\boldsymbol{\xi}_i$ used to express a (vector-valued) observable $\boldsymbol{\psi} = \sum_i \boldsymbol{\xi}_i \varphi_i$ as a linear combination of Koopman eigenfunctions are called the "Koopman modes" of the observable $\boldsymbol{\psi}$.*

If the observable $\boldsymbol{\psi}$ can be written as a linear combination of eigenfunctions φ_i of the Koopman generator with eigenvalues λ_i and Koopman modes $\boldsymbol{\xi}_i$, then the evolution of $\boldsymbol{\psi}$ is given by

$$U^t \boldsymbol{\psi} = \sum_{i=1}^{\infty} e^{\lambda_i t} \boldsymbol{\xi}_i \varphi_i. \tag{4.56}$$

Hence, we see that the Koopman modes of the observable $U^t\boldsymbol{\psi}$ evolve according to $e^{\lambda_i t}\boldsymbol{\xi}_i$, i.e., with a fixed frequency $\omega_i = \operatorname{Im}(\lambda_i)$ and decay rate $\sigma_i = -\operatorname{Re}(\lambda_i)$ over time.

Near a stable hyperbolic fixed point with linearized dynamics $\delta\dot{\boldsymbol{x}} = \boldsymbol{A}\delta\boldsymbol{x}$, the Koopman modes of the state vector corresponding to the eigenfunctions with non-vanishing gradients at the fixed point are the eigenvectors of $\boldsymbol{A}$ with the same eigenvalues. As discussed in [173], the Koopman modes of the state vector corresponding to the eigenfunctions whose eigenvalues have maximum real part determine the plane of asymptotic approach to the fixed point. The other Koopman modes associated with eigenfunctions having vanishing gradients at the fixed point account for high-order nonlinear effects far away from the fixed point. In the case of limit cycle dynamics, the Koopman modes corresponding to eigenfunctions with purely imaginary eigenvalues are the Fourier modes for the limit cycle dynamics. Here, the complex argument of the eigenfunction at the fundamental frequency provides the appropriate phase on the limit cycle.

When the Koopman semigroup $\{U^t\}_{t\ge 0}$ is strongly continuous and uniformly bounded on a *reflexive* Banach space $\mathcal{F}$, then it is possible to recover the Koopman modal diads $\boldsymbol{\xi}\varphi$ associated with complex eigenvalues of the Koopman generator by Fourier averaging. In particular, the mean

ergodic theorem for semigroups [86] shows that the Fourier average

$$P_{i\omega}\psi = \lim_{T\to\infty} \frac{1}{T} \int_0^T e^{-i\omega t} U^t \psi \, \mathrm{d}\, t, \qquad \omega \in \mathbb{R} \tag{4.57}$$

converges strongly. Moreover, the resulting bounded operator $P_{i\omega} : \mathcal{F} \to \mathcal{F}$ is a projection that commutes with each U^t and decomposes $\mathcal{F} = \mathrm{Range}(P_{i\omega}) \oplus \mathrm{Null}(P_{i\omega})$, where

$$\mathrm{Range}(P_{i\omega}) = \mathrm{Null}(V - i\omega I) \qquad \text{and} \qquad \mathrm{Null}(P_{i\omega}) = \overline{\mathrm{Range}(V - i\omega I)}. \tag{4.58}$$

Hence, if $i\omega$ is an eigenvalue of V, then $P_{i\omega}$ yields a projection onto the corresponding eigenspace. Consequently, if $i\omega$ is an eigenvalue of the Koopman generator with multiplicity 1 and φ is the eigenfunction, then the corresponding Koopman modal diad for the observable $\boldsymbol{\psi}$ is given by

$$\boldsymbol{\xi}\varphi = \lim_{T\to\infty} \frac{1}{T} \int_0^T e^{-i\omega t} \boldsymbol{\psi} \circ F^t \, \mathrm{d}\, t. \tag{4.59}$$

In the special case when $\{U^t\}_{t\in\mathbb{R}}$ is a unitary group on a Hilbert space, Stone's theorem (Theorem 4.1.5) yields a skew-adjoint generator V, which makes $P_{i\omega}$ an orthogonal projection. An idea related to Fourier averaging is the generalized Laplace analysis (GLA) developed in [37], which can be applied to find all of the Koopman modal diads for certain systems. However, GLA suffers from poor numerical stability, and the need for all eigenvalues of the Koopman generator to be known explicitly. In practice, accurate approximations of Koopman modes can be computed from a data set generated by the underlying system using Dynamic Mode Decomposition (DMD) [231] or Extended Dynamic Mode Decomposition (EDMD) [280].

4.3 Approximation techniques based on dictionaries

A variety of techniques are available for making finite-dimensional approximations of Koopman operators and generators based on data collected from the system under investigation. For a review of modern techniques, see S. E. Otto and C. W. Rowley [196]*. In this section, we briefly review Extended Dynamics Mode Decomposition (EDMD) [280] and its generalization to systems with actuation [279]. In particular, we shall focus on the application of these techniques by S. E. Otto in S. Peitz et al. [202]* to approximate the Koopman generators of input-affine systems.

In general, the Koopman operator is not compact, and so it is impossible to make finite-dimensional approximations that converge in the operator norm. This makes approximating the

spectrum of Koopman operators difficult since (almost) any finite-dimensional approximation will be diagonalizable, yet many or all of the eigenvalues and eigenfunctions may be spurious since the Koopman operator may not have any eigenfunctions besides the constant function. Before making a finite-dimensional approximation, it is necessary to compactify the Koopman operator or generator in a way that preserves meaningful spectral information. In the case when $\{U^t\}_{t\in\mathbb{R}}$ is a unitary group, the compactification technique developed by S. Das, D. Giannakis, and J. Slawinska [75] may be applied to approximate the Koopman generator and its entire Borel functional calculus including the exponential map used to construct each U^t. However, this result relies heavily on the skew-adjoint property of the generator, and it is unclear whether the method can be generalized to cases when the Koopman operators do not form a unitary group. While it may not be possible to approximate the spectrum and functional calculus of Koopman operators and generators using existing methods, the relatively simple techniques we describe in this section converge point-wise and yield accurate predictions of dynamics over finite time horizons. For a discussion of convergence results for EDMD, see M. Korda and I. Mezić [139].

4.3.1 Extended Dynamic Mode Decomposition (EDMD) [280]

In EDMD [280], we seek to approximate the Koopman operator U^t in the span of a pre-selected dictionary of functions $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_N)$, $\psi_i \in \mathcal{F}$, referred to as "observables". In most cases, we do not have a priori knowledge of any non-trivial Koopman-invariant subspaces, and so the subspace $\mathcal{V} \subset \mathcal{F}$ spanned our chosen $\psi_1, \ldots, \psi_N$ cannot reasonably be expected to be Koopman-invariant. We must therefore approximate each $U^t\psi_j$ in the subspace $\mathcal{V}$ by defining a suitable projection operator $P_{\mathcal{V}} : \mathcal{F} \to \mathcal{F}$ with range contained in $\mathcal{V}$. A matrix representation $\boldsymbol{U} \in \mathbb{C}^{N\times N}$ of the projected Koopman operator $P_{\mathcal{V}}U^t$ on the subspace $\mathcal{V}$ may then be constructed, i.e., a matrix $\boldsymbol{U}$ satisfying

$$\boxed{P_{\mathcal{V}}U^t(\boldsymbol{\psi}^T\boldsymbol{a}) = \boldsymbol{\psi}^T\boldsymbol{U}\boldsymbol{a}, \qquad \boldsymbol{\psi}^T\boldsymbol{a} := a_1\psi_1 + \cdots + a_N\psi_N \in \mathcal{V},} \tag{4.60}$$

for every $\boldsymbol{a} \in \mathbb{C}^N$.

The question now is how to construct a suitable projection operator? Suppose that $\mathcal{F}$ is a Banach space with strictly convex norm and $\mathcal{V}$ is a finite-dimensional subspace. It can be shown that there is a projection operator $P_{\mathcal{V}} : \mathcal{F} \to \mathcal{F}$ with the property that $P_{\mathcal{V}}f$ is the unique element in $\mathcal{V}$ that is closest to f with respect to the norm. If the norm of $\mathcal{F}$ is uniformly convex, i.e., $\mathcal{F}$ is reflexive, then this result is also true for any infinite-dimensional closed subspace $\mathcal{V} \subset \mathcal{F}$. In a Hilbert space $P_{\mathcal{V}}$ is simply the orthogonal projection onto $\mathcal{V}$. However, over Banach spaces of continuous functions

on manifolds, the norm is not strictly convex, and so minimizers with respect to the norm may not be unique. Moreover, it is usually impractical to compute the norms of arbitrary functions defined over high-dimensional state spaces such as those encountered in discretized fluid flows.

When the norm on $\mathcal{F}$ is defined by an integral with respect to a probability measure, then the Monte-Carlo approximation method can be employed. In particular, we suppose that $\mathcal{F} = L^2((\mathcal{X}, \mu); \mathbb{C})$, and we draw M independent identically distributed samples $x_1, \ldots, x_M$ from the distribution μ. By Kolmogorov's strong law of large numbers [137], we have

$$\langle f, g \rangle_M := \frac{1}{M} \sum_{i=1}^{M} \overline{f(x_i)} g(x_i) \to \int_{\mathcal{X}} \overline{f} g \, \mathrm{d}\, \mu =: \langle f, g \rangle_{L^2} \qquad \text{as} \qquad M \to \infty \tag{4.61}$$

almost surely for every $f, g \in L^2((\mathcal{X}, \mu); \mathbb{C})$. Approximating the inner product $\langle f, g \rangle_{L^2}$ by the sample-based semi-inner product $\langle f, g \rangle_M$ yields a projection operator

$$P_{\mathcal{V},M} f = \boldsymbol{\psi}^T \boldsymbol{G}_M^+ \begin{bmatrix} \langle \psi_1, f \rangle_M \\ \vdots \\ \langle \psi_N, f \rangle_M \end{bmatrix}, \qquad [\boldsymbol{G}_M]_{i,j} = \langle \psi_i, \psi_j \rangle_M \,, \tag{4.62}$$

where $\boldsymbol{G}_M^+$ denotes the Moore-Penrose pseudoinverse of the Gram matrix $\boldsymbol{G}_M$. The element $P_{\mathcal{V},M} f$ minimizes $\|f - \psi\|_M^2 = \langle f - \psi, f - \psi \rangle_M$ over $\psi \in \mathcal{V} = \operatorname{span}\{\psi_1, \ldots, \psi_N\}$, and is the unique minimizer when M is large enough so that $\boldsymbol{G}_M$ is positive-definite. As a consequence of the strong law of large numbers, this projection converges point-wise almost surely to the orthogonal projection operator $P_{\mathcal{V}}$ onto $\mathcal{V}$ in $L^2((\mathcal{X}, \mu); \mathbb{C})$. The EDMD method uses this projection operator to construct the matrix approximation

$$\boxed{\boldsymbol{U} = \boldsymbol{G}_M^+ \boldsymbol{A}_M, \qquad [\boldsymbol{A}_M]_{i,j} = \left\langle \psi_i, U^t \psi_j \right\rangle_M = \frac{1}{M} \sum_{k=1}^{M} \overline{\psi_i(x_k)} \psi_j(F^t(x_k)),} \tag{4.63}$$

of the Koopman operator according to Eq. 4.60. The key feature of this method is that $\boldsymbol{U}$ may be computed from data consisting of snapshot pairs $\boldsymbol{\psi}(x_i), \boldsymbol{\psi}(F^t(x_i))$ obtained by evaluating the same observables $\boldsymbol{\psi}$ at states x_i and at $F^t(x_i)$ after evolving the states under the flow map. These data are easy to obtain from numerical simulations or experiments, contributing to the widespread use of this technique.

Extended Dynamic Mode Decomposition (EDMD) [280] has a different interpretation when applied to Banach spaces of continuous functions. A more practical alternative to norm minimization

in a general Banach space $\mathcal{F}$ is to define the projection operator with respect to a collection of test functionals $\{\theta_1, \ldots, \theta_M\}$ in the dual Banach space $\mathcal{F}^*$. The dual spaces to Banach spaces of continuous functions such as $C(\mathcal{X})$ and $C_0(\mathcal{X})$ contain the Dirac measures δ_x defined by

$$\delta_x(A) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases} \tag{4.64}$$

for Borel measurable subsets $A \subset \mathcal{X}$. In EDMD, the functionals are chosen to be Dirac measures $\theta_i = \delta_{x_i}$ centered at a collection of sampled data points $x_1, \ldots, x_M \in \mathcal{X}$. One may then define the projection $P_{\mathcal{V},M}$ to return a minimizer of the square approximation error

$$\underset{\psi \in \mathcal{V}}{\text{minimize}} \quad \frac{1}{M} \sum_{i=1}^{M} |\langle \theta_i,\ f - \psi \rangle|^2 \tag{4.65}$$

of functions $f \in \mathcal{F}$ in the subspace $\mathcal{V}$ as viewed through the lens of the chosen functionals. The resulting projection operator is given by

$$P_{\mathcal{V}} f = \boldsymbol{\psi}^T \boldsymbol{T}^+ \begin{bmatrix} \langle \theta_1, f \rangle \\ \vdots \\ \langle \theta_M, f \rangle \end{bmatrix}, \qquad [\boldsymbol{T}]_{i,j} = \langle \theta_i, \psi_j \rangle, \quad 1 \le i \le M,\ 1 \le j \le N, \tag{4.66}$$

which yields the EDMD matrix approximation

$$\boxed{\boldsymbol{U} = \boldsymbol{T}^+ \boldsymbol{S}, \qquad [\boldsymbol{S}_M]_{i,j} = \langle \theta_i, U^t \psi_j \rangle, \quad 1 \le i \le M,\ 1 \le j \le N.} \tag{4.67}$$

It is interesting to note that the projection operator and matrix approximation given by Eq. 4.66 and Eq. 4.67 are identical to those given by Eq. 4.62 and Eq. 4.63, with a similar sample-based semi-inner product

$$\langle f, g \rangle_M := \frac{1}{M} \sum_{i=1}^{M} \overline{\langle \theta_i, f \rangle} \langle \theta_i, g \rangle. \tag{4.68}$$

In this case, we have $\boldsymbol{G}_M = \boldsymbol{T}^* \boldsymbol{T}$, where $\boldsymbol{T}^*$ denotes the complex conjugate transpose of the matrix $\boldsymbol{T}$. When $\theta_i = \delta_{x_i}$ and x_i are drawn independently at random according to a distribution μ, then the strong law of large numbers [137] implies that

$$\langle f, g \rangle_M = \frac{1}{M} \sum_{i=1}^{M} \overline{f(x_i)} g(x_i) \to \int_{\mathcal{X}} \overline{f} g \, \mathrm{d}\, \mu \qquad \text{as} \qquad M \to \infty \tag{4.69}$$

almost surely under the additional assumption that $\overline{f}g$ is absolutely integrable with respect to μ.

The range of the data-driven projection operator $P_{\mathcal{V}}$ is a subspace $\mathcal{U} \subset \mathcal{V}$, which is referred to as the "learning subspace". This subspace is important because it is the subspace in which EDMD is capable of accurately capturing the dynamics of Koopman eigenfunctions. In particular, if $\varphi \in \mathcal{U}$ is an eigenfunction of the Koopman operator U^t with eigenvalue λ, then the corresponding vector $\boldsymbol{v} \in \mathbb{C}^N$ such that $\varphi = \boldsymbol{\psi}^T \boldsymbol{v}$ is an eigenvector of the matrix approximation $\boldsymbol{U}$ with the same eigenvalue! On the other hand, all bets are off if $\varphi \in \mathcal{V}$ but $\varphi \notin \mathcal{U}$. The learning subspace is equal to $\mathcal{V}$ when the the sampling strategy has $M \geq N$ and yields a positive-definite Gram matrix $\boldsymbol{G}_M$, i.e., the matrix $\boldsymbol{T}$ has full column-rank. Otherwise, the learning subspace is a proper subspace of $\mathcal{V}$ given by

$$\mathcal{U} = \left\{ \boldsymbol{\psi}^T \boldsymbol{a} \ : \ \boldsymbol{a} \in \text{Range}(\boldsymbol{G}_M) = \text{Range}(\boldsymbol{T}^*) \right\}. \tag{4.70}$$

In practice, the Moore-Penrose pseudoinverses used to define the EDMD matrix $\boldsymbol{U}$ via Eq. 4.63 or Eq. 4.67 cannot be stably computed. Instead, we approximate the pseudoinverse by truncated Hermitian eigenvalue decomposition of $\boldsymbol{G}_M$ or equivalently by truncated singular value decomposition of $\boldsymbol{T}$. Truncating these decompositions yields smaller learning subspaces where the vector $\boldsymbol{a}$ in Eq. 4.70 lives in the range of the truncated matrix $\boldsymbol{G}_M$, i.e, in the span of the truncated right singular vectors of $\boldsymbol{T}$. Letting $\boldsymbol{Q} \in \mathbb{C}^{m \times r}$ be the matrix of truncated right singular vectors of $\boldsymbol{T}$, we find the the matrix approximation of the Koopman operator restricted to the r-dimensional learning subspace, $P_{\mathcal{V}} U P_{\mathcal{V}}$, is given by

$$(P_{\mathcal{V}} U P_{\mathcal{V}}) \boldsymbol{\psi}^T \boldsymbol{Q} \boldsymbol{a} = \boldsymbol{\psi}^T \hat{\boldsymbol{U}} \boldsymbol{a}, \qquad \hat{\boldsymbol{U}} = \boldsymbol{Q}^* \boldsymbol{U} \boldsymbol{Q}, \tag{4.71}$$

where $\boldsymbol{U}$ is the matrix approximation of $P_{\mathcal{V}} U$ given by Eq. 4.63 or Eq. 4.67. Here, $\hat{\boldsymbol{U}}$ is an $r \times r$ matrix, compared to $\boldsymbol{U}$, which is an $N \times N$ matrix. Working with $\hat{\boldsymbol{U}}$ rather than $\boldsymbol{U}$ enables a generalization of EDMD to infinite-dimensional feature maps $\boldsymbol{\psi}$ taking values in a reproducing kernel Hilbert space. This technique, called Kernel Dynamic Mode Decomposition (KDMD) is introduced by M. O. Williams et al. [281]. As we alluded to earlier in Section 3.1.2, when the same learning subspace is used to construct $\boldsymbol{U}$ and $\hat{\boldsymbol{U}}$, then $\boldsymbol{U}$ and $\hat{\boldsymbol{U}}$ have the same nonzero eigenvalues and the eigenvectors of $\boldsymbol{U}$ can be reconstructed from the eigenvectors of $\hat{\boldsymbol{U}}$. However, we note that working with an infinite-dimensional set of features is not necessarily helpful because the dimension of the learning subspace never exceeds the number of samples M.

Finally, EDMD can be used to construct an approximate Koopman mode decomposition for

a vector of observables $\boldsymbol{f} : \mathcal{X} \to \mathbb{C}^d$. If $\boldsymbol{U} = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^{-1}$, $\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_N \end{bmatrix}$ is an eigendecomposition for the matrix approximation $\boldsymbol{U}$ of the Koopman operator, then $\varphi_i = \boldsymbol{\psi}^T \boldsymbol{w}_i$ is an eigenvector of $P_{\mathcal{V}}U$ with the corresponding eigenvalue λ_i of $\boldsymbol{U}$. Therefore, $\boldsymbol{\varphi} = \boldsymbol{W}^T\boldsymbol{\psi}$ is the vector of approximate eigenfunctions computed using EDMD. Interpreting $\langle \theta_i, \boldsymbol{f} \rangle = (\langle \theta_i, f_1 \rangle, \ldots, \langle \theta_i, f_d \rangle)$ element-wise, a little bit of algebra yields a decomposition

$$\left(P_{\mathcal{V}}U^t\right)^k P_{\mathcal{V}}\boldsymbol{f} = \underbrace{\begin{bmatrix} \langle \theta_1, \boldsymbol{f} \rangle & \cdots & \langle \theta_M, \boldsymbol{f} \rangle \end{bmatrix} \boldsymbol{T}^{+T}\boldsymbol{W}^{-T}}_{\begin{bmatrix} \boldsymbol{\xi}_1 & \cdots & \boldsymbol{\xi}_N \end{bmatrix}} \boldsymbol{\Lambda}^k \boldsymbol{\varphi} = \sum_{i=1}^{N} \boldsymbol{\xi}_i \lambda_i^k \varphi_i, \qquad k = 0, 1, 2, \ldots \tag{4.72}$$

of $P_{\mathcal{V}}\boldsymbol{f}$ in terms of Koopman modes $\boldsymbol{\xi}_i$ for the approximate Koopman operator $P_{\mathcal{V}}U^t$.

4.3.2 EDMD-like method for bilinear Koopman generators

The EDMD approximation technique developed above for the Koopman operator U^t can also be applied directly to construct an analogous matrix approximation $\boldsymbol{V}$ of the Koopman generator V. This approach is especially advantageous when the continuous-time dynamics have affine-dependence on the input, or on known functions of the input. Consequently, in [202]*, S. E. Otto recognized that EDMD-like techniques can be applied to construct the terms in a corresponding input-affine approximation of the Koopman generator. The results obtained by S. Peitz using this technique to construct surrogate models for model-predictive control (MPC) can be found in [202]*. Related work by M. Korda and I Mezić [138] uses an EDMD-like technique to learn a linear, rather than an input-affine model for the dynamics of observables. At the end of this section, we show that such "lifted LTI" models are special cases of bilinear models, but lack the ability to accurately approximate certain systems. On the other hand, bilinear models are capable of approximating very large classes of nonlinear dynamical systems [256, 158].

So far, we have considered EDMD-based approximation techniques for Koopman operators associated with autonomous systems. As we have seen, when a constant actuation is applied over a time interval of length t, then the Koopman operator U_u^t depends on this input in a possibly complicated way. In [279], M. O. Williams et al. introduces an EDMD-like technique to construct a matrix approximation $\boldsymbol{U}_u$ of U_u^t, where the dependence on u is captured using a basis expansion

$$\boldsymbol{U}_u = \sum_{i=1}^{L} g_i(u)\boldsymbol{U}_i \in \mathbb{C}^{N \times N} \tag{4.73}$$

with user-defined basis functions $g_1, \ldots, g_L$. The terms $\boldsymbol{U}_i$ in the expansion are determined by solving a least-squares problem

$$\underset{\boldsymbol{U}_1,\ldots,\boldsymbol{U}_L}{\text{minimize}} \ \frac{1}{M}\sum_{i=1}^{M}\left\|(U_{u_i}^t\boldsymbol{\psi}^T)(x_i) - \sum_{k=1}^{L} g_k(u_i)\boldsymbol{\psi}(x_i)^T\boldsymbol{U}_k\right\|^2. \tag{4.74}$$

The difficulty with this approach is that the finite-time Koopman operator may have exceedingly complicated dependence on the input via the exponential map, even when the original governing equations depend on the input in a simple way. This requires a large number of terms $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_L$ in the basis expansion. Consequently a large amount of data $\{(\boldsymbol{\psi}(x_i), u_i, \boldsymbol{\psi}(F_{u_i}^t(x_i))\}_{i=1}^M$ is needed to accurately approximate the input-dependence.

On the other hand, we know that the Koopman generator for a control-affine system

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,x = f_0(x) + \sum_{i=1}^{m}[\boldsymbol{u}]_i f_i(x) \tag{4.75}$$

is also control-affine and so we may construct a control-affine matrix approximation of the Koopman generator using essentially the same method as in [279]. The result is a bilinear model for the dynamics of a lifted state vector $\boldsymbol{z} = \boldsymbol{\psi}(x)$ given by

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\boldsymbol{z} = \boldsymbol{V}_0^T\boldsymbol{z} + \sum_{i=1}^{m}[\boldsymbol{u}]_i\boldsymbol{V}_i^T\boldsymbol{z}, \tag{4.76}$$

with observations reconstructed in the span of the components of $\boldsymbol{\psi}$ according to $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{z}$. Here, $\boldsymbol{V}_i$ are matrix approximations of the Koopman generators corresponding to each component vector field f_i.

To construct the model, let us suppose that $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_N) : \mathcal{X} \to \mathbb{C}^N$ is a vector of observables in the domain of each component Koopman generator $\cap_{i=1}^m \mathrm{Dom}(V_i)$ associated with the vector fields f_i and that $P_{\mathcal{V}} : \mathcal{F} \to \mathcal{F}$ is a projection onto $\mathcal{V} = \mathrm{span}\{\psi_1, \ldots, \psi_N\}$. Then the matrix approximation of the Koopman generator V_u at actuation level u is then given by an affine combination

$$P_{\mathcal{V}}V_{\boldsymbol{u}}\boldsymbol{\psi}^T\boldsymbol{a} = \left(P_{\mathcal{V}}V_0 + \sum_{i=1}^{m}[\boldsymbol{u}]_i P_{\mathcal{V}}V_i\right)\boldsymbol{\psi}^T\boldsymbol{a} = \boldsymbol{\psi}^T\left(\boldsymbol{V}_0 + \sum_{i=1}^{m}[\boldsymbol{u}]_i\boldsymbol{V}_i\right)\boldsymbol{a} \qquad \forall \boldsymbol{a} \in \mathbb{C}^N \tag{4.77}$$

of matrix approximations $\boldsymbol{V}_i$ for the Koopman generators V_i associated with each vector field f_i. This means that we can construct the EDMD matrix approximation of V_u from the corresponding matrix approximations for each vector field. Recall that the EDMD projection operator $P_{\mathcal{V}}$ is defined by the data set $\{x_1, \ldots, x_M\}$. Thus, this approach requires that we use the same sample points to

approximate each component Koopman generator.

Alternatively, we can construct the matrix approximations $\boldsymbol{V}_0, \ldots, \boldsymbol{V}_m$ directly from a single data set $\left\{ \left(\boldsymbol{\psi}(x_i), \boldsymbol{u}_i, \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\psi}(F_{\boldsymbol{u}_i}^t(x_i))\big|_{t=0} \right) \right\}_{i=1}^M$ by solving a least squares problem

$$\underset{\boldsymbol{V}_0,\ldots,\boldsymbol{V}_m}{\text{minimize}} \ \frac{1}{M} \sum_{i=1}^{M} \left\| (V_{\boldsymbol{u}_i} \boldsymbol{\psi}^T)(x_i) - \boldsymbol{\psi}(x_i)^T \boldsymbol{V}_0 - \sum_{k=1}^{m} [\boldsymbol{u}_i]_k \boldsymbol{\psi}(x_i)^T \boldsymbol{V}_k \right\|^2. \tag{4.78}$$

analogous to Eq. 4.74. Unfortunately, this does not result in a projection operator $P_{\mathcal{V}}$ defined independently from the input, but rather a projection operator $\tilde{P}_{\mathcal{V}^{m+1}}$ on $\mathcal{F}^{m+1}$ into $\mathcal{V}^{m+1}$. Here, $\mathcal{F}^{m+1}$ and $\mathcal{V}^{m+1}$ denote the spaces of input-affine functions of the form $g(\boldsymbol{u}, x) = g_0(x) + \sum_{i=1}^m [\boldsymbol{u}]_i g_i(x)$ with each $g_i \in \mathcal{F}$ and $g_i \in \mathcal{V}$, respectively. The space $\mathcal{V}^{m+1}$ is spanned by the elements of the vector $\tilde{\boldsymbol{\psi}}$, where

$$\tilde{\boldsymbol{\psi}}(\boldsymbol{u}, x) = (1, \boldsymbol{u}) \otimes \boldsymbol{\psi}(x) = (\boldsymbol{\psi}, [\boldsymbol{u}]_1 \boldsymbol{\psi}, \ldots, [\boldsymbol{u}]_m \boldsymbol{\psi}). \tag{4.79}$$

In particular, the projection operator

$$\tilde{P}_{\mathcal{V}^{m+1}} f = \tilde{\boldsymbol{\psi}}^T \tilde{\boldsymbol{T}}^+ \begin{bmatrix} f(\boldsymbol{u}_1, x_1) \\ \vdots \\ f(\boldsymbol{u}_M, x_M) \end{bmatrix}, \qquad \tilde{\boldsymbol{T}} = \begin{bmatrix} \tilde{\boldsymbol{\psi}}(\boldsymbol{u}_1, x_1)^T \\ \vdots \\ \tilde{\boldsymbol{\psi}}(\boldsymbol{u}_M, x_M)^T \end{bmatrix} = \begin{bmatrix} (1, \boldsymbol{u}_1)^T \otimes \boldsymbol{\psi}(x_1)^T \\ \vdots \\ (1, \boldsymbol{u}_M)^T \otimes \boldsymbol{\psi}(x_M)^T \end{bmatrix} \tag{4.80}$$

yields the unique minimizer of

$$\underset{g \in \mathcal{V}^{m+1}}{\text{minimize}} \ \frac{1}{M} \sum_{i=1}^{M} |f(\boldsymbol{u}_i, x_i) - g(\boldsymbol{u}_i, x_i)|^2 \tag{4.81}$$

for every $f \in \mathcal{F}^{m+1}$. Consequently, the matrices obtained by solving Eq. 4.78 yeild approximations of the operator $\tilde{V} : \bigcap_{i=0}^m \mathrm{Dom}(V_i) \to \mathcal{F}^{m+1}$ defined by

$$(\tilde{V}\psi)(\boldsymbol{u}, x) = (V_{\boldsymbol{u}}\psi)(x) = \left(V_0 \psi + \sum_{i=1}^{m} [\boldsymbol{u}]_i V_i \psi \right)(x) \tag{4.82}$$

in the sense that

$$\left(\tilde{P}_{\mathcal{V}^{m+1}} \tilde{V} \right) \boldsymbol{\psi}^T \boldsymbol{a} = \tilde{\boldsymbol{\psi}}^T \begin{bmatrix} \boldsymbol{V}_0 \\ \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_m \end{bmatrix} \boldsymbol{a}. \tag{4.83}$$

The solution is given by

$$\begin{bmatrix} \boldsymbol{V}_0 \\ \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_m \end{bmatrix} = \tilde{\boldsymbol{T}}^+ \tilde{\boldsymbol{S}}, \qquad \tilde{\boldsymbol{S}} = \begin{bmatrix} (V_{\boldsymbol{u}_1} \boldsymbol{\psi}^T)(x_1) \\ \vdots \\ (V_{\boldsymbol{u}_M} \boldsymbol{\psi}^T)(x_M) \end{bmatrix}. \tag{4.84}$$

In [202]*, we show that this approach can be used to accurately approximate the Koopman generators and predict the dynamics of several systems. Moreover, we describe how the low-dimensional surrogate models in the form of Eq. 4.76 obtained using this approach can be used for model-predictive control.

A special case of the lifted bilinear model Eq. 4.76 that has received a lot of attention in the literature [211, 138, 11, 186, 128] is the lifted linear-time-invariant (LTI) model

$$\begin{aligned} \dot{\boldsymbol{z}} &= \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{u}, \qquad \boldsymbol{z} = \boldsymbol{\psi}(x) \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{z}. \end{aligned} \tag{4.85}$$

To see that this is a special case of the bilinear model Eq. 4.76, consider the observables $(1, \psi_1, \ldots, \psi_N)$ that evolve according to

$$\begin{bmatrix} 0 \\ \dot{\boldsymbol{z}} \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{z} \end{bmatrix} + \sum_{k=1}^{\dim u} u_k \begin{bmatrix} 0 & \boldsymbol{0} \\ \boldsymbol{b}_k & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{z} \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \cdots & \boldsymbol{b}_{\dim u} \end{bmatrix}. \tag{4.86}$$

Lifted LTI models are appealing because standard machinery from linear systems and control theory can be applied to them directly. While lifted LTI models are a strictly broader class than LTI systems on the original state space, the dynamics of observed quantities $\boldsymbol{y}$ must still obey the principle of linear superposition. This is a rather serious limitation as the following example demonstrates.

Example 4.3.1 (a system that doesn't admit an accurate lifted LTI model)**.** *There are some very simple systems, for instance*

$$\begin{aligned} \dot{x} &= ux, \qquad x(0) = 1 \\ y &= x, \end{aligned} \tag{4.87}$$

that violate the superposition principle so badly that there cannot exist an accurate lifted LTI model in the form of Eq. 4.85, regardless of how large the dimension, $\dim \boldsymbol{z}$*, of such an approximation is taken to be. To see why this system does not admit a lifted LTI approximation, consider the three*

trajectories

$$u_a \equiv -1 \quad \Rightarrow \quad x_a(t) = e^{-t} \tag{4.88}$$

$$u_b \equiv -3 \quad \Rightarrow \quad x_b(t) = e^{-3t} \tag{4.89}$$

$$u_c \equiv 1 \quad \Rightarrow \quad x_c(t) = e^{t}. \tag{4.90}$$

and observe that for any dictionary of observables $\boldsymbol{\psi}$*, we have*

$$\boldsymbol{\psi}(x_c(0)) = 2\boldsymbol{\psi}(x_a(0)) - \boldsymbol{\psi}(x_b(0)) = \boldsymbol{\psi}(1) \tag{4.91}$$

$$u_c = 2u_a - u_b. \tag{4.92}$$

Therefore, if a lifted LTI model in the form of Eq. 4.85 agrees with the first two trajectories $x_a(t)$ *and* $x_b(t)$*, then it must predict the third trajectory to be*

$$x_c^{(LTI)}(t) = 2x_a(t) - x_b(t) = 2e^{-t} - e^{-3t}. \tag{4.93}$$

This is a very poor prediction because it decays to zero exponentially fast, whereas, the real trajectory $x_c(t)$ *blows up exponentially fast.*

4.4 Linearly recurrent autoencoder networks (LRAN)

The main difficulty with approximation techniques for the Koopman operator and generator based on dictionaries is that their performance depends heavily on the choice of the dictionary $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_N) : \mathcal{X} \to \mathbb{C}^N$. If one has an unlimited amount of data, then the performance of dictionary-based methods increases as the number of basis functions N increases. However, in practice, the amount of data M one may collect from the system is limited. In this case, the performance of EDMD may not improve, and can even become worse, as the number of dictionary functions increases beyond the dimension of the learning subspace, which is at most M. When $N \geq M$ and $\boldsymbol{\psi}(x_1), \ldots, \boldsymbol{\psi}(x_M)$ are linearly independent in $\mathbb{C}^N$, then the EDMD matrix approximation $\boldsymbol{U}$ of the Koopman operator obtained using Eq. 4.63 or Eq. 4.67 fits the data perfectly in the sense that

$$(U^t \boldsymbol{\psi}^T)(x_i) = \boldsymbol{\psi}(x_i)^T \boldsymbol{U} \qquad \text{for every} \qquad i = 1, \ldots, M. \tag{4.94}$$

However, S. E. Otto and C. W. Rowley [194]** show that the predictive performance on new data can be extremely poor. This problem is called over-fitting, and it essentially means that for EDMD to perform well using a fixed amount of data, we must choose a dictionary of functions with $N < M$ that span a subspace that is (nearly) Koopman-invariant. Yet, in most cases, a Koopman-invariant subspace is precisely what we are trying to find, and so we cannot expect to know one ahead of time.

A solution to this dilemma proposed in [194]** is to allow a very small dictionary $N \ll M$ to be optimized with respect to two criteria: approximate invariance under the Koopman operator, and informativeness in the sense that these functions can be used to closely reconstruct the state of the system or other relevant observables. In particular, we parametrize the function $x \mapsto \boldsymbol{\psi}(x) = \boldsymbol{\psi}_e(x;\theta)$ using a neural network forming the encoder in the Linearly-Recurrent Autoencoder Network (LRAN) architecture shown in Figure 4.4.1. Together with the weights defining the encoder, we simultaneously train the parameters defining a decoder $\boldsymbol{\psi}_d$ and a matrix $\boldsymbol{U}$ defining the linear dynamics of the observables. By penalizing the difference between the predicted linear evolution and the actual evolution of the observables $\boldsymbol{\psi}_e$ over time, we favor observables that span a subspace that is as close to Koopman-invariant as possible. However, to avoid trivialities like constant observables appearing in $\boldsymbol{\psi}_e$, we also penalize the reconstruction error of a collection of relevant observables $\boldsymbol{g} : \mathcal{X} \to \mathbb{C}^d$ from the latent space using the decoder $\boldsymbol{\psi}_d$. In [194]**, the state of the system is taken to be a real vector and the relevant observables $\boldsymbol{g}$ are taken to be the coordinate functions on the state space. Penalizing the reconstruction error favors observables $\boldsymbol{\psi}_e$ that preserve information about the state, as in a standard autoencoder (see Section 3.2).

In [194]**, we also develop a balanced truncation technique for reducing the dimension of EDMD and KDMD-based models. Using EDMD or KDMD, we arrive at a large, but finite-dimensional model of the form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\boldsymbol{\varphi}(x(t)) &= \boldsymbol{\Lambda}\boldsymbol{\varphi}(x(t)) \\ \boldsymbol{g}(x(t)) &= \boldsymbol{\Xi}\boldsymbol{\varphi}(x(t)), \end{aligned} \tag{4.95}$$

where $\boldsymbol{\varphi}$ is a vector of eigenfunctions of the approximate Koopman generator $P_{\mathcal{V}}V$ and $\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{\xi}_1 & \cdots & \boldsymbol{\xi}_N \end{bmatrix}$ is the matrix of approximate Koopman modes for a collection of relevant observables $\boldsymbol{g}$. We view the sampled trajectories as impulse-responses of this model using an additional input term of the form $\boldsymbol{Bu}$ where $\boldsymbol{B}$ is chosen to reproduce the second moments of the distribution of initial conditions

$$\mathbb{E}[\boldsymbol{\varphi}(x(0))\boldsymbol{\varphi}(x(0))^*] = \boldsymbol{B}\boldsymbol{B}^*. \tag{4.96}$$

We then use balanced truncation [182] or BPOD [225] to find a low-dimensional linear system

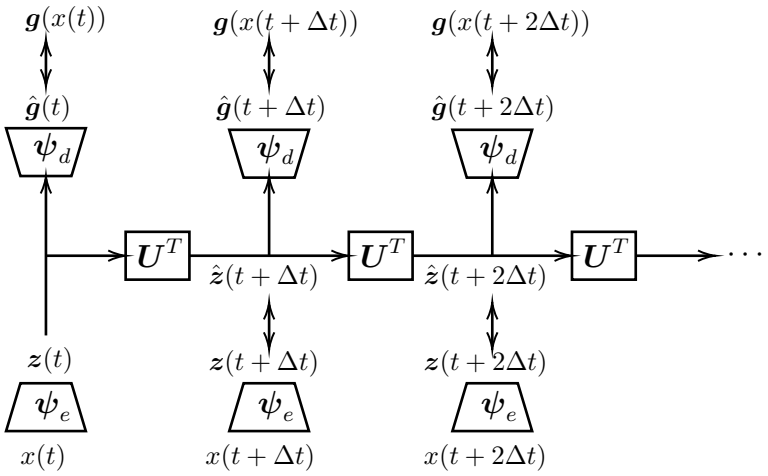
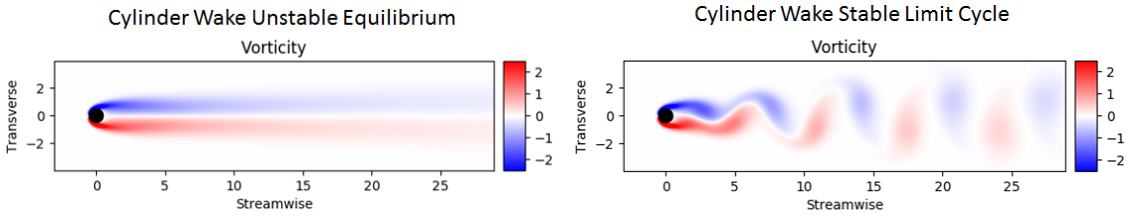


Figure 4.4.1: The architecture of a Linearly-Recurrent Autoencoder Network (LRAN) [194]** consists of encoder and decoder neural networks, denoted $\boldsymbol{\psi}_e$ and $\boldsymbol{\psi}_d$, together with linear dynamics in the latent space described by a matrix $\boldsymbol{U}$. The parameters of $\boldsymbol{\psi}_e$, $\boldsymbol{\psi}_d$, and $\boldsymbol{U}$ are optimized simultaneously during training to minimize a loss function measuring the prediction error of observables $\boldsymbol{g}$ and the prediction error in the latent space over a data set containing sequential snapshots of a system's state x.

that closely approximates the dynamics of our original (E/K)DMD-based model. In reducing the dimension, we introduce additional errors in the evolution and reconstruction of the observables $\boldsymbol{g}$. We recognize that with fewer observables in the reduced-order model, it may not be possible to accurately construct the functions $\boldsymbol{g}$ as linear combinations. Therefore, we allow the reconstruction function for the reduced-order model to be nonlinear, and we use a partially linear kernel regression technique to find it. The resulting model resembles the LRAN architecture in Figure 4.4.1, where $\boldsymbol{\psi}_e$ are the reduced-order model observables and ψ_d is the nonlinear reconstruction function. However, in this case, the parameters defining the encoder and decoder are found sequentially rather than simultaneously as in the training technique employed by the LRAN.

4.4.1 Summary of results

In [194]**, we demonstrate and compare these techniques on three examples: a Duffing equation with two stable fixed points, a cylinder wake flow transitioning from a neighborhood of an unstable steady state to limit cycle dynamics, and a chaotic Kuramoto-Sivashinsky equation. In each case, the LRAN out-performs methods based on KDMD. However, the LRAN takes considerably more time to train, and is somewhat sensitive to the choices of hyper-parameters such as number of nodes and layers in each neural network, as well as the optimization algorithm and learning rate. We summarize some of the results below, and we refer to [194]** for additional results and details about



(a) (b)

Figure 4.4.2: Example cylinder wake flow snapshots at the unstable equilibrium and on the stable limit cycle

these examples.

In our cylinder wake flow example, we consider an incompressible two-dimensional flow around an infinite circular cylinder. The behavior of this system is determined by the non-dimensional Reynolds number $Re = \frac{U_\infty D}{\nu}$, where U_∞ is the free-stream velocity of the fluid at the inflow boundary, D is the diameter of the cylinder, and ν is the kinematic viscosity of the fluid. We consider the $Re = 60$ flow with an initial condition shown in Figure 4.4.2a near an unstable equilibrium state. From this equilibrium, the state evolves along a slow manifold until it reaches a stable limit cycle where vortices are shed in an alternating fashion from the top an bottom sides of the cylinder as in Figure 4.4.2b. We trained an LRAN model with 5 latent state dimensions on 1000 snapshots along a trajectory from the unstable equilibrium to the stable limit cycle, and tested the performance on a held-out set of 500 snapshots. The accuracy of the LRAN-based models with linear and nonlinear decoders, as well as models obtained using KDMD and balanced truncation are compared in Figure 4.4.3. We see that nonlinear reconstruction improves performance in both cases, with the LRAN having significantly lower prediction error than models based on KDMD.

We also compared out modeling approaches on the Kuramoto-Sivashinsky equation, a spatio-temporal PDE, in a periodic spatial domain of length $L = 8\pi$. With a domain of this size, the system exhibits chaotic dynamics near a low-dimensional underlying manifold [124, 135, 114]. Models with 16 latent states were trained using 10000 snapshots and produced the trajectory predictions shown in Figure 4.4.4 on an unseen testing data set of the same size. The prediction errors on the testing data in Figure 4.4.5 indicate that the LRAN and truncated KDMD models are capable of making accurate short-horizon predictions. We observe that the LRAN is more accurate the the KDMD-based model and is about twice as accurate for time horizons ranging in length from 2.5 to 7.5.

It is also possible to use the LRAN architecture to approximate Koopman eigenfunctions associated with known Koopman eigenvalues by constraining the matrix $\boldsymbol{U}$ governing how the latent state

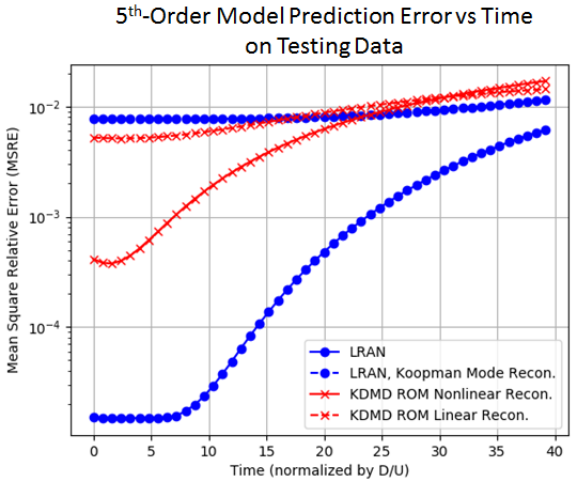
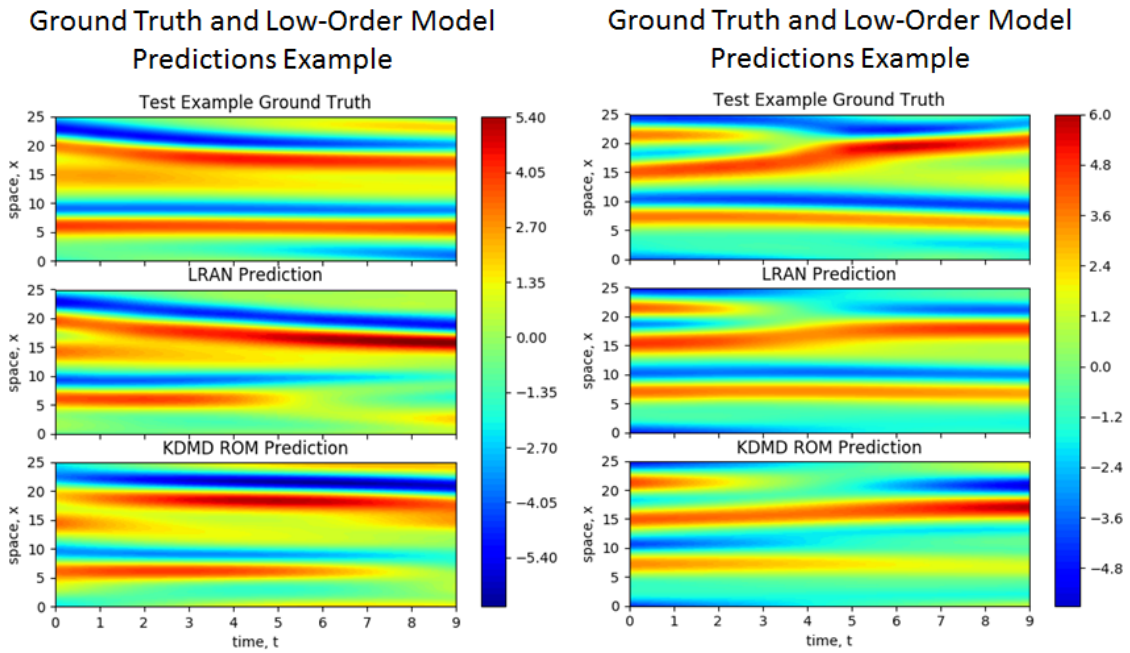


Figure 4.4.3: Cylinder wake testing data mean square relative prediction errors for each model plotted against the prediction time



(a) (b)

Figure 4.4.4: LRAN and KDMD ROM model predictions on Kuramoto-Sivashinsky test data examples

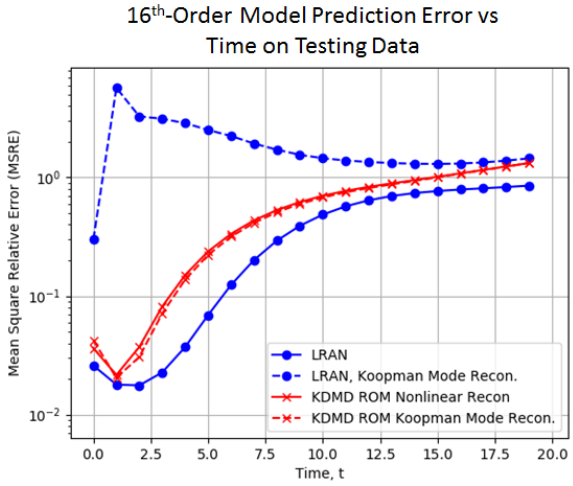


Figure 4.4.5: Kuramoto-Sivashinsky testing data mean square relative prediction errors for each model plotted against the prediction time

evolves. We illustrate this in [194]** by considering a Duffing oscillator

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\delta x_2 - x_1(\beta + \alpha x_1^2) \end{aligned} \tag{4.97}$$

with the same parameters $\delta = 0.5$, $\beta = -1$, and $\alpha = 1$ as in [280]. This system has an unstable saddle point at the origin and two stable spiral equilibria at $x_1 = \pm 1$ and $x_2 = 0$. As pointed out by M. O. Williams et al. [280] and described in Sections 4.2.1 and 4.2.2, the Duffing equation may be globally linearized in the basins of attraction corresponding to the two stable fixed points by Koopman eigenfunctions sharing the same eigenvalues as the system's linearization about the fixed points. In particular, the magnitude and phase of the eigenfunctions associated with these eigenvalues $\lambda_{1,2} = \frac{1}{4}(-1 \pm \sqrt{31}i)$ provide action-angle coordinates in each basin. There is also an eigenfunction with eigenvalue $\lambda_0 = 0$ that takes different constant values in each basin. Together, these eigenfunctions are sufficient to recover the state (x_1, x_2). We trained an LRAN with 5 latent states and constrained the evolution matrix $\boldsymbol{U}$ so that 3 of its 5 eigenvalues were $\lambda_0, \lambda_1, \lambda_2$. The corresponding approximate Koopman eigenfunctions shown in Figure 4.4.6 reveal the basins of attraction, the locations of all three fixed points, and qualitatively correct action-angle paramterizations of each basin.

4.4.2 Future work: introducing actuation and handling infinities

Originally, in [194]**, we considered only autonomous systems. However, it is very easy to incorporate actuation into an LRAN-like architecture by approximating an input-parametrized Koopman

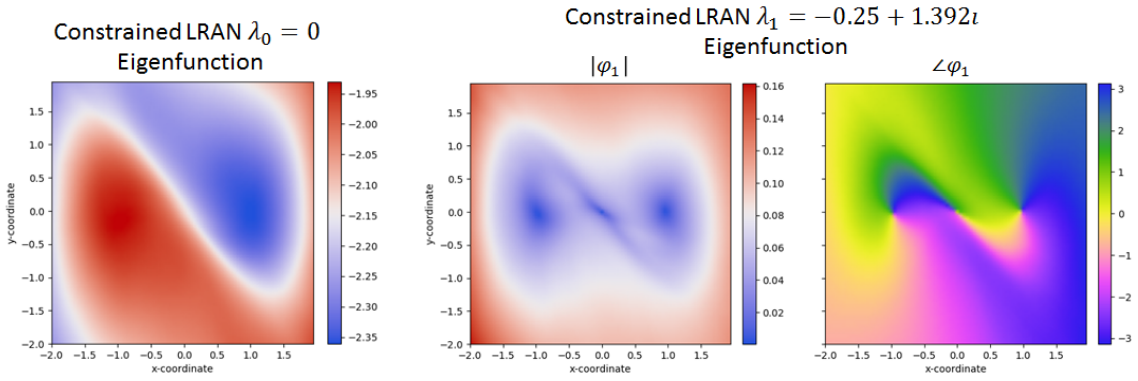


Figure 4.4.6: Duffing eigenfunction approximations learned using a constrained LRAN

generator. For an input-affine system, we may replace $\boldsymbol{U}$ with a matrix exponential of an input-affine approximation of the Koopman generator

$$\boldsymbol{U} = \exp\left(\Delta t \boldsymbol{V}_0 + \Delta t \sum_{i=1}^{m} [\boldsymbol{u}]_i \boldsymbol{V}_i\right). \tag{4.98}$$

Here, the matrix exponential must be differentiated in order to compute the gradient of the loss function with respect to each $\boldsymbol{V}_i$. Fortunately, the derivative is easily computed using the techniques described in [185], [169], and [7].

Another problem that occurs for systems with stable and unstable attractors, such as the cylinder wake flow and the Duffing equation is that the associated Koopman eigenfunctions take infinite values. For instance, the Koopman eigenfunction describing how the cylinder wake flow settles onto its stable limit cycle is equal to zero on the limit cycle, but must become infinite as the unstable equilibrium is approached. This is because it takes more time to reach the limit cycle the closer to the equilibrium the initial condition is chosen to be. One way to handle such infinities is to use steriographic projection to map the latent space onto a unit sphere of the same dimension with one point removed. In particular, the final layer of the encoder can be normalized to output points on the unit sphere, which are then mapped via steriographic projection to points in the latent space. After evolving the linear dynamics in the latent space, the steriorgraphic projection can be inverted to produce points on the unit sphere prior to applying the decoder. This way, the output of the encoder and the input of the decoder are always bounded (confined to the unit sphere), while allowing for latent states that approach infinity.

4.5 Learning bilinear approximations of Koopman generators using expectation-maximization

In principle, techniques like the LRAN can be used to learn accurate finite-dimensional approximations of the Koopman operator for actuated systems provided we have access to snapshot sequences of the system's state. When the system is not actuated, Takens' theorem [258, 189] allows us to replace the full state with an embedding obtained by time delayed observations of one or more observables. Time-delay embedding is useful in experimental settings where we only have access to a small number of sensor measurements or observations that do not capture the entire state. This approach is used in [130] and [74] to construct approximations of the Koopman operator. However, the time-delay embedding approach breaks down entirely when the data is obtained from trajectories with actuation. This is because the time-delayed observations now depend on the applied input rather than solely depending on the initial state in the time-delayed sequence. In many practical applications involving actuation of high-dimensional systems, such as controlling fluid flows, it would be very useful to construct approximations of the Koopman operator or generator without having to observe the entire state of the system.

In this section, we describe an approach for approximating actuated Koopman generators from partially-observed trajectories subject to actuation. Here, we learn the parameters of a bilinear Hidden Markov Model (HMM) approximating the action of the Koopman generator using an Expectation-Maximization (EM) algorithm. Another benefit of this approach is that it does not require an explicit choice of observables as in (E/K)DMD, and relies only on observed data along a collection of trajectories.

We construct our HMM based on the structure of finite-dimensional models of the Koopman generator for input-affine systems based on dictionaries. To simplify the presentation, we assume that we have an input-linear system,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, x &= \sum_{i=1}^{m} [\boldsymbol{u}]_i f_i(x), \\ \boldsymbol{y} &= \boldsymbol{g}(x) \end{aligned} \tag{4.99}$$

because we can always let $[\boldsymbol{u}]_1 \equiv 1$ if there is a drift term. We do not assume that we have a dictionary; but if we did, it would be a vector-valued function $\boldsymbol{\psi} = (1, \psi_1 \ldots, \psi_N) = (1, \tilde{\boldsymbol{\psi}}) : \mathcal{X} \to \mathbb{C}^{N+1}$, where the constant function is always to be included by design. A finite-dimensional

approximation of the Koopman generator for Eq. 4.99 using such a dictionary has the form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\boldsymbol{\psi}(x(t)) &= \sum_{i=1}^{m}[\boldsymbol{u}]_i \boldsymbol{V}_i^T \boldsymbol{\psi}(x(t)), & \boldsymbol{V} &= \begin{bmatrix} \boldsymbol{0} & \tilde{\boldsymbol{V}} \end{bmatrix} \\ \boldsymbol{y} = \boldsymbol{g}(x(t)) &= \boldsymbol{C}\boldsymbol{\psi}, & \boldsymbol{C} &= \begin{bmatrix} \boldsymbol{c}_0 & \tilde{\boldsymbol{V}} \end{bmatrix}. \end{aligned} \tag{4.100}$$

If the rows of $\tilde{\boldsymbol{C}}$ are linearly independent, then we may always find a linear transformation $\tilde{\boldsymbol{T}}$ of the unspecified observables $\tilde{\boldsymbol{\psi}}$ such that

$$\boldsymbol{C}\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{h} & \tilde{\boldsymbol{C}}\tilde{\boldsymbol{T}}^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{\boldsymbol{T}}\tilde{\boldsymbol{\psi}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_0 & \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{\boldsymbol{T}}\tilde{\boldsymbol{\psi}} \end{bmatrix}. \tag{4.101}$$

Since $\boldsymbol{\psi}$ are to be determined implicitly, we assume that $\boldsymbol{C}$ always takes the above form where only $\boldsymbol{c}_0$ is to be determined.

The key feature of the model Eq. 4.100 is that the values taken by the observables $\tilde{\boldsymbol{\psi}}(x(t))$ play the same role as the hidden variables in a HMM. In particular, if we know the matrices $\boldsymbol{V}_i$ and $\boldsymbol{C}$, then it is possible to estimate the values of $\tilde{\boldsymbol{\psi}}(x(t))$ given observations $\boldsymbol{y}(t)$ along a trajectory. We build our HMM from Eq. 4.100 by replacing $\tilde{\boldsymbol{\psi}}(x(t))$ with a hidden variable $\boldsymbol{z}$. The evolution of the hidden variable is modeled in discrete time using an explicit Euler discretization of Eq. 4.100. In particular, we construct the HMM

$$\boxed{\begin{aligned} \begin{bmatrix} 1 \\ \boldsymbol{z}_{l+1} \end{bmatrix} &= \left(\boldsymbol{I} + \Delta t \sum_{i=1}^{m} [\boldsymbol{u}_l]_i \boldsymbol{V}_i \right)^T \begin{bmatrix} 1 \\ \boldsymbol{z}_l \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{v}_l \end{bmatrix} \\ \boldsymbol{y}_l &= \boldsymbol{c}_0 + \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \boldsymbol{z}_l + \boldsymbol{w}_l, \end{aligned}} \tag{4.102}$$

where the process and measurement noise $\boldsymbol{v}_l$ and $\boldsymbol{w}_l$ are assumed to be independent identically distributed zero mean and Gaussian with covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{v}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{w}}$. We also assume that the initial conditions for $\boldsymbol{z}_0$ are Gaussian with mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$.

In the next section, we shall describe an Expectation-Maximization (EM) algorithm that can estimate the parameters,

$$\mathcal{P} = \left\{ \tilde{\boldsymbol{V}}_1, \ldots, \tilde{\boldsymbol{V}}_m, \boldsymbol{c}_0, \boldsymbol{\Sigma}_{\boldsymbol{v}}, \boldsymbol{\Sigma}_{\boldsymbol{w}}, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0 \right\}, \tag{4.103}$$

in the HMM Eq. 4.102 from a collection of time histories of observations $\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_L\}$ with known

inputs $\{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{L-1}\}$. The EM algorithm is an iterative approach that begins with an initial guess for the parameters $\mathcal{P}$ and proceeding in two steps, an Expectation or E-step and a Maximization or M-step. In the E-step, the parameters $\mathcal{P}$ are fixed and we use a standard Kalman filter and smoother to construct optimal estimates of the hidden variables $\boldsymbol{z}_l$ given the observations $\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_L\}$. With the optimal hidden variable estimates in hand, we solve least squares problems to update the parameters $\mathcal{P}$ of the model during the M-step in a way that always increases the likelihood, or probability density, of generating the outputs $\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_L\}$ from Eq. 4.102. Remarkably, the least squares problem we solve during the M-step to update $\{\boldsymbol{V}_1, \ldots, \boldsymbol{V}_m\}$ closely resembles the EDMD method for bilinear Koopman generator we developed in [202]* and described in Section 4.3.2.

4.5.1 EM Algorithm for Learning Koopman Generator Approximations

Maximum likelihood estimation entails maximizing the probability of the observed data over the model parameters. In particular, let us combine all of our observations $\{\boldsymbol{y}_l^{(m)}\}_{0 \leq l \leq L}$ along the mth independent trajectory into a matrix $\boldsymbol{Y}^{(m)}$ and denote the joint probability density of these observations under the model parameters by $\tilde{\boldsymbol{Y}} \mapsto P_{\boldsymbol{Y}}(\tilde{\boldsymbol{Y}};\ \mathcal{P})$. We aim to maximize the log "likelihood" of these observations given by

$$L(\mathcal{P}) = \sum_{m=1}^{M} \log P_{\boldsymbol{Y}}(\boldsymbol{Y}^{(m)};\ \mathcal{P}). \tag{4.104}$$

The log likelihood is used instead of the raw likelihood because the probability $P_{\boldsymbol{Y}}(\boldsymbol{Y}^{(m)};\ \mathcal{P})$ factors into a product of many terms due to the Markov property of the model Eq. 4.102.

The required density $P_{\boldsymbol{Y}}(\boldsymbol{Y};\ \mathcal{P})$ can be expressed using the model Eq. 4.102 by recognizing that it is a marginal distribution

$$P_{\boldsymbol{Y}}(\boldsymbol{Y}^{(m)};\ \mathcal{P}) = \int P_{\boldsymbol{Z},\boldsymbol{Y}}(\boldsymbol{Z}, \boldsymbol{Y}^{(m)};\ \mathcal{P}) d\boldsymbol{Z}, \tag{4.105}$$

where $P_{\boldsymbol{Z},\boldsymbol{Y}}$ is the joint density of the observed trajectory $\boldsymbol{Y}$ and the hidden variables $\{\boldsymbol{z}_l\}_{l=0}^{L}$ stacked into a matrix $\boldsymbol{Z}$. Yet this high-dimensional integral is rather difficult to evaluate and makes direct optimization of Eq. 4.104 futile. By introducing a new probability density $\boldsymbol{Z} \mapsto Q^{(m)}(\boldsymbol{Z})$ that is to be determined and a random variable $\hat{\boldsymbol{Z}}^{(m)}$ with density $Q^{(m)}$, the above integral is converted into an expectation

$$P_{\boldsymbol{Y}}(\boldsymbol{Y}^{(m)};\ \mathcal{P}) = \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}}\left[\frac{P_{\boldsymbol{Z},\boldsymbol{Y}}(\hat{\boldsymbol{Z}}^{(m)}, \boldsymbol{Y}^{(m)};\ \mathcal{P})}{Q(\hat{\boldsymbol{Z}}^{(m)})}\right]. \tag{4.106}$$

Recognizing that log is a concave function, we apply Jensen's inequality to obtain a lower bound on the log likelihood,

$$\boxed{\begin{aligned} L(\mathcal{P}) \geq \hat{L}_Q(\mathcal{P}) &= \sum_{m=1}^{M} \left\{ \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}} \left[\log P_{\boldsymbol{Z},\boldsymbol{Y}}(\hat{\boldsymbol{Z}}^{(m)}, \boldsymbol{Y}^{(m)};\ \mathcal{P}) \right] - \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}} \left[\log Q^{(m)}(\hat{\boldsymbol{Z}}^{(m)}) \right] \right\} \\ &= L(\mathcal{P}) - \sum_{m=1}^{M} D_{KL}(Q^{(m)} \| P_{\boldsymbol{Z}|\boldsymbol{Y}=\boldsymbol{Y}^{(m)}}), \end{aligned}} \tag{4.107}$$

which is commonly referred to as the variational lower bound [26] or Evidence Lower Bound (ELBO) [27]. The quantity

$$D_{KL}(Q^{(m)} \| P_{\boldsymbol{Z}|\boldsymbol{Y}=\boldsymbol{Y}^{(m)}}) = \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}} \left[\log \left(\frac{Q^{(m)}(\hat{\boldsymbol{Z}}^{(m)})}{P_{\boldsymbol{Z}|\boldsymbol{Y}=\boldsymbol{Y}^{(m)}}(\hat{\boldsymbol{Z}}^{(m)};\ \mathcal{P})} \right) \right] \tag{4.108}$$

is called the Kullback-Leibler (KL)-divergence or the entropy of $P_{\boldsymbol{Z}|\boldsymbol{Y}=\boldsymbol{Y}^{(m)}}$ relative to $Q^{(m)}$.

A key observation is that the inequality in Eq. 4.107 becomes equality when the probability density $Q^{(m)}$ is chosen to be the conditional density of $\boldsymbol{Z}$ given $\boldsymbol{Y}^{(m)}$, that is

$$Q^{(m)} = P_{\boldsymbol{Z}|\boldsymbol{Y}=\boldsymbol{Y}^{(m)}},\ m = 1, \ldots, M \quad \Rightarrow \quad L(\mathcal{P}) = \hat{L}_Q(\mathcal{P}). \tag{4.109}$$

This is an immediate consequence of the definition of the KL-divergence in Eq. 4.108. For this reason, Q is often referred to as the "inference" distribution since its optimal form allows one to infer the values of the latent variables from the observations.

The ELBO Eq. 4.107 and the observation Eq. 4.109 suggest a kind of coordinate ascent optimization procedure for maximizing the likelihood that iteratively updates the inference distributions $Q^{(m)}$ with the parameters $\mathcal{P}$ fixed and then updates the parameters $\mathcal{P}$ with the inference distributions $Q^{(m)}$ fixed. This is called the Expectation-Maximization (EM) algorithm [78] because in the first step one computes the inference distribution at the current model parameters $\mathcal{P}_0$ and uses it to assemble the ELBO $\mathcal{P} \mapsto \hat{L}_Q(\mathcal{P})$ as an expectation Eq. 4.107 that is maximized in the second step. If $\mathcal{P}_1$ are the new model parameters found by the maximization step, then we have

$$L(\mathcal{P}_1) \geq \hat{L}_Q(\mathcal{P}_1) \geq \hat{L}_Q(\mathcal{P}_0) = L(\mathcal{P}_0), \tag{4.110}$$

where the first inequality is Eq. 4.107, the second inequality is due to maximization, and the equality on the right is by definition of the expectation step and Eq. 4.109. This means that *the EM*

algorithm always produces a sequence of models with increasing log likelihoods until convergence when the maximization step fails to produce an increase in $\hat{L}_Q$. The convergence properties of the EM algorithm were studied by C. F. Jeff Wu in [285]. It is shown that if the superlevel set $\{\mathcal{P} : L(\mathcal{P}) \geq L(\mathcal{P}_0)\}$ is compact, then the limit points of the sequence of EM iterates $\mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, ...$ are stationary points of the likelihood function. However, unless certain, challenging to verify conditions are met, the EM algorithm may converge to a local maximum or even a saddle point instead of a desired global maximum of the likelihood. For more details about the EM algorithm, one can consult [26] and [27].

Remark 4.5.1. *The compactness assumption on* $\{\mathcal{P} : L(\mathcal{P}) \geq L(\mathcal{P}_0)\}$ *may be satisfied via regularization of the log likelihood function by introducing a prior probability density for the model's parameters. However, we have not found this to be necessary in practice when there is enough data to constrain the model's parameters.*

A key feature of the evidence lower bound given by Eq. 4.107 for our model Eq. 4.102 is that maximization over the parameters $\mathcal{P}$ has an explicit solution that we state in Theorem 4.5.2.

Theorem 4.5.2 (Maximization Step)**.** *Supposing that the log likelihood function* $\mathcal{P} \mapsto L(\mathcal{P})$ *in Eq. 4.104 is bounded from above, let us denote the mean and joint covariance of the fixed inference distributions* $Q^{(m)}$, $m = 1, \ldots, M$ *by*

$$\hat{\boldsymbol{\mu}}_k^{(m)} = \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}}\left[\hat{\boldsymbol{z}}_k^{(m)}\right] \qquad \text{and} \qquad \hat{\boldsymbol{\Sigma}}_{k,l}^{(m)} = \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}}\left[\left(\hat{\boldsymbol{z}}_k^{(m)} - \hat{\boldsymbol{\mu}}_k^{(m)}\right)\left(\hat{\boldsymbol{z}}_l^{(m)} - \hat{\boldsymbol{\mu}}_l^{(m)}\right)^T\right]. \tag{4.111}$$

and define the matrices

$$\boldsymbol{G}_l^{(m)} = \begin{bmatrix} 1 & (\hat{\boldsymbol{\mu}}_l^{(m)})^T \\ \hat{\boldsymbol{\mu}}_l^{(m)} & \hat{\boldsymbol{\Sigma}}_{l,l}^{(m)} + \hat{\boldsymbol{\mu}}_l^{(m)}(\hat{\boldsymbol{\mu}}_l^{(m)})^T \end{bmatrix}, \tag{4.112}$$

$$\tilde{\boldsymbol{H}}_l^{(m)} = \begin{bmatrix} \left(\frac{\hat{\boldsymbol{\mu}}_{l+1}^{(m)} - \hat{\boldsymbol{\mu}}_l^{(m)}}{\Delta t}\right) & \left(\frac{\hat{\boldsymbol{\Sigma}}_{l+1,l}^{(m)} - \hat{\boldsymbol{\Sigma}}_{l+1,l}^{(m)}}{\Delta t} + \frac{\hat{\boldsymbol{\mu}}_{l+1}^{(m)} - \hat{\boldsymbol{\mu}}_l^{(m)}}{\Delta t}(\hat{\boldsymbol{\mu}}_l^{(m)})^T\right) \end{bmatrix}. \tag{4.113}$$

Then, with $\otimes$ *denoting the Kronecker product, the parameters* $\mathcal{P}$ *that maximize the evidence lower bound* $\hat{L}_Q(\mathcal{P})$ *in Eq. 4.107 are given by*

$$\boldsymbol{\mu}_0 = \frac{1}{M}\sum_{m=1}^{M} \hat{\boldsymbol{\mu}}_0^{(m)}, \tag{4.114}$$

$$\boldsymbol{\Sigma}_0 = \frac{1}{M}\sum_{m=1}^{M}\left[\hat{\boldsymbol{\Sigma}}_{0,0}^{(m)} + \left(\hat{\boldsymbol{\mu}}_0^{(m)} - \boldsymbol{\mu}_0\right)\left(\hat{\boldsymbol{\mu}}_0^{(m)} - \boldsymbol{\mu}_0\right)^T\right] \tag{4.115}$$

$$\boldsymbol{c}_0 = \frac{1}{M(L+1)} \sum_{m=1}^{M} \sum_{l=0}^{L} \left(\boldsymbol{y}_l^{(m)} - \tilde{\boldsymbol{C}} \hat{\boldsymbol{\mu}}_l^{(m)} \right) \tag{4.116}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{v}} = \frac{1}{M(L+1)} \sum_{m=1}^{M} \sum_{l=0}^{L} \left[\tilde{\boldsymbol{C}} \hat{\boldsymbol{\Sigma}}_{l,l}^{(m)} \tilde{\boldsymbol{C}}^T + \left(\boldsymbol{y}_l^{(m)} - \boldsymbol{c}_0 - \tilde{\boldsymbol{C}} \hat{\boldsymbol{\mu}}_l^{(m)} \right) \left(\boldsymbol{y}_l^{(m)} - \boldsymbol{c}_0 - \tilde{\boldsymbol{C}} \hat{\boldsymbol{\mu}}_l^{(m)} \right)^T \right] \tag{4.117}$$

$$\boxed{\begin{bmatrix} \tilde{\boldsymbol{V}}_0 \\ \vdots \\ \tilde{\boldsymbol{V}}_{\dim \boldsymbol{u}} \end{bmatrix} = \left(\sum_{m=1}^{M} \sum_{l=0}^{L-1} \boldsymbol{u}_l^{(m)} \otimes (\boldsymbol{u}_l^{(m)})^T \otimes \boldsymbol{G}_l^{(m)} \right)^{-1} \left(\sum_{m=1}^{M} \sum_{l=0}^{L-1} \boldsymbol{u}_l^{(m)} \otimes (\tilde{\boldsymbol{H}}_l^{(m)})^T \right)} \tag{4.118}$$

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{w}} = \frac{1}{ML} \sum_{m=1}^{M} \sum_{l=0}^{L-1} \Big[& \hat{\boldsymbol{\Sigma}}_{l+1,l+1}^{(m)} - \boldsymbol{A}_l^{(m)} \hat{\boldsymbol{\Sigma}}_{l,l+1}^{(m)} - \hat{\boldsymbol{\Sigma}}_{l+1,l}^{(m)} (\boldsymbol{A}_l^{(m)})^T + \boldsymbol{A}_l^{(m)} \hat{\boldsymbol{\Sigma}}_{l,l}^{(m)} (\boldsymbol{A}_l^{(m)})^T \\ & + \left(\hat{\boldsymbol{\mu}}_{l+1}^{(m)} - \boldsymbol{A}_l^{(m)} \hat{\boldsymbol{\mu}}_l^{(m)} - \boldsymbol{b}_l^{(m)} \right) \left(\hat{\boldsymbol{\mu}}_{l+1}^{(m)} - \boldsymbol{A}_l^{(m)} \hat{\boldsymbol{\mu}}_l^{(m)} - \boldsymbol{b}_l^{(m)} \right)^T \Big]. \end{aligned} \tag{4.119}$$

Proof. The proof is involved, so we give it in Appendix 4.A. □

Remark 4.5.3 (connection with input-affine EDMD for the generator)**.** *It is interesting to observe that the solution for the generators during the maximization step of the EM algorithm provided by Theorem 4.5.2 in Eq. 4.118 can be viewed as a limiting case of the control-affine extended dynamic mode decomposition technique presented in [202]*. In particular, suppose we draw K independent identically distributed trajectories $\{\hat{\boldsymbol{z}}_l^{(m_k)}\}_{l=0}^{L}$, $k = 1, \ldots, K$ uniformly at random from the posterior distributions $\{P_{\boldsymbol{Z}|\boldsymbol{Y}=\boldsymbol{Y}^{(1)}}, \ldots, P_{\boldsymbol{Z}|\boldsymbol{Y}=\boldsymbol{Y}^{(M)}}\}$. If we construct the matrices*

$$\boldsymbol{\Psi}_K = \begin{bmatrix} \boldsymbol{u}_0^{(m_1)} \otimes \begin{pmatrix} 1 \\ \hat{\boldsymbol{z}}_0^{(m_1)} \end{pmatrix} & \cdots & \boldsymbol{u}_{L-1}^{(m_K)} \otimes \begin{pmatrix} 1 \\ \hat{\boldsymbol{z}}_{L-1}^{(m_K)} \end{pmatrix} \end{bmatrix}, \tag{4.120}$$

$$\dot{\boldsymbol{\Psi}}_K = \begin{bmatrix} \left(\frac{\hat{\boldsymbol{z}}_1^{(m_1)} - \hat{\boldsymbol{z}}_0^{(m_1)}}{\Delta t} \right) & \cdots & \left(\frac{\hat{\boldsymbol{z}}_L^{(m_K)} - \hat{\boldsymbol{z}}_{L-1}^{(m_K)}}{\Delta t} \right) \end{bmatrix}, \tag{4.121}$$

then the same generator approximations given by Theorem 4.5.2 in Eq. 4.118 are found in the limit

$$\begin{bmatrix} \tilde{\boldsymbol{V}}_0 \\ \vdots \\ \tilde{\boldsymbol{V}}_{\dim \boldsymbol{u}} \end{bmatrix} = \lim_{K \to \infty} \left(\boldsymbol{\Psi}_K \boldsymbol{\Psi}_K^T \right) \boldsymbol{\Psi}_K \dot{\boldsymbol{\Psi}}_K^T. \tag{4.122}$$

Each term in the sequence are the generators computed by [202] from the data $\{\hat{\boldsymbol{z}}_l^{(m_k)}\}_{l=0}^{L}$, $k =$*

$1, \ldots, K$.

Another important observation is that the explicit solutions for the parameters during the maximization step of the EM algorithm given in Theorem 4.5.2 depend only on a few parameters of the inference distribution. Therefore, during the expectation or E-step of the EM algorithm, we need only compute the conditional or "posterior" mean and covariance

$$\hat{\boldsymbol{\mu}}_l^{(m)} = \mathbb{E}\left[\boldsymbol{z}_l \mid \boldsymbol{Y} = \boldsymbol{Y}^{(m)}\right], \qquad \hat{\boldsymbol{\Sigma}}_{l,l}^{(m)} = \mathbb{E}\left[\left(\boldsymbol{z}_l - \hat{\boldsymbol{\mu}}_l^{(m)}\right)\left(\boldsymbol{z}_l - \hat{\boldsymbol{\mu}}_l^{(m)}\right)^T \mid \boldsymbol{Y} = \boldsymbol{Y}^{(m)}\right] \tag{4.123}$$

at each time step given the observations $\boldsymbol{Y}^{(m)}$ along each trajectory as well as the posterior covariance between adjacent time steps

$$\hat{\boldsymbol{\Sigma}}_{l,l+1}^{(m)} = \mathbb{E}\left[\left(\boldsymbol{z}_l - \hat{\boldsymbol{\mu}}_l^{(m)}\right)\left(\boldsymbol{z}_{l+1} - \hat{\boldsymbol{\mu}}_{l+1}^{(m)}\right)^T \mid \boldsymbol{Y} = \boldsymbol{Y}^{(m)}\right]. \tag{4.124}$$

Fortunately, there is a very efficient algorithm due to R. H. Shumway and D. S. Stoffer [249] for computing these conditional expectations that proceeds by passing over each trajectory twice. On the first pass, we move forward along the trajectory using a Kalman filter [129] to assimilate the observations made up to each given time step. On the second pass, we move backward along the trajectory using a smoother [217] to assimilate the observations made after each given time step.

In particular, we recall that with the parameters $\mathcal{P}$ fixed, as the they are during the expectation step of the EM algorithm, the dynamics of our discrete-time model Eq. 4.102 can be written as

$$\begin{aligned} \boldsymbol{z}_{l+1} &= \boldsymbol{A}_l \boldsymbol{z}_l + \boldsymbol{b}_l + \boldsymbol{w}_l \\ \boldsymbol{y}_l &= \boldsymbol{c}_0 + \tilde{\boldsymbol{C}} \boldsymbol{z}_l + \boldsymbol{v}_l, \end{aligned} \tag{4.125}$$

where $\boldsymbol{A}_l$, $\boldsymbol{b}_l$, $\boldsymbol{c}_0$, and $\tilde{\boldsymbol{C}}$ are all known. Furthermore, the initial condition, process noise, and measurement noise have independent Gaussian distributions

$$\boldsymbol{z}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0), \qquad \boldsymbol{w}_l \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{w}}), \qquad \boldsymbol{v}_l \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{v}}), \tag{4.126}$$

with known means and covariances. Therefore, computing the posterior means and covariances in Eqs. 4.123 and 4.124 of the states $\boldsymbol{z}_l$ given the observations $\mathbf{Y}^{(m)}$ along the mth independent trajectory is a standard state estimation problem for the linear time-varying dynamical system Eq. 4.125. The appropriate Kalman filtering and smoothing equations can be found by trivially modifying those found in Byron et al. [42] and Ghahramani et al. [94] to allow for time-varying

system matrices.

4.5.2 Preliminary results and future work

We consider a toy system from [196]*,

$$\begin{aligned} \dot{x}_1 &= -\alpha x_1 + u \\ \dot{x}_2 &= \beta\left(x_1^3 - x_2\right) \\ y &= x_2 + w, \end{aligned} \tag{4.127}$$

which was originally adapted based on a similar model in [33]. This system can be described explicitly in a finite-dimensional Koopman-invariant subspace as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1^3 \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -\alpha & 0 & 0 & 0 \\ 0 & 0 & -\beta & 0 & \beta \\ 0 & 0 & 0 & -2\alpha & 0 \\ 0 & 0 & 0 & 0 & -3\alpha \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1^3 \end{bmatrix} + u \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1^3 \end{bmatrix} \\ y &= x_2 + w. \end{aligned} \tag{4.128}$$

We choose $\alpha = 1$ and $\beta = 5$ with zero process noise and Gaussian measurement noise w with zero mean and variance 0.01. The inputs are held constant over time intervals of length 0.5 and take values drawn from a Gaussian distribution with zero mean and variance $\sigma_u^2 = 5$.

We collected 50 independent trajectories with 500 observations recorded at intervals $\Delta t = 0.01$. The initial conditions were drawn from an isotropic Gaussian distribution with zero mean and unit variance. The first 250 points along each trajectory were used for training while the rest were saved for testing the model.

After training the model, we used it to predict a trajectory and its uncertainty shown in Figure 4.5.1 on the left. This was done by using the training portion of the trajectory to obtain an optimal estimate of the initial condition as well as its uncertainty. The mean and uncertainty were then propagated forward by the model dynamics from the initial condition to produce the predicted observations and 2σ confidence envelope shown in Figure 4.5.1. The eigenvalues of the matrix approximation of the drift Koopman generator learned by our model are compared to the ground truth eigenvalues in Figure 4.5.1 on the right, showing excellent agreement.

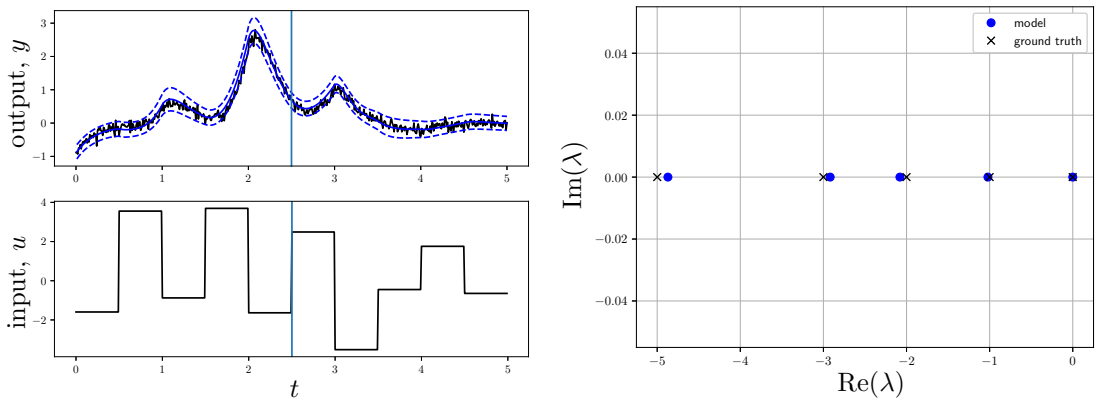


Figure 4.5.1: Left: Observations from the actuated toy model (black lines) together with our model's prediction (blue line) and 2σ confidence interval (dashed blue lines). The initial condition for the model prediction was computed using optimal state estimation from the training interval (left of vertical line). Right: Eigenvalues of the matrix approximation for the drift Koopman generator learned by the EM algorithm for the toy model.

In the above example, it was possible to find an exact Koopman invariant subspace for the system containing observables from which the observed state could be linearly reconstructed. However, for more complicated systems, this may not be possible and the EM algorithm will have to find the closest approximation in the form of our bilinear model Eq. 4.102, even though no such model describes the system perfectly. On such problems we may incur significant error by demanding that the observations are reconstructed linearly from a low-dimensional approximately Koopman-invariant subspace.

As we found in [194]** and in Section 4.4, allowing for nonlinear reconstruction from an approximate Koopman-invariant subspace can improve the accuracy of the model's predictions. However, incorporating a nonlinear decoder into our hidden Markov model will require a more complicated training process than for Eq. 4.102. In particular, the nonlinear reconstruction function would require us to use techniques like particle or unscented filtering during the E-step. Depending on how the nonlinear reconstruction map is parametrized, we may need to use gradient descent during the M-step to optimize these parameters. With this added complexity and cost associated with the nonlinear reconstruction map, it no longer simplifies the method to use discrete-time latent space dynamics with the same time step as the sampling. Instead, we could learn the parameters of a bilinear stochastic differential equation governing the latent state dynamics. Developing such a model would be an interesting direction for future work.

Appendix

4.A Chapter 4 Proofs

Proof of Theorem 4.1.1. We begin by showing that each U^t is well-defined on $C_0(\mathcal{X})$, i.e., that $f \circ F^t \in C_0(\mathcal{X})$ for every $f \in C_0(\mathcal{X})$. Throughout the proof, the norm $\|\cdot\|$ is understood to be the norm on $C(\mathcal{X})$. Choose any $f \in C_0(\mathcal{X})$ and observe that $f \circ F^t \in C(\mathcal{X})$ because the composition of continuous functions is continuous. Choose any $\varepsilon > 0$ and take $f_\varepsilon \in C_c(\mathcal{X})$ such that $\|f - f_\varepsilon\| < \varepsilon$. Let $\pi : \mathcal{X} \times [0,\infty) \to \mathcal{X}$ be the canonical projection defined by $\pi(x,t) = x$. Since $\operatorname{supp} f_\varepsilon \times \{t\}$ is compact in $\mathcal{X} \times [0,\infty)$, our assumption that $\tilde{F}$ is proper implies that

$$\operatorname{supp}\left(f_\varepsilon \circ F^t\right) = (F^t)^{-1}(\operatorname{supp} f_\varepsilon) = \pi\left(\tilde{F}^{-1}(\operatorname{supp} f_\varepsilon \times \{t\})\right) \tag{4.129}$$

is compact for every $t \geq 0$. Therefore, $f_\varepsilon \circ F^t \in C_c(\mathcal{X})$ and

$$\|f \circ F^t - f_\varepsilon \circ F^t\| \leq \|f - f_\varepsilon\| < \varepsilon, \tag{4.130}$$

proving that $f \circ F^t$ is in the closure of $C_c(\mathcal{X})$ in $C(\mathcal{X})$, i.e., $f \circ F^t \in C_0(\mathcal{X})$.

Now we prove that $\{U^t\}_{t\geq 0}$ defined on $C_0(\mathcal{X})$ is strongly continuous. Choose any $\varepsilon > 0$ and any $f \in C_0(\mathcal{X})$ and take $f_\varepsilon \in C_c(\mathcal{X})$ such that $\|f - f_\varepsilon\| < \varepsilon$. Since $\tilde{F}$ is proper and $\operatorname{supp} f_\varepsilon \times [0,1] \subset \mathcal{X} \times [0,\infty)$ is compact, the set

$$K = \bigcup_{t\in[0,1]} \operatorname{supp}\left(f_\varepsilon \circ F^t\right) = \pi\left(\tilde{F}^{-1}(\operatorname{supp} f_\varepsilon \times [0,1])\right) \tag{4.131}$$

is compact. Since the function defined by $(x,t) \mapsto |f_\varepsilon(F(x,t)) - f_\varepsilon(x)|$ is continuous on $\mathcal{X} \times [0,\infty)$, the pre-image set

$$\mathcal{V} = \{(x,t) \in \mathcal{X} \times [0,\infty) \ : \ |f_\varepsilon(F(x,t)) - f_\varepsilon(x)| < \varepsilon\} \tag{4.132}$$

is open in $\mathcal{X} \times [0,\infty)$. Moreover, $\mathcal{V}$ contains $\mathcal{X} \times \{0\}$ because $F(x,0) = x$ for every $x \in \mathcal{X}$. Consequently, $\mathcal{V}$ contains an open neighborhood of $(x,0)$ for each $x \in \mathcal{X}$. In particular, for every $x \in K$ there is an open set $\mathcal{U}_x \subset \mathcal{X}$ and $\delta_x > 0$ such that $x \in \mathcal{U}_x$ and the cylinder $\mathcal{U}_x \times [0,\delta_x)$ is contained in $\mathcal{V}$. The sets $\{U_x\}_{x\in K}$ form an open cover of K, and so we may extract a finite sub-cover $\{U_i = U_{x_i}\}_{i=1}^M$ of K and the corresponding values $\delta_i = \delta_{x_i}$. Taking

$$0 \leq t < \delta = \min\{1, \delta_1, \ldots, \delta_M\} \tag{4.133}$$

ensures that $|f_\varepsilon(F(x,t)) - f_\varepsilon(x)| < \varepsilon$ for every $x \in K$, This follows because every $x \in K$ is contained in some U_i and for $0 \leq t < \delta$ we have $(x,t) \in U_i \times [0,\delta_i) \subset \mathcal{V}$. Since $\operatorname{supp}(f_\varepsilon \circ F^t) \subset K$ for all $t \in [0,\delta)$, we have

$$\|f_\varepsilon \circ F^t - f_\varepsilon\| = \sup_{x \in K} |f_\varepsilon(F(x,t)) - f_\varepsilon(x)| < \varepsilon. \tag{4.134}$$

Finally, for $0 \leq t < \delta$ we obtain

$$\|U^t f - f\| \leq \|U^t f - U^t f_\varepsilon\| + \|U^t f_\varepsilon - f_\varepsilon\| + \|f - f_\varepsilon\| < 3\varepsilon \tag{4.135}$$

proving the strong continuity of the Koopman semigroup on $C_0(\mathcal{X})$.

To prove that the generator V of the Koopman semigroup is the closure of $\tilde{V}$, we follow Example 3.28 in [86]. In particular, it suffices to show that $\mathrm{Dom}(\tilde{V}) = C_c^1(\mathcal{X})$ is a core of $\mathrm{Dom}(V)$, that is, a subset of $\mathrm{Dom}(V)$ which is dense in the graph norm $\|x\|_V = \|x\| + \|Vx\|$. By Proposition 1.7 in Chapter 2 of [86], to show that a subspace $D \subset \mathrm{Dom}(V)$ is a core, it suffices to show that D is invariant under $\{U^t\}_{t \geq 0}$ and that D is dense in $\mathcal{F}$. We observe that $C_c^1(\mathcal{X})$ is dense in $C_0(\mathcal{X})$ and invariant under $\{U^t\}_{t \geq 1}$ when F is continuously differentiable. Thus, it remains to show that $C_c^1(\mathcal{X})$ is a subset of $\mathrm{Dom}(V)$ on which $\tilde{V}$ agrees with V. By Theorem 1.10 in Chapter 2 of [86], the resolvent operator $(\lambda I - V)^{-1} : \mathcal{F} \to \mathrm{Dom}(V)$ is given by the improper Riemann integral

$$(\lambda I - V)^{-1} = \int_0^\infty e^{-\lambda t} U^t \,\mathrm{d}\, t, \tag{4.136}$$

when λ is in the resolvent set $\rho(V)$, i.e., when $\lambda I - V : \mathrm{Dom}(V) \to \mathcal{F}$ is bijective. Moreover, since $\|U^t\| \leq 1$ for every $t \geq 0$, every λ with $\mathrm{Re}(\lambda) > 0$ is in the resolvent set. Choosing any $f \in C_c^1(\mathcal{X})$ we therefore have

$$(I - V)^{-1}(I - \tilde{V})f = \int_0^\infty e^{-t} f \circ F^t \,\mathrm{d}\, t - \int_0^\infty e^{-t} \frac{\mathrm{d}}{\mathrm{d}\, \mathrm{t}} \left(f \circ F^t\right) \mathrm{d}\, t = f, \tag{4.137}$$

from which we conclude that $f \in \mathrm{Dom}(V)$ and $\tilde{V} f = V f$. This completes the proof that V is the closure of $\tilde{V}$ in the graph norm. □

Proof of Lemma 4.2.1 (Product rule for Koopman generators). The proof is essentially the same as the proof of the product rule in calculus. Since $\psi_2 \in \mathrm{Dom}(V)$, we must have $U^t \psi_2(x) \to \psi_2(x)$ for

every $x \in \mathcal{X}$. Consequently, for any $x \in \mathcal{X}$ we have

$$\begin{aligned} V(\psi_1\psi_2)(x) &= \lim_{t\to 0} \frac{(U^t\psi_1)(x)(U^t\psi_2)(x) - \psi_1(x)(U^t\psi_2)(x) + \psi_1(x)(U^t\psi_2)(x) - \psi_1(x)\psi_2(x)}{t} \\ &= \lim_{t\to 0} \left\{ (U^t\psi_2)(x)\frac{(U^t\psi_1)(x) - \psi_1(x)}{t} + \psi_1(x)\frac{(U^t\psi_2)(x) - \psi_2(x)}{t} \right\} \\ &= \psi_2(x)(V\psi_1)(x) + \psi_1(x)(V\psi_2)(x). \end{aligned} \tag{4.138}$$

Since $\mathcal{F}$ is closed under point-wise multiplication and $V\psi_1, V\psi_2 \in \mathcal{F}$, it follows that $V(\psi_1\psi_2) \in \mathcal{F}$, and so $\psi_1\psi_2 \in \mathrm{Dom}(V)$ and Eq. 4.49 holds. □

Proof of Theorem 4.5.2 (Maximization step of EM algorithm). During maximization of the evidence lower bound $\hat{L}_Q(\mathcal{P})$ in Eq. 4.107 over the parameters $\mathcal{P}$ with fixed inference distributions $\{Q^{(m)}\}_{m=1}^M$, we need only consider the first term

$$\sum_{m=1}^{M} \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}} \left[\log P_{\boldsymbol{Z},\boldsymbol{Y}}(\hat{\boldsymbol{Z}}^{(m)}, \boldsymbol{Y}^{(m)};\ \mathcal{P}) \right],$$

since the second term $\sum_{m=1}^M \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}} \left[\log Q^{(m)}(\hat{\boldsymbol{Z}}^{(m)}) \right]$ does not depend on $\mathcal{P}$. In Lemma 4.A.1 we make use of the Markov property of Eq. 4.102 to decouple the maximization objective into three parts that each depend on different parameters and can be maximized separately.

Lemma 4.A.1 (Decoupled Objectives for Maximization Step)**.** *Denoting the means and joint covariances of the inference distributions $Q^{(m)}$, $m = 1, \ldots, M$ by*

$$\hat{\boldsymbol{\mu}}_k^{(m)} = \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}} \left[\hat{\boldsymbol{z}}_k^{(m)} \right] \qquad \textit{and} \qquad \hat{\boldsymbol{\Sigma}}_{k,l}^{(m)} = \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}} \left[\left(\hat{\boldsymbol{z}}_k^{(m)} - \hat{\boldsymbol{\mu}}_k^{(m)} \right) \left(\hat{\boldsymbol{z}}_l^{(m)} - \hat{\boldsymbol{\mu}}_l^{(m)} \right)^T \right], \tag{4.139}$$

then the first term in the ELBO Eq. 4.107 decouples into

$$\begin{aligned} &\sum_{m=1}^{M} \mathbb{E}_{\hat{\boldsymbol{Z}}^{(m)}} \left[\log P_{\boldsymbol{Z},\boldsymbol{Y}}(\hat{\boldsymbol{Z}}^{(m)}, \boldsymbol{Y}^{(m)};\ \mathcal{P}) \right] \\ &\qquad = -\frac{1}{2} \left[L_1(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) + L_2(\boldsymbol{c}_0, \boldsymbol{\Sigma}_{\boldsymbol{v}}) + L_3(\tilde{\boldsymbol{V}}_0, \ldots, \tilde{\boldsymbol{V}}_{\dim \boldsymbol{u}}, \boldsymbol{\Sigma}_{\boldsymbol{w}}) \right]. \end{aligned} \tag{4.140}$$

The first term

$$L_1(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) = M \log\det(2\pi\boldsymbol{\Sigma}_0) + \mathrm{Tr}\left\{ \boldsymbol{\Sigma}_0^{-1} \sum_{m=1}^{M} \left[\hat{\boldsymbol{\Sigma}}_{0,0}^{(m)} + \left(\hat{\boldsymbol{\mu}}_0^{(m)} - \boldsymbol{\mu}_0 \right) \left(\hat{\boldsymbol{\mu}}_0^{(m)} - \boldsymbol{\mu}_0 \right)^T \right] \right\}, \tag{4.141}$$

is a loss function for the initial condition parameters. The second term

$$L_2(\tilde{\boldsymbol{C}}, \boldsymbol{c}_0, \boldsymbol{\Sigma}_{\boldsymbol{v}}) = M(L+1)\log\det\left(2\pi\boldsymbol{\Sigma}_{\boldsymbol{v}}\right) + \operatorname{Tr}\left\{\boldsymbol{\Sigma}_{\boldsymbol{v}}^{-1}\sum_{m=1}^{M}\sum_{l=0}^{L}\left[\tilde{\boldsymbol{C}}\hat{\boldsymbol{\Sigma}}_{l,l}^{(m)}\tilde{\boldsymbol{C}}^T + \left(\boldsymbol{y}_l^{(m)} - \boldsymbol{c}_0 - \tilde{\boldsymbol{C}}\hat{\boldsymbol{\mu}}_l^{(m)}\right)\left(\boldsymbol{y}_l^{(m)} - \boldsymbol{c}_0 - \tilde{\boldsymbol{C}}\hat{\boldsymbol{\mu}}_l^{(m)}\right)^T\right]\right\}, \quad (4.142)$$

is a loss function for the observation map parameters. And the third term

$$\begin{aligned} L_3(\tilde{\boldsymbol{V}}_0, \ldots, \tilde{\boldsymbol{V}}_{\dim \boldsymbol{u}}, \boldsymbol{\Sigma}_{\boldsymbol{w}}) &= ML\log\det\left(2\pi\boldsymbol{\Sigma}_{\boldsymbol{w}}\right) \\ &+ \operatorname{Tr}\left\{\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\sum_{m=1}^{M}\sum_{l=0}^{L-1}\left[\hat{\boldsymbol{\Sigma}}_{l+1,l+1}^{(m)} - \boldsymbol{A}_l^{(m)}\hat{\boldsymbol{\Sigma}}_{l,l+1}^{(m)} - \hat{\boldsymbol{\Sigma}}_{l+1,l}^{(m)}(\boldsymbol{A}_l^{(m)})^T + \boldsymbol{A}_l^{(m)}\hat{\boldsymbol{\Sigma}}_{l,l}(\boldsymbol{A}_l^{(m)})^T\right.\right. \\ &\left.\left. + \left(\hat{\boldsymbol{\mu}}_{l+1}^{(m)} - \boldsymbol{A}_l^{(m)}\hat{\boldsymbol{\mu}}_l^{(m)} - \boldsymbol{b}_l^{(m)}\right)\left(\hat{\boldsymbol{\mu}}_{l+1}^{(m)} - \boldsymbol{A}_l^{(m)}\hat{\boldsymbol{\mu}}_l^{(m)} - \boldsymbol{b}_l^{(m)}\right)^T\right]\right\} \end{aligned} \quad (4.143)$$

is a loss function for the dynamical parameters including the matrix approximations of the Koopman generators.

Proof. We begin by considering a single trajectory with fixed m, removing it from the superscript, and take the sum over m at the end. To avoid cluttered equations for the time being, we also drop the explicit dependence of the probability distributions on the model parameters $\mathcal{P}$ and remove the subscripts from probability densities where the appropriate random variables are clear from context. If we let $\boldsymbol{H}_l = (\boldsymbol{z}_0, \boldsymbol{y}_0, \ldots, \boldsymbol{z}_l, \boldsymbol{y}_l)$, then by the Markov property we have

$$\begin{aligned} P(\boldsymbol{Z}, \boldsymbol{Y}) &= P(\boldsymbol{H}_L) \\ &= P(\boldsymbol{H}_{L-1}) \cdot P\left(\boldsymbol{z}_L, \boldsymbol{y}_L \mid \boldsymbol{H}_{L-1}\right) \\ &= P(\boldsymbol{H}_{L-1}) \cdot P\left(\boldsymbol{z}_L, \boldsymbol{y}_L \mid \boldsymbol{z}_{L-1}\right) \\ &\vdots \\ &= P\left(\boldsymbol{z}_0, \boldsymbol{y}_0\right) \cdot \prod_{l=0}^{L-1} P\left(\boldsymbol{z}_{l+1}, \boldsymbol{y}_{l+1} \mid \boldsymbol{z}_l\right). \end{aligned} \quad (4.144)$$

Making use of the conditional independence of the observation $\boldsymbol{y}_{l+1}$ and previous state $\boldsymbol{z}_l$ given $\boldsymbol{z}_{l+1}$, we obtain

$$P(\boldsymbol{Z}, \boldsymbol{Y}) = P\left(\boldsymbol{z}_0\right) \cdot \prod_{l=0}^{L} P\left(\boldsymbol{y}_l \mid \boldsymbol{z}_l\right) \cdot \prod_{l=0}^{L-1} P\left(\boldsymbol{z}_{l+1} \mid \boldsymbol{z}_l\right). \quad (4.145)$$

Recalling our dynamical model Eq. 4.102, the log joint probability is given by

$$\begin{aligned}\log P(\boldsymbol{Z},\boldsymbol{Y}) = & -\frac{1}{2}\log\det\left(2\pi\boldsymbol{\Sigma}_0\right)-\frac{1}{2}\left(\boldsymbol{z}_0-\boldsymbol{\mu}_0\right)^T\boldsymbol{\Sigma}_0^{-1}\left(\boldsymbol{z}_0-\boldsymbol{\mu}_0\right)\\ & -\frac{1}{2}(L+1)\log\det\left(2\pi\boldsymbol{\Sigma}_{\boldsymbol{v}}\right)-\frac{1}{2}\sum_{l=0}^{L}\left(\boldsymbol{y}_l-\boldsymbol{c}_0-\tilde{\boldsymbol{C}}\boldsymbol{z}_l\right)^T\boldsymbol{\Sigma}_{\boldsymbol{v}}^{-1}\left(\boldsymbol{y}_l-\boldsymbol{c}_0-\tilde{\boldsymbol{C}}\boldsymbol{z}_l\right)\\ & -\frac{1}{2}L\log\det\left(2\pi\boldsymbol{\Sigma}_{\boldsymbol{w}}\right)-\frac{1}{2}\sum_{l=0}^{L-1}\left(\boldsymbol{z}_{l+1}-\boldsymbol{A}_l\boldsymbol{z}_l-\boldsymbol{b}_l\right)^T\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\left(\boldsymbol{z}_{l+1}-\boldsymbol{A}_l\boldsymbol{z}_l-\boldsymbol{b}_l\right).\end{aligned} \tag{4.146}$$

Taking the expectation with respect to the inference distribution, we obtain

$$\mathbb{E}_{\hat{\boldsymbol{Z}}}\left[\log P_{\boldsymbol{Z},\boldsymbol{Y}}(\hat{\boldsymbol{Z}},\boldsymbol{Y};\ \mathcal{P})\right]=-\frac{1}{2}\left[\tilde{L}_1(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)+\tilde{L}_2(\tilde{\boldsymbol{C}},\boldsymbol{c}_0,\boldsymbol{\Sigma}_{\boldsymbol{v}})+\tilde{L}_3(\tilde{\boldsymbol{V}}_0,\ldots,\tilde{\boldsymbol{V}}_q,\boldsymbol{\Sigma}_{\boldsymbol{w}})\right], \tag{4.147}$$

where

$$\tilde{L}_1(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)=\log\det\left(2\pi\boldsymbol{\Sigma}_0\right)+\mathrm{Tr}\left[\boldsymbol{\Sigma}_0^{-1}\left(\hat{\boldsymbol{\Sigma}}_{0,0}+\left(\hat{\boldsymbol{\mu}}_0-\boldsymbol{\mu}_0\right)\left(\hat{\boldsymbol{\mu}}_0-\boldsymbol{\mu}_0\right)^T\right)\right] \tag{4.148}$$

$$\begin{aligned}\tilde{L}_2(\tilde{\boldsymbol{C}},\boldsymbol{c}_0,\boldsymbol{\Sigma}_{\boldsymbol{v}}) = & (L+1)\log\det\left(2\pi\boldsymbol{\Sigma}_{\boldsymbol{v}}\right)\\ & +\mathrm{Tr}\left\{\boldsymbol{\Sigma}_{\boldsymbol{v}}^{-1}\sum_{k=0}^{L}\left[\tilde{\boldsymbol{C}}\hat{\boldsymbol{\Sigma}}_{k,k}\tilde{\boldsymbol{C}}^T+\left(\boldsymbol{y}_k-\boldsymbol{c}_0-\tilde{\boldsymbol{C}}\hat{\boldsymbol{\mu}}_k\right)\left(\boldsymbol{y}_k-\boldsymbol{c}_0-\tilde{\boldsymbol{C}}\hat{\boldsymbol{\mu}}_k\right)^T\right]\right\}\end{aligned} \tag{4.149}$$

$$\begin{aligned}\tilde{L}_3(\tilde{\boldsymbol{V}}_0,\ldots,\tilde{\boldsymbol{V}}_q,\boldsymbol{\Sigma}_{\boldsymbol{w}}) = & L\log\det\left(2\pi\boldsymbol{\Sigma}_{\boldsymbol{w}}\right)+\mathrm{Tr}\left\{\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\sum_{k=0}^{L-1}\left[\hat{\boldsymbol{\Sigma}}_{k+1,k+1}-\boldsymbol{A}_k\hat{\boldsymbol{\Sigma}}_{k,k+1}\right.\right.\\ & \left.\left.-\hat{\boldsymbol{\Sigma}}_{k+1,k}\boldsymbol{A}_k^T+\boldsymbol{A}_k\hat{\boldsymbol{\Sigma}}_{k,k}\boldsymbol{A}_k^T+\left(\hat{\boldsymbol{\mu}}_{k+1}-\boldsymbol{A}_k\hat{\boldsymbol{\mu}}_k-\boldsymbol{b}_k\right)\left(\hat{\boldsymbol{\mu}}_{k+1}-\boldsymbol{A}_k\hat{\boldsymbol{\mu}}_k-\boldsymbol{b}_k\right)^T\right]\right\}\end{aligned} \tag{4.150}$$

The final result is obtained by summing over m. □

We observe that by Lemma 4.A.1, it suffices to minimize the three terms L_1, L_2, and L_3 separately. Each term has the same form,

$$L_i(\boldsymbol{\Sigma}_i,\mathcal{P}_i)=\alpha_i\log\det\left(2\pi\boldsymbol{\Sigma}_i\right)+\mathrm{Tr}\left[\boldsymbol{\Sigma}_i^{-1}\boldsymbol{W}_i(\mathcal{P}_i)\right], \tag{4.151}$$

where $\boldsymbol{\Sigma}_i$ is a covariance matrix to be determined, $\alpha_i>0$ is a constant, and $\boldsymbol{W}_i$ is a symmetric, positive semi-definite matrix-valued function of the remaining parameters $\mathcal{P}_i\subset\mathcal{P}\setminus\{\boldsymbol{\Sigma}_i\}$ to be

optimized. In fact, we know even more: our assumption that the log likelihood has an upper bound implies that $\boldsymbol{W}_i(\mathcal{P}_i)$ is positive definite for all possible values of the parameters $\mathcal{P}_i$. Suppose that $\boldsymbol{W}_i(\mathcal{P}_i)$ is singular, having eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n = 0$, and take $\boldsymbol{\Sigma}_i = \boldsymbol{W}_i(\mathcal{P}_i) + \varepsilon \boldsymbol{I}$ with $\varepsilon > 0$. Then we have

$$L_i(\boldsymbol{\Sigma}_i, \mathcal{P}_i) = \alpha \sum_{i=1}^{n} \log\left[2\pi(\lambda_i + \varepsilon)\right] + \sum_{i=1}^{n} \frac{\lambda_i}{\lambda_i + \varepsilon}, \tag{4.152}$$

which approaches $-\infty$ as $\varepsilon \to 0$, meaning that the evidence lower bound $\hat{L}_Q(\mathcal{P})$ in Eq. 4.107 approaches $+\infty$, thereby contradicting our assumption that the true log likelihood $L(\mathcal{P})$ is bounded from above.

The proof is completed by employing the following Lemma on each loss function separately.

Lemma 4.A.2. *Let $\alpha > 0$ and suppose that $\mathcal{P}' \mapsto \boldsymbol{W}(\mathcal{P}')$ is a positive-definite matrix-valued function of some parameters $\mathcal{P}'$. Any minimizer of*

$$L(\boldsymbol{\Sigma}, \mathcal{P}') = \alpha \log\det\left(2\pi\boldsymbol{\Sigma}\right) + \operatorname{Tr}\left[\boldsymbol{\Sigma}^{-1}\boldsymbol{W}(\mathcal{P}')\right] \tag{4.153}$$

satisfies $\boldsymbol{\Sigma} = \frac{1}{\alpha}\boldsymbol{W}(\mathcal{P}')$. The remaining variables $\mathcal{P}'$ minimize $\log\det \boldsymbol{W}(\mathcal{P}')$.

Proof. If $(\boldsymbol{\Sigma}, \mathcal{P}')$ is an extremum of L then the derivative of L with respect to $\boldsymbol{\Sigma}$ must vanish. That is, for any variation $\delta\boldsymbol{\Sigma}$ we have

$$\begin{aligned} 0 = \frac{\partial L}{\partial \boldsymbol{\Sigma}}\delta\boldsymbol{\Sigma} &= \alpha \operatorname{Tr}\left(\boldsymbol{\Sigma}^{-1}\delta\boldsymbol{\Sigma}\right) - \operatorname{Tr}\left[\boldsymbol{\Sigma}^{-1}(\delta\boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1}\boldsymbol{W}(\mathcal{P}')\right] \\ &= \operatorname{Tr}\left\{\left[\alpha\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{W}(\mathcal{P}')\boldsymbol{\Sigma}^{-1}\right]\delta\boldsymbol{\Sigma}\right\} \end{aligned} \tag{4.154}$$

Since this holds for any variation, we must have

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\alpha}\boldsymbol{\Sigma}^{-1}\boldsymbol{W}(\mathcal{P}')\boldsymbol{\Sigma}^{-1}, \tag{4.155}$$

for otherwise we could choose $\delta\boldsymbol{\Sigma} = \left[\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{W}(\mathcal{P}')\boldsymbol{\Sigma}^{-1}\right]^T$ and produce a contradiction. Multiplying both sides by $\boldsymbol{\Sigma}$ proves the first claim. Substituting $\boldsymbol{\Sigma} = \frac{1}{\alpha}\boldsymbol{W}(\mathcal{P}')$ into L shows that $\mathcal{P}'$ must minimize

$$L\left(\frac{1}{\alpha}\boldsymbol{W}(\mathcal{P}'), \mathcal{P}'\right) = \alpha \log\det \boldsymbol{W}(\mathcal{P}') + \alpha \dim\left(\boldsymbol{\Sigma}\right)\log\left(\frac{2\pi}{\alpha}\right) + \alpha \dim\left(\boldsymbol{\Sigma}\right). \tag{4.156}$$

Since the last two terms and factor α on the first term are constants, this is equivalent to minimizing $\log\det \boldsymbol{W}(\mathcal{P}')$. □

□

Chapter 5

Measurement selection for states in nonlinear sets

Measurement selection, sparse sensor placement, and feature selection are closely related problems that have a wide range of applications in engineering, design of experiments, and modeling complex systems. In this chapter, we explore these applications and develop a geometric viewpoint that leads to new methods for problems with challenging nonlinearities. The problems we consider here involve choosing from a pre-defined discrete set of available measurements, sensors, or features, the ones that allow us to best predict or reconstruct some quantities of interest. Unlike may of the optimization problems we have encountered in previous chapters, these problems are inherently discrete and combinatorial in nature. Since the number of possible sensor combinations in problems of interest is enormous, algorithms that explore all possibilities are impractical. Therefore, a key challenge is to formulate measurement selection problems in ways that admit high quality approximate solutions by efficient algorithms that explore only a tiny subset of all possible combinations. The predominant viewpoint for measuring the quality of the selected measurements is statistical. However, the simplifying assumptions needed to arrive at practical algorithms often result in poor performance on problems possessing a high degree of nonlinearity. In this chapter, we depart somewhat from the statistical viewpoint and develop a complementary geometric viewpoint that allows us to grapple with nonlinearity in its natural habitat. We hope to reunite these viewpoints in future work.

We focus on a general class of problems described in [195]** that we shall refer to broadly as "measurement selection" problems. However, in principal, this class encompass many types of problems pertaining to sensor placement, feature selection, design of experiments, etc.. This

viewpoint can probably be generalized further, but we present a version that applies in most practical settings. In [195]**, we suppose that there is an underlying state space $\mathcal{X}$ and a vector-valued function $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^q$ containing quantities of interest. We do not get to measure the state $x \in \mathcal{X}$ or the quantities of interest $\boldsymbol{g}(x)$ directly. Instead, there is a set of costly measurements $\mathcal{M} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_M\}$ where each $\boldsymbol{m}_j : \mathcal{X} \to \mathbb{R}^{d_j}$ is a vector-valued function. These allowable measurements are usually determined by engineering constraints such as feasible locations for sensors and the physical quantities that sensors can measure. In many cases it is impossible to directly measure the quantities of interest using sensors. Our goal is to choose a subset $\mathcal{S} = \{\boldsymbol{m}_{j_1}, \ldots, \boldsymbol{m}_{j_K}\} \subset \mathcal{M}$ so that the values we measure

$$\boldsymbol{m}_{\mathcal{S}}(x) = (\boldsymbol{m}_{j_1}(x), \ldots, \boldsymbol{m}_{j_K}(x)) \in \mathbb{R}^{d_{\mathcal{S}}} \tag{5.1}$$

allow us to accurately infer the quantities of interest $\boldsymbol{g}(x)$. In particular, we want to ensure that there is a function $\boldsymbol{\Phi}_{\mathcal{S}} : \mathbb{R}^{d_{\mathcal{S}}} \to \mathbb{R}^q$ such that

$$\boldsymbol{g}(x) = \boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{m}_{\mathcal{S}}(x)) \qquad \text{for every } x \in \mathcal{X}. \tag{5.2}$$

The quality of the sensors $\mathcal{S}$ is measured by whether such a function $\boldsymbol{\Phi}_{\mathcal{S}}$ exists and how sensitive the function is to measurement errors, noise, or disturbances. In other words, the value of $\boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{m}_{\mathcal{S}}(x))$ should not change drastically when the measurements $\boldsymbol{m}_{\mathcal{S}}(x)$ are perturbed.

In general, measurement selection problems of the kind described above are combinatorial in nature, making it impractical to evaluate the performance of all $\binom{M}{K}$ choices of K measurement functions for many problems of interest. Therefore, efficient algorithms that approximate an optimal solution are employed. *A main goal of this chapter is to formulate various objectives for measurement selection that admit efficient approximate optimization algorithms.* For instance, a common approach (e.g., [125]) is to relax the original performance criterion described over discrete collections of measurements into a convex objective defined on a continuous space. Inclusion in the set $\mathcal{S}$ may be indicated using a binary vector $\boldsymbol{s} \in \{0, 1\}^M$ that is subsequently relaxed to take continuous values in the unit cube $[0, 1]^M$. The resulting convex optimization problem can then be solved using various standard methods [29]. To recover a discrete approximation, thresholds or randomized rounding can be applied to the real-valued $\boldsymbol{s}$. A drawback of convex optimization approaches is that they can become computationally expensive as the number of sensors and optimization constraints grows.

Another common approach we employ is to select the measurements sequentially using efficient

greedy algorithms. One advantage is that these "greedy" selection methods tend to have better scaling with problem size than convex optimization. In the simplest greedy algorithm, one starts off with no measurements, and at each step of the greedy algorithm adds the measurement to the collection $\mathcal{S}$ that produces the greatest increase in the performance objective. In general, this myopic approach is incapable of identifying collections of sensors that achieve superior combined performance compared to the sum of each sensor's individual performance. Remarkably, however, greedy algorithms have near-optimal performance when the objective function being maximized has a diminishing returns property called "submodularity", defined below:

Definition 5.0.1 (Submodular function)**.** *Let $2^{\mathcal{M}}$ denote all subsets of the finite set $\mathcal{M}$. A function $f : 2^{\mathcal{M}} \to \mathbb{R}$ is called submodular if given any element $j \in \mathcal{M}$ and subsets $\mathcal{S} \subset \mathcal{S}' \subset \mathcal{M} \setminus \{j\}$, the function f increases when j is added to the smaller subset $\mathcal{S}$ by at least as much as f increases when j is added to the larger subset $\mathcal{S}'$, i.e.,*

$$\mathcal{S} \subset \mathcal{S}' \subset \mathcal{M} \setminus \{j\} \qquad \Rightarrow \qquad f(\mathcal{S} \cup \{j\}) - f(\mathcal{S}) \geq f(\mathcal{S}' \cup \{j\}) - f(\mathcal{S}'). \tag{5.3}$$

When the optimization objective is submodular, monotone non-decreasing, and normalized, a result by G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher [187] (Theorem 5.0.2) says that the greedy algorithm achieves a value of the objective function within a constant factor $(1 - 1/e)$ of the optimal value. The proof of this result is extremely elegant, and so we provide it here as well.

Theorem 5.0.2 (G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, 1978 [187])**.** *Let $f : 2^{\mathcal{M}} \to \mathbb{R}$ be a submodular function that is monotone non-decreasing,*

$$\mathcal{S} \subset \mathcal{S}' \subset \mathcal{M} \qquad \Rightarrow \qquad f(\mathcal{S}) \leq f(\mathcal{S}'), \tag{5.4}$$

and normalized such that $f(\emptyset) = 0$. Let $\mathcal{S}_K^* \subset \mathcal{M}$ be a solution of the optimization problem

$$\underset{\mathcal{S} \subset \mathcal{M}}{\text{maximize}} \;\; f(\mathcal{S}) \qquad s.t. \qquad |\mathcal{S}| \leq K. \tag{5.5}$$

If a collection of subsets $\emptyset = \mathcal{S}_0, \ldots, \mathcal{S}_M = \mathcal{M}$ with $\mathcal{S}_k = \mathcal{S}_{k-1} \cup \{j_k\}$ satisfies the greedy maximization property

$$f(\mathcal{S}_{k-1} \cup \{j_k\}) - f(\mathcal{S}_{k-1}) \geq f(\mathcal{S}_{k-1} \cup \{j\}) - f(\mathcal{S}_{k-1}) \qquad \forall j \in \mathcal{M} \setminus \mathcal{S}_{k-1} \tag{5.6}$$

for each $k = 1, \ldots, M$*, then*

$$\boxed{f(\mathcal{S}_k) \geq \left(1 - e^{-k/K}\right) f(\mathcal{S}_K^*) \qquad \forall k = 1, \ldots, M.} \tag{5.7}$$

Proof of Theorem 5.0.2 (G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, 1978 [187]). The result follows almost immediately from the sequence of inequalities

$$\begin{aligned} f(\mathcal{S}_K^*) - f(\mathcal{S}_{k-1}) &\leq f(\mathcal{S}_K^* \cup \mathcal{S}_{k-1}) - f(\mathcal{S}_{k-1}) \\ &\leq \sum_{j \in \mathcal{S}_K^* \setminus \mathcal{S}_{k-1}} \left(f(\mathcal{S}_{k-1} \cup \{j\}) - f(\mathcal{S}_{k-1})\right. \\ &\leq K \left(f(\mathcal{S}_k) - f(\mathcal{S}_{k-1})\right), \end{aligned} \tag{5.8}$$

where the first inequality follows from monotonicity, the second from submodularity, and the third from the definition of the greedy algorithm. Rearranging this inequality, we find

$$f(\mathcal{S}_K^*) - f(\mathcal{S}_k) \leq \left(1 - \frac{1}{K}\right) \left(f(\mathcal{S}_K^*) - f(\mathcal{S}_{k-1})\right), \tag{5.9}$$

which, when applied inductively, yields the desired result

$$f(\mathcal{S}_K^*) - f(\mathcal{S}_k) \leq \left(1 - \frac{1}{K}\right)^k \left(f(\mathcal{S}_K^*) - f(\mathcal{S}_0)\right) = \left(1 - \frac{1}{K}\right)^k f(\mathcal{S}_K^*) \leq e^{-k/K} f(\mathcal{S}_K^*), \tag{5.10}$$

where second inequality follows from convexity of the exponential $e^{-1/K} \geq 1 - 1/K$. □

Classical results by L. A. Wolsey [283] show that greedy algorithms may also be used to achieve near-optimal performance on an important class of problems for sensor placement called submodular "set-covering" problems. In a submodular set-covering problem we seek to choose a subset $\mathcal{S} \subset \mathcal{M}$ of minimum size, or with the lowest cost, that achieves a certain level of performance described by $f(\mathcal{S}) = f(\mathcal{M})$ for a normalized, non-decreasing, submodular function f. Here, the condition $f(\mathcal{S}) = f(\mathcal{M})$ may be describing some minimal desired level of performance for the sensors, such as the existence of a reconstruction function $\mathbf{\Phi}_{\mathcal{S}}$ with desirable properties as in [195]**.

Recently, it has been shown by A. A. Bian et al. [24], that greedy algorithms may also be applied to non-submodular objective functions with guaranteed performance depending on additional parameters quantifying how much the objective departs from submodularity. This is important because many sensor placement and measurement selection objectives of interest are not submodular; the average square error of the optimal linear estimator is one such non-submodular objective [71].

5.1 Applications of measurement selection problems

5.1.1 Sensor placement in dynamical systems

The measurement selection framework described above includes sensor selection problems for dynamical systems in which we measure finite time histories from each sensor. To illustrate this, we first consider a discrete-time linear system $\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t$ where our goal is to choose a subset of available observation matrices $\{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_M\}$ in order to estimate the initial condition $\boldsymbol{x}_0$ from a finite time-history of L measurements. The measurements made by the jth observation matrix from the initial condition $\boldsymbol{x}$ are given by

$$\boldsymbol{m}_j(\boldsymbol{x}) = \left(\boldsymbol{C}_j\boldsymbol{x}, \boldsymbol{C}_j\boldsymbol{A}\boldsymbol{x}, \ldots, \boldsymbol{C}_j\boldsymbol{A}^{L-1}\boldsymbol{x}\right) = \boldsymbol{O}_j\boldsymbol{x}, \tag{5.11}$$

and the measurements made by a collection of observation matrices $\boldsymbol{C}_{\mathcal{S}}^T = \begin{bmatrix}\boldsymbol{C}_{j_1}^T & \cdots & \boldsymbol{C}_{j_K}^T\end{bmatrix}$ are given, up to row permutation, by $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) = \boldsymbol{O}_{\mathcal{S}}\boldsymbol{x}$. When it exists, the reconstruction function for the initial condition $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$ using a given collection of observations $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})$ is given by

$$\boldsymbol{x} = \boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{m}_{\mathcal{S}}(x)) = \boldsymbol{W}_{\mathcal{S}}^{-1}\boldsymbol{O}_{\mathcal{S}}^T\boldsymbol{m}_{\mathcal{S}}(x), \tag{5.12}$$

where

$$\boldsymbol{W}_{\mathcal{S}} = \boldsymbol{O}_{\mathcal{S}}^T\boldsymbol{O}_{\mathcal{S}}^T = \sum_{t=0}^{L-1}(\boldsymbol{A}^t)^T\boldsymbol{C}_{\mathcal{S}}^T\boldsymbol{C}_{\mathcal{S}}\boldsymbol{A}^t = \sum_{j\in\mathcal{S}}\boldsymbol{W}_j \tag{5.13}$$

is the time-L observability Gramian. Sensor (and actuator) placement methods for linear systems based on the observability (and controllability) Gramian have been discussed by T. H. Summers et al. in [252] and [253].

On the other hand, our framework also allows us to describe observability for nonlinear systems based on analogous time-delayed measurement sequences. As an illustration, we consider nonlinear discrete-time dynamics $x_{t+1} = F(x_t)$ and choose from among a set of nonlinear state observation functions $\mathcal{M} = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_M\}$. In this case, the time histories of measurements from an initial condition $x \in \mathcal{X}$ are given by

$$\boldsymbol{m}_j(x) = \left(\boldsymbol{h}_j(x), \boldsymbol{h}_j(F(x)), \ldots, \boldsymbol{h}_j(F^{L-1}(x))\right) \tag{5.14}$$

and we select a set of state observations $\mathcal{S} \subset \mathcal{M}$ such that relevant information about the initial condition $\boldsymbol{g}(x)$ can be reconstructed from $\boldsymbol{m}_{\mathcal{S}}(x)$. Finally, state estimation problems for nonlin-

ear systems $x_{t+1} = F(x_t, \boldsymbol{u}_t)$ with inputs $\boldsymbol{u}_t \in \mathcal{U} \subset \mathbb{R}^m$ may be described in our framework by constructing measurements on an augmented state space $\mathcal{X} \times \mathcal{U}^{L-1}$ given by

$$\boldsymbol{m}_{\mathbb{S}}(x, \boldsymbol{u}_0, \ldots, \boldsymbol{u}_{L-2}) = (\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{L-2}, \boldsymbol{h}_{\mathbb{S}}(x), \boldsymbol{h}_{\mathbb{S}}(F(x, \boldsymbol{u}_0)), \ldots, \boldsymbol{h}_{\mathbb{S}}(F(\ldots F(x, \boldsymbol{u}_0), \ldots \boldsymbol{u}_{L-2}))) \,. \tag{5.15}$$

That is, we measure both the time histories of the input and the selected output observables.

5.1.2 Reduced-order modeling

Measurement selection techniques can be used to achieve computationally efficient dimensionality reduction for discretized fluid flows and other spatio-temporal partial differential equations (PDEs) exhibiting coherent structures. In this setting, these techniques select a small number of spatial locations in the physical domain at which to measure the state variables or their time derivatives. These measurements are then used to reconstruct the relevant quantities needed to evolve the system in time.

Dimensionality reduction based on sampled state variables has advantages over other more general nonlinear embeddings described in Section 3.2. First, the sampled state variables are easy to interpret. In contrast, it may be difficult to determine exactly what a coordinate in the latent space of an autoencoder (see Section 3.2) corresponds to physically; even linear combinations of state variables can be difficult to interpret, especially if they include multiple physical quantities like non-dimensional velocities, pressures, and/or reacting chemical species.

The second advantage comes from the local structure of discretized spatio-temporal PDEs. The time derivatives of state variables at the sampled locations depend only on the values of state variables in a small neighborhood of each sample point. This means that the time derivatives of the sampled state variables can be efficiently computed by reconstructing the other states only on these small patches. In contrast, if the embedding depends on state variables at every physical location, e.g., by projecting onto POD modes or via a nonlinear encoder, then it becomes necessary to evaluate the time derivative of the full-order model over the entire spatial domain. Even though such a reduced-order model may have a small number of state variables, it will still be computationally expensive to simulate because it requires evaluating the entire time derivative of the full-order model.

Recognizing these computational difficulties, S. Chaturantabut and D. C. Sorensen [60] propose a Discrete-Empirical Interpolation Method (DEIM) in which the system's time derivative is reconstructed from its values at a small number of carefully chosen physical locations. In particular, they consider a POD-Galerkin reduced-order model (see Section 3.1.1) where the POD coefficients $\boldsymbol{z} \in \mathbb{R}^r$

evolve according to

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\boldsymbol{z} = \boldsymbol{U}^T \boldsymbol{f}(\boldsymbol{U}\boldsymbol{z}). \tag{5.16}$$

In general, it is impossible to evolve $\boldsymbol{z}$ using such a model without computing the time derivative of the high-dimensional full-order model $\boldsymbol{f}(\boldsymbol{U}\boldsymbol{z}) \in \mathbb{R}^n$ at each time step. Instead, DEIM constructs a second $\tilde{r}$-dimensional POD basis $\tilde{\boldsymbol{U}}$ for $\boldsymbol{f}$ and a sparse sampling matrix $\boldsymbol{S} \in \{0,1\}^{\tilde{r}\times n}$ such that $\boldsymbol{S}\tilde{\boldsymbol{U}}$ is invertible and well-conditioned. Consequently, an approximation for $\boldsymbol{f}(\boldsymbol{U}\boldsymbol{z})$ can be constructed in the POD subspace Range($\tilde{\boldsymbol{U}}$) from sparse samples of $\boldsymbol{f}(\boldsymbol{U}\boldsymbol{z})$ according to

$$\boldsymbol{f}(\boldsymbol{U}\boldsymbol{z}) \approx \tilde{\boldsymbol{U}}(\boldsymbol{S}\tilde{\boldsymbol{U}})^{-1}\boldsymbol{S}\boldsymbol{f}(\boldsymbol{U}\boldsymbol{z}). \tag{5.17}$$

The resulting reduced-order model becomes

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\boldsymbol{z} = \underbrace{\boldsymbol{U}^T\tilde{\boldsymbol{U}}(\boldsymbol{S}\tilde{\boldsymbol{U}})^{-1}}_{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{f}(\boldsymbol{U}\boldsymbol{z}), \tag{5.18}$$

where the matrix $\boldsymbol{T}$ can be pre-computed offline. The key observation is that $\boldsymbol{S}\boldsymbol{f}$ can be efficiently evaluated if the support of each row of $\boldsymbol{S}$ contains state variables at a small number of spatial locations in the physical domain, requiring us to access only the elements of $\boldsymbol{U}\boldsymbol{z}$ in small neighborhoods of these points to compute finite-differences.

Remark 5.1.1 (DEIM for autoencoder-based ROMs)**.** *It should also be possible to use DEIM for model reduction using nonlinear embeddings provided by autoencoders of the form*

$$\boldsymbol{\psi}_e(\boldsymbol{x}) = \tilde{\boldsymbol{\psi}}_e(\boldsymbol{\Psi}^T\boldsymbol{x}) \qquad \textit{and} \qquad \boldsymbol{\psi}_d(\boldsymbol{z}) = \boldsymbol{\Phi}\tilde{\boldsymbol{\psi}}_d(\boldsymbol{z}), \tag{5.19}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{n\times r_e}$ and $\boldsymbol{\Phi} \in \mathbb{R}^{n\times r_d}$ with $r_e, r_d \ll n$. In this case, the latent state evolves according to

$$\frac{d}{d\,\mathrm{t}}\,\boldsymbol{z} = \mathrm{D}\,\tilde{\boldsymbol{\psi}}_e(\boldsymbol{\Psi}^T\boldsymbol{\Phi}\tilde{\boldsymbol{\psi}}_d(\boldsymbol{z}))\underbrace{\boldsymbol{\Psi}^T\tilde{\boldsymbol{U}}(\boldsymbol{S}\tilde{\boldsymbol{U}})^{-1}}_{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{f}(\boldsymbol{\Phi}\tilde{\boldsymbol{\psi}}_{\boldsymbol{d}}(\boldsymbol{z})), \tag{5.20}$$

where the matrices $\boldsymbol{T}$ and $\boldsymbol{\Psi}^T\boldsymbol{\Phi}$ can be pre-computed during an offline stage. One could also optimize the weights in the autoencoder with a sparsity-promoting penalty on the rows of $\boldsymbol{\Psi}$, which would eliminate the need for DEIM-based reconstruction of $\boldsymbol{f}$.

Modern techniques for constructing the sampling matrix $\boldsymbol{S}$ use pivoted QR factorization [82] or strong rank-revealing QR factorization [83] and provide guarantees on the accuracy of the DEIM

approximation in Eq. 5.17. In particular, one computes a pivoted QR factorization

$$\tilde{\boldsymbol{U}}^T \boldsymbol{P} = \boldsymbol{Q}\boldsymbol{R} \tag{5.21}$$

and defines $\boldsymbol{S}^T$ to be the first $\tilde{r}$ columns of the permutation matrix $\boldsymbol{P}$. The corresponding columns of $\boldsymbol{R}$ form a square upper-triangular matrix $\boldsymbol{R}_1$ whose diagonal entries are referred to as "pivots".

DEIM can be viewed as a measurement selection problem. In the most straightforward point of view, our states are the values of $\boldsymbol{f}$ and we are trying to reconstruct $\boldsymbol{g}(\boldsymbol{f}) = \boldsymbol{f}$ by selecting from available measurements $\boldsymbol{m}_j(\boldsymbol{f}) = \boldsymbol{e}_j^T \boldsymbol{f}$, where $\boldsymbol{e}_j$ is the jth column of the $n \times n$ identity matrix. However, even if we are only interested in the first $\tilde{r}$ POD coefficients $\boldsymbol{g}(\boldsymbol{f}) = \tilde{\boldsymbol{U}}^T \boldsymbol{f}$, exact reconstruction will generally be impossible using a function $\boldsymbol{\Phi}_{\mathbb{S}}$ of a proper subset of measurements $\boldsymbol{m}_{\mathbb{S}}$. Hence, we seek to select sensors to minimize the degree to which $\boldsymbol{g}(\boldsymbol{f})$ fails to be a function of $\boldsymbol{m}_{\mathbb{S}}$. Alternatively, the measurement-selection problem can be formulated using states states $\tilde{\boldsymbol{f}}$ lying in the $\tilde{r}$-dimensional POD subspace Range($\tilde{\boldsymbol{U}}$). Here, we select measurement functions $\boldsymbol{m}_j(\tilde{\boldsymbol{f}}) = \boldsymbol{e}_j \tilde{\boldsymbol{f}}$ in order to reconstruct $\boldsymbol{g}(\tilde{\boldsymbol{f}}) = \tilde{\boldsymbol{f}}$. The difference is that in this case, we may actually reconstruct the desired variables

$$\tilde{\boldsymbol{f}} = \underbrace{\tilde{\boldsymbol{U}}(\boldsymbol{S}\tilde{\boldsymbol{U}})^+}_{\boldsymbol{\Phi}_{\mathbb{S}}} \underbrace{\boldsymbol{S}\tilde{\boldsymbol{f}}}_{\boldsymbol{m}_{\mathbb{S}}(\tilde{\boldsymbol{f}})}, \qquad \boldsymbol{S}^T = \begin{bmatrix} \boldsymbol{e}_{j_1} & \cdots & \boldsymbol{e}_{j_K} \end{bmatrix} \tag{5.22}$$

for any $\tilde{\boldsymbol{f}} \in$ Range($\tilde{\boldsymbol{U}}$) as long as $\boldsymbol{S}\tilde{\boldsymbol{U}}$ is injective. In practice, the actual values of the full-order model's time derivatives $\boldsymbol{f}(\boldsymbol{U}\boldsymbol{z})$ do not lie perfectly in the POD subspace Range($\tilde{\boldsymbol{U}}$). These differences may be treated as measurement noise within our framework. So, when $\boldsymbol{f} = \tilde{\boldsymbol{f}} + (\boldsymbol{I} - \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^T)\boldsymbol{f}$ is the decomposition of $\boldsymbol{f}$ into its components in Range($\tilde{\boldsymbol{U}}$) and Range($\tilde{\boldsymbol{U}}$)$^\perp$, then we actually measure

$$\boldsymbol{S}\boldsymbol{f} = \boldsymbol{m}_{\mathbb{S}}(\tilde{\boldsymbol{f}}) + \underbrace{\boldsymbol{S}(\boldsymbol{I} - \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^T)\boldsymbol{f}}_{\text{noise}}. \tag{5.23}$$

The degree to which this noise causes errors in the reconstruction $\boldsymbol{\Phi}_{\mathbb{S}}(\boldsymbol{S}\boldsymbol{f})$ is determined by how much $\boldsymbol{\Phi}_{\mathbb{S}}$ amplifies the noise. In the setting of DEIM using pivoted QR factorization, we have

$$\boldsymbol{\Phi}_{\mathbb{S}} = \tilde{\boldsymbol{U}}\boldsymbol{R}_1^{-T}\boldsymbol{Q}^T\boldsymbol{S}, \tag{5.24}$$

where the maximum amplification of $\boldsymbol{\Phi}_{\mathbb{S}}$ is determined by the smallest singular value of $\boldsymbol{R}_1$ according to $\sigma_{\min}(\boldsymbol{R}_1)^{-1}$. Because $\tilde{\boldsymbol{U}}$ is an isometry, it follows that $\det(\boldsymbol{R}_1) = \prod_{i=1}^{\tilde{r}} [\boldsymbol{R}_1]_{i,i}$, which is greedily maximized during QR pivoting, is a lower bound for $\sigma_{\min}(\boldsymbol{R}_1)$ (for proof, see [193]*).

Sparse measurement selection may also be used directly to provide an embedding of the state for reduced-order modeling in a similar way to the autoencoders discussed in Section 3.2. In particular, the selected measurements can serve as an encoder $\boldsymbol{\psi}_e(\boldsymbol{x}) = \boldsymbol{m}_S(\boldsymbol{x})$ with the decoder provided by the reconstruction function $\boldsymbol{\Phi}_S$ for the full-state observable $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$ lying on or near an underlying low-dimensional manifold. It may also be possible to select measurements capturing dynamically significant low-energy features of the dynamics $\boldsymbol{x}_{t+1} = \boldsymbol{F}(\boldsymbol{x}_t)$ (see Section 3.3) by seeking to reconstruct a sequence of future states,

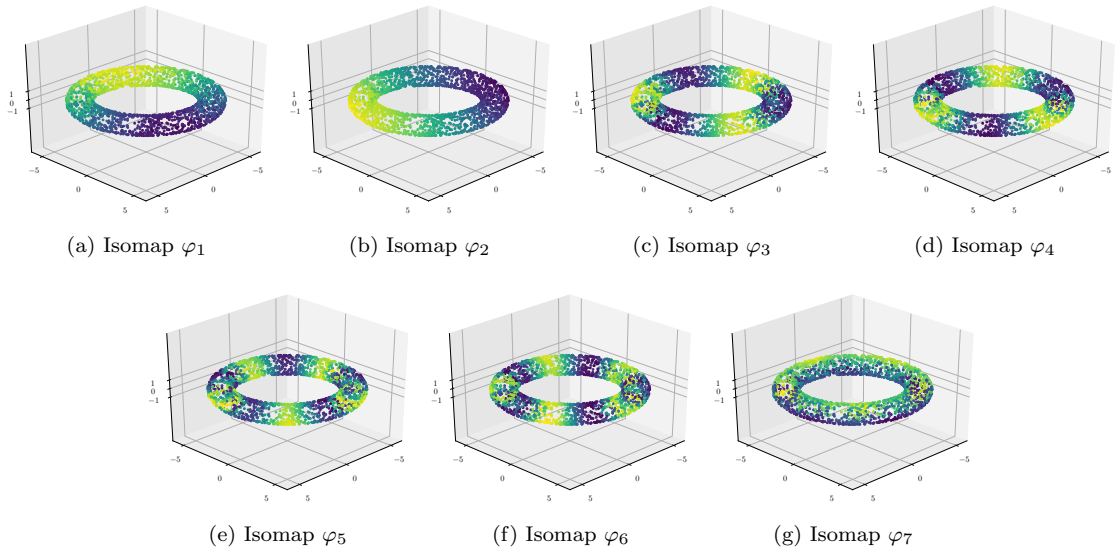
$$\boldsymbol{g}(\boldsymbol{x}) = (\boldsymbol{x}, \boldsymbol{F}(\boldsymbol{x}), \ldots, \boldsymbol{F}^{L-1}(\boldsymbol{x})). \tag{5.25}$$

Here, we want to select measurement functions $\boldsymbol{m}_S$ whose values at any two states $\boldsymbol{x}$, $\boldsymbol{x}'$ differ in proportion to the difference between trajectories $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x}')$. This is the same as minimizing the amplification (Lipschitz constant) of the reconstruction map $\boldsymbol{\Phi}_S$.

5.1.3 Selecting fundamental eigenfunctions

In Section 4.2 we discussed how to extract coherent observables for dynamical systems using eigenfunctions or approximate eigenfunctions of the Koopman operator. For instance, approximate eigenfunctions of the Koopman operator can be computed from data using Extended Dynamic Mode Decomposition (EDMD) [280] (see Section 4.3.1). Similarly, the eigenfunctions of operators defined on graphs and manifolds, often arising from a data set with known pair-wise similarity, may be used to construct embeddings in Euclidean space that capture salient features of the geometry. For instance, the diffusion maps algorithm [68] provides an embedding for a data set that captures its multi-scale structure based on the eigenfunctions of a diffusion operator defined on a graph. Other related techniques include Laplacian eigenmaps [15], Isomap [260], and kernel PCA [244].

The techniques mentioned above all produce a large collection of eigenfunctions, from which we are usually only interested in a small subset. Fundamental eigenfunctions are a minimal set from which all other eigenfunctions can be reconstructed. For spectral embedding techniques like diffusion maps, a set of fundamental eigenfunctions provides an embedding of the underlying data set in Euclidean space. Since the remaining eigenfunctions can be reconstructed as functions of the fundamental ones, these extra eigenfunctions are not needed for the embedding. For instance, in Figure 5.1.1 (appearing in [195]**) we show the leading seven Isomaps eigenfunctions computed from data lying on a torus. Because of the torus's rotational symmetry, several eigenfunctions $\varphi_3, \ldots, \varphi_6$ are harmonics of the leading two and provide redundant information. The subset $\{\varphi_1, \varphi_2, \varphi_7\}$ is fundamental and provides an embedding of the torus in $\mathbb{R}^3$. Likewise, Koopman eigenfunctions



(a) Isomap φ_1 (b) Isomap φ_2 (c) Isomap φ_3 (d) Isomap φ_4

(e) Isomap φ_5 (f) Isomap φ_6 (g) Isomap φ_7

Figure 5.1.1: Isomap coordinates [260] computed using 2000 randomly sampled points on a torus, appearing in [195]**. The leading six eigenfunctions resemble the real and imaginary components of $e^{ik\theta_1}$, $k = 1, 2, 3$, due to the rotational symmetry, providing redundant information about θ_1 and no information about θ_2. The eigenfunctions φ_1, φ_2, and φ_7 form a fundamental set. They provide an embedding of the data that captures its toroidal structure.

form a spectral lattice where products of eigenfunctions are also eigenfunctions (see Section 4.2.2) that carry no additional information about the dynamics. Consequently, we are interested in a minimal set of fundamental Koopman eigenfunctions, or approximate eigenfunctions, that carry the same information as the full set.

Selecting a set of fundamental eigenfunctions can be cast in the measurement selection framework. In particular, the state space $\mathcal{X}$ is the underlying manifold or graph, each measurement is an eigenfunction $m_j = \varphi_j$ and the relevant information to be reconstructed is the vector of all the eigenfunctions $\boldsymbol{g} = (\varphi_1, \ldots, \varphi_N)$. A fundamental set of eigenfunctions is a subset $\mathcal{S} = \{\varphi_{j_1}, \ldots, \varphi_{j_K}\}$ of minimal size such that $\boldsymbol{g}$ can be reconstructed as a function of $\boldsymbol{m}_\mathcal{S}$: $\boldsymbol{g} = \boldsymbol{\Phi}_\mathcal{S} \circ \boldsymbol{m}_\mathcal{S}$. In this framework, we may also consider a relaxed notion of a fundamental set by choosing a small collection of eigenfunctions that includes a fundamental set together with additional eigenfunctions that allow reconstruction by a map $\boldsymbol{\Phi}_\mathcal{S}$ with less sensitivity.

5.1.4 Feature selection in machine learning

In statistics and machine learning, feature selection refers to choosing a subset of random variables that allow for construction of an accurate statistical model. We consider the case where the model is used to predict another random variable of interest. For more details, one can consult the review

articles [76, 107] or the book by T. Hastie, R. Tibshirani, and J. Friedman [110]. In the statistical setting, the underlying state space $\mathcal{X}$ is a probability space with the available features $\boldsymbol{m}_j$ and relevant quantities $\boldsymbol{g}$ being random vectors, i.e., measurable functions on this probability space. Feature selection techniques choose a subset of features $\boldsymbol{m}_{\mathcal{S}}$ that allows adequate predictions of the value of the $\boldsymbol{g}$ using a function $\boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{m}_{\mathcal{S}})$. Most approaches to feature selection optimize $\boldsymbol{\Phi}_{\mathcal{S}}$ over a particular class of functions $\mathcal{F}$ and measure the performance of the selected features $\mathcal{S}$ according to the best reconstruction performance over $\boldsymbol{\Phi}_{\mathcal{S}} \in \mathcal{F}$. For instance, $\mathcal{F}$ may be the parametric space of linear functions over which we minimize the average square error of the reconstruction. Usually, one only has access to samples from the underlying probability space. In this, the performance of the sensors is measured with respect to the best performing reconstruction function $\boldsymbol{\Phi}_{\mathcal{S}} \in \mathcal{F}$ on the data.

5.1.5 Parameter estimation via design of experiments

Suppose that there is a physical process with an outcome depending on a set of parameters that we would like to estimate. The outcome of the jth experiment is a collection of measurements that can be viewed as a function $\boldsymbol{m}_j(x)$ of an underlying state $x = (\boldsymbol{p}, z)$ of the system that includes the parameters $\boldsymbol{p} = \boldsymbol{g}(x)$ that we would like to determine as well as other variables z like noise that affect the outcome. We may treat the experimental outcome $\boldsymbol{m}_j(\boldsymbol{p}, z)$ as a perturbation around a nominal outcome $\boldsymbol{m}_j(\boldsymbol{p}, z_0)$. We can then choose to perform a small collection $\mathcal{S}$ of experiments so that $\boldsymbol{p}$ can be reconstructed from their nominal outcomes $\boldsymbol{m}_j(\boldsymbol{p}, z_0)$ by a function $\boldsymbol{\Phi}_{\mathcal{S}}$ with minimal sensitivity to the perturbations about the nominal outcomes. The main difference between this setting and the ordinary measurement selection problem, is that the same experiment can be performed multiple times, i.e., $\mathcal{S}$ must be allowed to contain repeated elements.

5.2 Inadequacy of linear techniques

The majority of available techniques require the reconstruction map $\boldsymbol{\Phi}_{\mathcal{S}}$ to be linear, and the sensors are chosen to optimize various measures of performance associated with the linear reconstruction. However, as we point out in [195]**, requiring linear reconstruction can lead to an excessive number of measurements being selected, even when the desired information $\boldsymbol{g}(\boldsymbol{x})$ can be nonlinearly reconstructed from a very small number of measurements. Consider the problem of reconstructing states lying in a set that is not accurately represented in a low-dimensional subspace, e.g., a curved manifold (see Section 3.2). Linear reconstruction will require a number of measurements at least as large

as the dimension of an approximating subspace. In contrast, it is often be possible to reconstruct the states using a nonlinear function of fewer measurements.

5.2.1 Overview of linear techniques

Sensor placement, measurement selection, inverse problems, and experimental design are extensive fields with a variety of available techniques. A comprehensive review is beyond the scope of this work. However, we can shed some light on the predominant approaches. The use either greedy selection [247, 198, 265, 295, 166, 175] or convex relaxations [125, 262, 296, 79] to select measurements based on a range of objectives including: the performance of Bayesian and maximum likelihood estimators [56, 245], information theoretic criteria [154, 55, 248, 141], and measures of observability in linear dynamical systems [157, 156, 267, 299, 252, 253]. In order to make techniques based on statistical criteria tractable, it is common to make linear and Gaussian assumptions. Simplifying assumptions like linearity are also made in order to yield algorithms that provide theoretical guarantees on performance using greedy algorithms or convex relaxations.

Suppose we are interested in reconstructing a linear function $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{T}\boldsymbol{x}$ of an underlying state $\boldsymbol{x}$ drawn from a probability distribution with zero mean and covariance $\boldsymbol{C}_{\boldsymbol{x}}$. Suppose that our measurements are also linear $\boldsymbol{m}_j(\boldsymbol{x}) = \boldsymbol{M}_j\boldsymbol{x}$ and are corrupted by zero-mean state-independent noise $\boldsymbol{n}_j$. Here, we may select a collection $\mathcal{S}$ of random vectors

$$\boldsymbol{y}_j = \boldsymbol{M}_j\boldsymbol{x} + \boldsymbol{n}_j, \qquad j = 1, \ldots, M. \tag{5.26}$$

Let $\boldsymbol{y}_{\mathcal{S}} = (\boldsymbol{y}_{j_1}, \ldots, \boldsymbol{y}_{j_K})$ be the random vector we observe, with corresponding measurement matrix $\boldsymbol{M}_{\mathcal{S}}^T = \begin{bmatrix} \boldsymbol{M}_{j_1}^T & \cdots & \boldsymbol{M}_{j_K}^T \end{bmatrix}$ and noise $\boldsymbol{n}_{\mathcal{S}} = (\boldsymbol{n}_{j_1}, \ldots, \boldsymbol{n}_{j_K})$ with covariance $\boldsymbol{C}_{\boldsymbol{n}_{\mathcal{S}}}$. Under these assumptions, the optimal linear estimate of $\boldsymbol{g}(\boldsymbol{x})$ and its error covariance are given by Proposition 5.2.1.

Proposition 5.2.1 (Optimal Linear Estimator)**.** *Noting the assumed independence $\boldsymbol{n}_{\mathcal{S}} \perp\!\!\!\perp \boldsymbol{x}$, define the covariance matrices*

$$\begin{aligned} \boldsymbol{C}_{\boldsymbol{g}} &= \mathbb{E}[\boldsymbol{g}\boldsymbol{g}^T] = \boldsymbol{T}\boldsymbol{C}_{\boldsymbol{x}}\boldsymbol{T}^T \\ \boldsymbol{C}_{\boldsymbol{y}_{\mathcal{S}}} &= \mathbb{E}[\boldsymbol{y}_{\mathcal{S}}\boldsymbol{y}_{\mathcal{S}}^T] = \boldsymbol{M}_{\mathcal{S}}\boldsymbol{C}_{\boldsymbol{x}}\boldsymbol{M}_{\mathcal{S}}^T + \boldsymbol{C}_{\boldsymbol{n}_{\mathcal{S}}} \\ \boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_{\mathcal{S}}} &= \mathbb{E}[\boldsymbol{g}\boldsymbol{y}_{\mathcal{S}}^T] = \boldsymbol{T}\boldsymbol{C}_{\boldsymbol{x}}\boldsymbol{M}_{\mathcal{S}}^T, \end{aligned} \tag{5.27}$$

and let $(\cdot)^+$ denote the Moore-Penrose pseudoinverse. Then the Optimal Linear Estimate (OLE), $\hat{\boldsymbol{g}} = \boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{y}_{\mathcal{S}})$, that minimizes the average square error $\mathbb{E}\,\|\hat{\boldsymbol{g}} - \boldsymbol{g}\|_2^2$ among all linear functions of $\boldsymbol{y}_{\mathcal{S}}$ is

given explicitly by

$$\boxed{\hat{\boldsymbol{g}} = \boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{y}_{\mathcal{S}}) = \boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_{\mathcal{S}}} \boldsymbol{C}^{+}_{\boldsymbol{y}_{\mathcal{S}}} \boldsymbol{y}_{\mathcal{S}}.} \tag{5.28}$$

The optimal estimate $\hat{\boldsymbol{g}}$ *is always unique, while the optimal estimator* $\boldsymbol{\Phi}_{\mathcal{S}}$ *is unique if and only if* $\boldsymbol{C}_{\boldsymbol{y}_{\mathcal{S}}}$ *is invertible. The error covariance of the optimal linear estimate depends on the choice of measurements* $\mathcal{S} \subset \mathcal{M}$ *and is given by*

$$\boxed{\boldsymbol{C}_{\boldsymbol{e}}(\mathcal{S}) = \mathbb{E}[(\boldsymbol{g} - \hat{\boldsymbol{g}})(\boldsymbol{g} - \hat{\boldsymbol{g}})^T] = \boldsymbol{C}_{\boldsymbol{g}} - \boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_{\mathcal{S}}} \boldsymbol{C}^{+}_{\boldsymbol{y}_{\mathcal{S}}} \boldsymbol{C}_{\boldsymbol{y}_{\mathcal{S}},\boldsymbol{g}} = \boldsymbol{C}_{\boldsymbol{g}} - \boldsymbol{C}_{\hat{\boldsymbol{g}}}(\mathcal{S}),} \tag{5.29}$$

where $\boldsymbol{C}_{\hat{\boldsymbol{g}}}(\mathcal{S})$ *is the covariance of the estimate. When* $\boldsymbol{C}_{\boldsymbol{x}} \succ 0$ *and* $\boldsymbol{C}_{\boldsymbol{n}_{\mathcal{S}}} \succ 0$ *are positive-definite, then the error covariance can be re-written using the matrix inversion lemma as*

$$\boxed{\boldsymbol{C}_{\boldsymbol{e}}(\mathcal{S}) = \boldsymbol{T}\bar{\boldsymbol{P}}(\mathcal{S})^{-1}\boldsymbol{T}^T, \qquad \bar{\boldsymbol{P}}(\mathcal{S}) = \boldsymbol{C}_{\boldsymbol{x}}^{-1} + \boldsymbol{P}(\mathcal{S}), \qquad \boldsymbol{P}(\mathcal{S}) = \boldsymbol{M}_{\mathcal{S}}^T \boldsymbol{C}_{\boldsymbol{n}_{\mathcal{S}}}^{-1} \boldsymbol{M}_{\mathcal{S}}.} \tag{5.30}$$

Proof. We give the proof in Appendix 5.A. □

When the random variables involved are Gaussian, then the optimal linear estimator described by Proposition 5.2.1 is optimal among all (nonlinear) estimators since $\hat{\boldsymbol{g}}$ given by Eq. 5.28 is the conditional expectation of $\boldsymbol{g}(\boldsymbol{x})$ given $\boldsymbol{y}_{\mathcal{S}}$.

Various performance metrics for sensor placement use the error covariance matrix in Eq. 5.29. Minimizing the average square error $\operatorname{Tr} \boldsymbol{C}_{\boldsymbol{e}}(\mathcal{S})$ is called Bayesian A-optimality [56]. Here, the "Baysian" characterization refers to the fact that the variable $\boldsymbol{x}$ has a prior distribution. Related "maximum likelihood" approaches can be applied when there is no prior distribution for $\boldsymbol{x}$. When the noise from different sensors are independent, $\boldsymbol{n}_i \perp\!\!\!\perp \boldsymbol{n}_j$, $i \neq j$, and the covariance matrices $\boldsymbol{C}_{\boldsymbol{x}}$ and each $\boldsymbol{C}_{\boldsymbol{n}_j}$ are positive-definite, the average square error objective for an optimal linear estimator can be optimized using a convex relaxation approach described by S. Joshi and S. Boyd [125]. Under these assumptions, $\boldsymbol{P}(\mathcal{S})$ in the expression for the error covariance given by Eq. 5.30 becomes a modular matrix-valued function, that is,

$$\boldsymbol{P}(\mathcal{S}) = \sum_{j \in \mathcal{S}} \boldsymbol{M}_j^T \boldsymbol{C}_{\boldsymbol{n}_j}^{-1} \boldsymbol{M}_j = \sum_{j \in \mathcal{S}} \boldsymbol{P}(\{j\}) \qquad \forall \mathcal{S} \subset \mathcal{M}. \tag{5.31}$$

Thanks to the convexity of matrix inversion with respect to the positive-definite (Loewner) ordering (see Lemma 5.A.2 in Appendix 5.A for a proof), the Bayesian A-optimality objective has the

following convex relaxation

$$f(\boldsymbol{s}) = \operatorname{Tr}\Big[\boldsymbol{T}\Big(\boldsymbol{C}_{\boldsymbol{x}}^{-1} + \sum_{j=1}^{M} s_j \boldsymbol{P}(\{j\})\Big)^{-1}\boldsymbol{T}^T\Big], \qquad \boldsymbol{s} = (s_1, \ldots, s_M). \tag{5.32}$$

In [125], it is shown that the relaxed square error objective given by Eq. 5.32 can be efficiently optimized with respect to a convex relaxation of the cardinality constraint $|\mathcal{S}| \leq K$ given by

$$\sum_{j=1}^{M} s_j = K, \qquad 0 \leq s_j \leq 1 \quad j = 1, \ldots, M. \tag{5.33}$$

We note that in the design of experiments, the constraint $0 \leq [\boldsymbol{s}]_j \leq 1$ may be removed when the same experiment can be performed multiple times.

Unfortunately, the average square error of the optimal linear estimate does not lead to a submodular optimization objective, except under the restrictive assumptions explored by A. Das and D. Kempe in [71]. In particular, we may consider a candidate greedy optimization objective

$$f(\mathcal{S}) = \operatorname{Tr}\left(\boldsymbol{T}\bar{\boldsymbol{P}}(\emptyset)^{-1}\boldsymbol{T}^T\right) - \operatorname{Tr}\left(\boldsymbol{T}\bar{\boldsymbol{P}}(\mathcal{S})^{-1}\boldsymbol{T}^T\right), \tag{5.34}$$

which is normalized so that $f(\emptyset) = 0$ and monotone non-decreasing thanks to Lemma 5.A.1 in Appendix 5.A. However, Example 5.2.2 below demonstrates that this objective can fail to be submodular, and the ratio describing this failure can be arbitrarily large. The composition of any concave function like $x \mapsto -x^{-1}$ with a modular function $\mathcal{S} \mapsto \sum_{j\in\mathcal{S}} p_j$ is submodular (we give a proof in the appendix of [195]**). However, this does not hold for matrix-valued modular functions like $\mathcal{S} \mapsto \boldsymbol{P}(\mathcal{S})$ even though $\boldsymbol{P} \mapsto -\operatorname{Tr}\left[\boldsymbol{T}\left(\boldsymbol{C}_{\boldsymbol{x}}^{-1} + \boldsymbol{P}\right)^{-1}\boldsymbol{T}^T\right]$ is concave with respect to positive semi-definite matrices $\boldsymbol{P}$.

Example 5.2.2 (the MSE objective function Eq. 5.34 is NOT submodular)**.** *Choose* $\alpha \neq 0$, $\boldsymbol{T} = \boldsymbol{I}_2$ *and let*

$$\bar{\boldsymbol{P}}(\emptyset) = \begin{bmatrix} 1 & 0 \\ 0 & \alpha^2 \end{bmatrix}, \qquad \boldsymbol{P}(\{1\}) = \begin{bmatrix} 1 & \alpha \\ \alpha & \alpha^2 \end{bmatrix}, \qquad \boldsymbol{P}(\{2\}) = \begin{bmatrix} 0 & 0 \\ 0 & \alpha^2 \end{bmatrix}. \tag{5.35}$$

A staight-forward calculation shows that

$$f(\{2\}) - f(\emptyset) = \operatorname{Tr}\left[\bar{\boldsymbol{P}}(\emptyset)^{-1}\right] - \operatorname{Tr}\left[\left(\bar{\boldsymbol{P}}(\emptyset) + \boldsymbol{P}(\{2\})\right)^{-1}\right] = \frac{1}{2\alpha^2} \tag{5.36}$$

and

$$\begin{aligned} f(\{1,2\}) - f(\{1\}) &= \operatorname{Tr}\left[\left(\bar{\boldsymbol{P}}(\emptyset) + \boldsymbol{P}(\{1\})\right)^{-1}\right] - \operatorname{Tr}\left[\left(\bar{\boldsymbol{P}}(\emptyset) + \boldsymbol{P}(\{1\}) + \boldsymbol{P}(\{2\})\right)^{-1}\right] \\ &= \frac{1}{15}\left(\frac{4}{\alpha^2} + 1\right). \end{aligned} \tag{5.37}$$

The ratio of the increase in the objective is then

$$\frac{f(\{1,2\}) - f(\{1\})}{f(\{2\}) - f(\emptyset)} = \frac{2}{15}\left(\alpha^2 + 4\right), \tag{5.38}$$

which can be made arbitrarily large by increasing the constant α*. However, this contradicts the definition of submodularity which says that this ratio cannot exceed* 1*.*

Fortunately it is possible to formulate other objectives based on the error covariance of the optimal linear estimate that are submodular and admit efficient greedy approximation algorithms with the performance guarantees provided by Theorem 5.0.2. In particular, under the additional assumption that $\boldsymbol{T}$ is invertible, M. Shamaiah et al. [247] show that the objective

$$f(\mathcal{S}) = \log\det\left(\boldsymbol{T}\bar{\boldsymbol{P}}(\emptyset)^{-1}\boldsymbol{T}^T\right) - \log\det\left(\boldsymbol{T}\bar{\boldsymbol{P}}(\mathcal{S})^{-1}\boldsymbol{T}^T\right), \tag{5.39}$$

is submodular, in addition to being normalized and monotone non-decreasing. Since Kalman filtering a special case of optimal linear estimation, such an objective has been used by V. Tzoumas et al. in [267] to greedily place sensors for optimal state estimation in linear systems. A related submodular objective based on the log determinant of observability and controllability Gramians was considered by T. H. Summers et al. in [252, 253]. Unfortunately, the objective in Eq. 5.39 may fail to be submodular when $\boldsymbol{T}$ is not invertible. Even when $\boldsymbol{T}$ fails to be invertible, the log determinant objective admits a straightforward convex relaxation analogous to Eq. 5.32, which is also explored by S. Joshi and S. Boyd in [125]. Minimizing the determinant of the error covariance is referred to a D-optimality in the design of experiments [56]. Under Gaussian assumptions it corresponds to minimizing the entropy of the estimated variables given the measurements.

The group-LASSO technique proposed by M. Yuan and Y. Lin in [296] takes a data-driven approach to feature selection in linear estimation and regression problems. In particular, they solve a regularized least squares problem

$$\underset{\boldsymbol{A}_1,\ldots,\boldsymbol{A}_M}{\text{minimize}} \sum_{i=1}^{M} \left\| \boldsymbol{g}(\boldsymbol{x}_i) - \sum_{j=1}^{M} \boldsymbol{A}_j \boldsymbol{m}_j(\boldsymbol{x}_i) \right\|_2^2 + \gamma \sum_{j=1}^{M} \|\boldsymbol{A}_j\|_F \tag{5.40}$$

for matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_M$ defining a linear reconstruction function based on a collection of sampled data consisting of relevant quantities $\boldsymbol{g}(\boldsymbol{x}_i)$ and the measurements from every sensor $\boldsymbol{m}_{\mathcal{M}}(\boldsymbol{x}_i)$) for each sample $i = 1, \ldots, m$. As the coefficient $\gamma \geq 0$ is increased, the regularization encourages many of the matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_M$ to be identically zero. The sensors or features chosen by the group LASSO method correspond to the remaining nonzero matrices. The original LASSO method proposed by R. Tibshirani [262] is a special case of Eq. 5.40 in which each matrix $\boldsymbol{A}_j$ consists of a single element.

5.2.2 The need for nonlinear reconstruction

In some problems the relationship between the relevant quantities $\boldsymbol{g}$ and any small collection of measurements $\boldsymbol{m}_{\mathcal{S}}$ is not approximated well by any linear function. One of the situations where this problem arises is in reconstructing states that live on low-dimensional manifolds that are not captured in low-dimensional subspaces, as we discussed earlier in Section 3.2. This makes linear reconstruction impossible because such a reconstruction

$$\boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{m}_{\mathcal{S}}(x)) = \boldsymbol{A}\boldsymbol{m}_{\mathcal{S}}(x) \tag{5.41}$$

is confined to a subspace of dimension at most $d_{\mathcal{S}}$ contained in the range of $\boldsymbol{A}$. As we discuss in [195]**, if $\sigma_1^2 \geq \cdots \geq \sigma_q^2 \geq 0$ are the principal variances of $\boldsymbol{g}(x)$, then the greatest possible fraction of the variance one can reconstruct using $d_{\mathcal{S}}$ measurements is bounded by

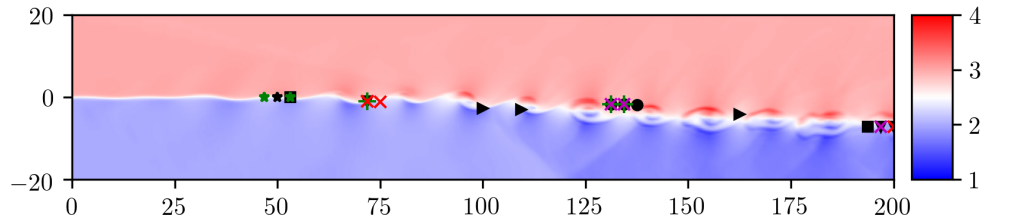
$$R^2 \leq \frac{\sigma_1^2 + \cdots + \sigma_{d_{\mathcal{S}}}^2}{\sigma_1^2 + \cdots + \sigma_q^2}. \tag{5.42}$$

Of course achieving equality in this upper bound requires the measurements to be linearly isomorphic to the leading $d_{\mathcal{S}}$ principal components of $\boldsymbol{g}(x)$, which is rarely the case in practice. Alternatively, we may have a very low-dimensional $\boldsymbol{g}$ whose entries are poorly represented in the span of every small subset of the measurement functions $\boldsymbol{m}_j$.

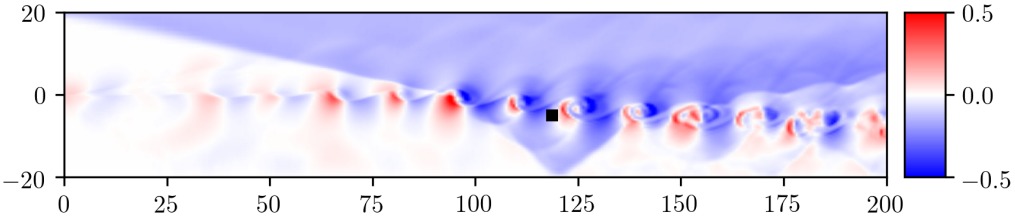
To illustrate, suppose we wish to select fundamental eigenfunctions using spectral embedding techniques as described in Section 5.1.3. Orthonormality of the eigenfunctions $\varphi_1, \ldots, \varphi_N$ means that the covariance matrix for $\varphi_i \varphi_j$ is isotropic,

$$\mathbb{E}[\varphi_i \varphi_j] = \int_{\mathcal{X}} \varphi_i(x)\varphi_j(x) \, \mathrm{d}\, \mu(x) = \delta_{i,j}. \tag{5.43}$$

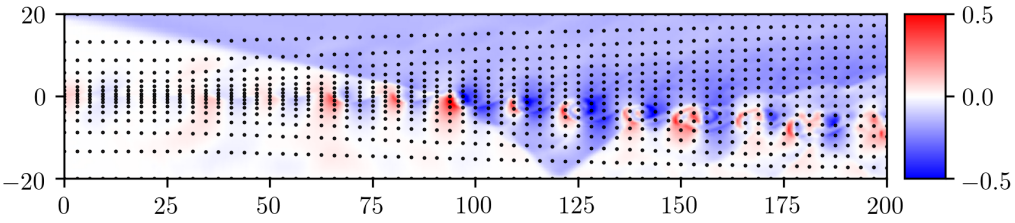
Consequently, if we chose any $d_{\mathcal{S}}$ eigenfunctions, then we can linearly reconstruct precisely $R^2 =$



(a) stream-wise u velocity component



(b) transverse v velocity component



(c) available sensor locations

Figure 5.2.1: A snapshot of the u and v velocity components in the shock mixing-layer flow is shown in (a) and (b) along with the sensors selected using various methods from among the two components at 1105 available locations shown in (c). These methods include LASSO with PCA (black o), LASSO with Isomap (red x) greedy Bayes D-optimality (magenta x), convex Bayes D-optimality (black >), convex D-optimality for modes 3 and 4 (black v), QR pivoting (green +), and secant-based techniques using detectable differences (#1, #2: green star, #3: black star) and the amplification threshold method (black square).

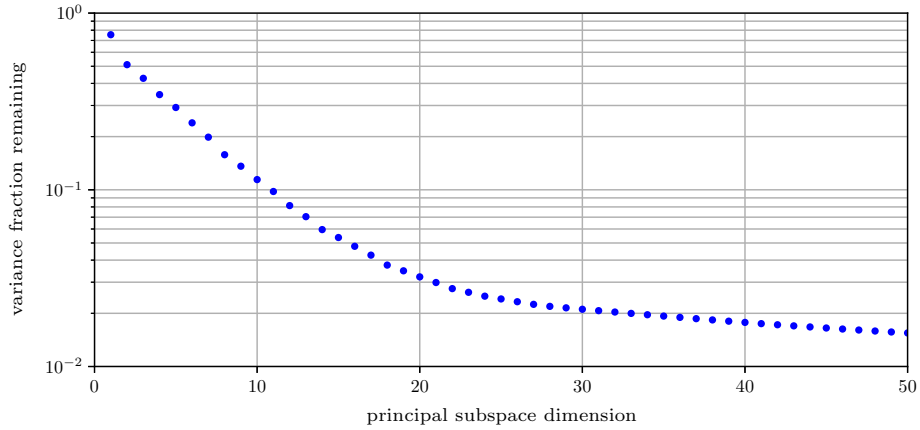
d_S/N of the total variance of $\boldsymbol{g}(x) = (\varphi_1(x), \ldots, \varphi_N(x))$. In particular, we can linearly reconstruct the eigenfunctions we chose, and none of the variance of the others. On the other hand, if we choose a fundamental set of eigenfunctions, then the rest can be nonlinearly reconstructed.

Problems with linear reconstruction also appear in fluid dynamics. We illustrate this in [195]** by considering a fluid flow proposed by H. C. Yee et al. [292] in which an oblique shock wave interacts with a spatially developing mixing layer as shown in Figure 5.2.1. The complicated physics arising from this interaction leads to small-scale advecting flow structures and moving shock waves which cannot be represented using a low-dimensional superposition of modes. This is indicated by the slow decay of the variance not captured by principal subspaces plotted in Figure 5.2.2a. By the argument above, we would need at least $d_S = 11$ well-chosen sensor measurements to have a hope of linearly reconstructing 90% of the variance in this flow's velocity field.

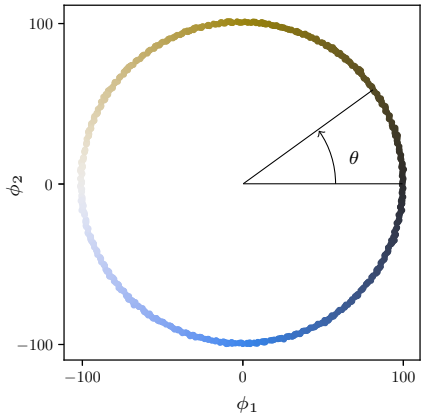
On the other hand, the flow structures appearing downstream in the shock-mixing layer flow are essentially driven by the periodic fluctuations in the mixing layer that appear upstream. Consequently, this flow is very nearly periodic and the states lie close to a one-dimensional loop in state space. Embedding this loop as a circle in the plane shows that the state of this flow can be reconstructed with high accuracy as a nonlinear function of two measurements providing a coordinate system in the plane. Figure 5.2.2b shows the leading two nonlinear embedding coordinates provided by the isomaps algorithm [260], which we use to define the flow's phase on its orbit. As we will see later, it is possible to reconstruct the state of this flow from the velocity measurements taken at the two locations in the flow marked by green stars in Figure 5.2.1a. In fact, we used these two measurements to reconstruct $R^2 = 0.986$ of the velocity field's variance on unseen snapshots by using Gaussian process regression [216] to fit a nonlinear reconstruction map. On the other hand, with only two sensor measurements, the highest fraction of the velocity field's variance one can capture using linear reconstruction is $R^2 < 0.5$.

5.2.3 The need for nonlinear selection

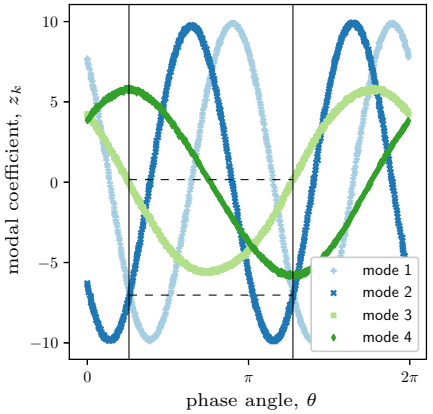
Because of the need for nonlinear reconstruction when using small numbers of measurements, linear reconstruction performance is a poor criterion for selecting small numbers of sensors. However, in the case when lower energy components of the flow are functions of the higher energy components, linear techniques (see Section 5.2.1) aiming to reconstruct the most energy possible may still find sensors that yield acceptable nonlinear reconstruction performance. On the other hand, when higher-energy components are functions of lower-energy ones, linear sensor placement techniques consistently fail to identify small sets of sensors that enable nonlinear reconstruction. The shock-mixing layer flow presents such a challenge because its two highest energy principal components oscillate at twice the fundamental frequency of the flow as seen in Figure 5.2.2c. In [195]** we use several representative linear techniques to choose sets of three sensors among the available locations shown in Figure 5.2.1c. These sensor locations are indicated on Figures 5.2.1a and 5.2.1b using different markers. The corresponding measurements made by these sensors are plotted in Figure 5.2.3 and colored according to the phase of the underlying state. We see that each set of sensors chosen by linear methods produces nearly identical measurements at multiple distinct phases around the shock-mixing layer flow's orbit. Consequently, these states cannot be reconstructed (linearly or nonlinearly) from the measurements. Interestingly, three of the linear methods: LASSO to reconstruct the leading 100 principal components in Eq. 5.40, greedy Bayesian D-optimality, and pivoted QR factorization (see



(a) variance orthogonal to principal subspaces

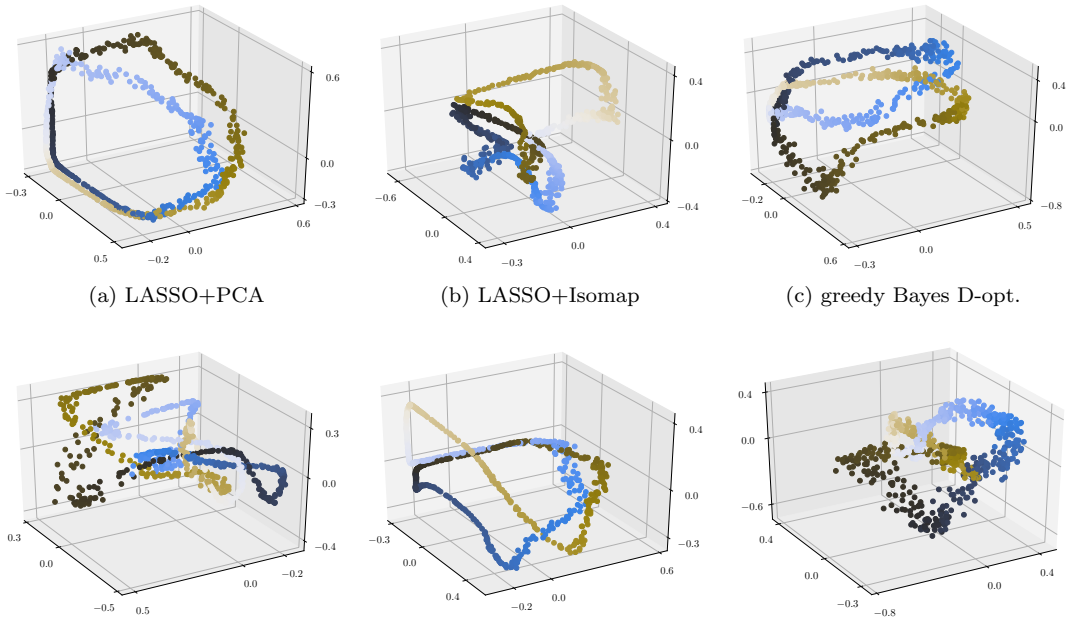


(b) Isomap coordinates



(c) PCA coefficients

Figure 5.2.2: The linear and nonlinear dimension reduction techniques PCA (a.k.a POD) and Isomap are applied to the shock-mixing layer data. (a) shows the remaining fraction of the total variance orthogonal to each leading principal subspace. (b) plots the data in the leading two Isomap embedding coordinates, revealing that it lies very near a loop in state space. (c) shows how the leading principal components (modal coefficients) vary with the phase angle around the loop. The black vertical lines reveal distinct points where the leading three principal components are identical.



(a) LASSO+PCA (b) LASSO+Isomap (c) greedy Bayes D-opt.

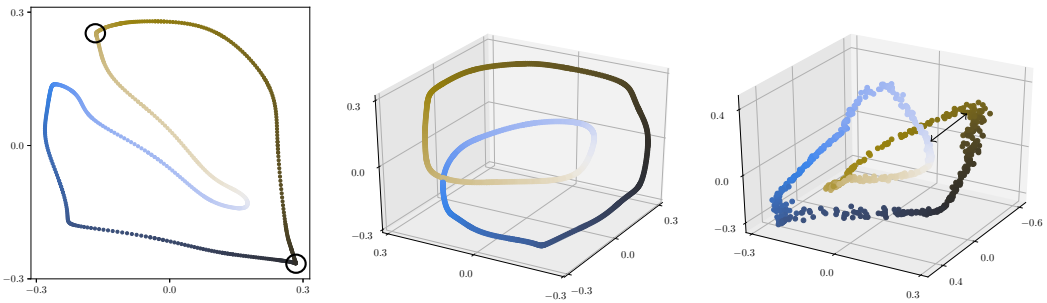
(d) convex Bayes D-opt. (e) pivoted QR (f) convex M.L. D-opt., modes $3, 4$

Figure 5.2.3: these plots show the measurements made by sensors selected using various linear methods on the shock-mixing layer flow problem. Each dot indicates the values measured by the sensors and its color indicates the phase of the corresponding flowfield. Each set of sensors make identical or nearly identical measurements on distinct flowfields, indicated by overlapping points with different colors. These sensors cannot tell those flowfields apart since the measurements are the same.

implementation details in [195]**) all produce self-intersecting measurements separated by approximately 180 degrees of phase. This is exactly what we would expect if we were measuring the leading three most energetic principal components, as illustrated by the two phase angles marked by black vertical lines 180 degrees apart in Figure 5.2.2c where the leading three principal components take identical values.

5.3 Greedy selection based on secants

In this section, we summarize the nonlinear sensor placement techniques developed by S. E. Otto and C. W. Rowley in [195]**. These methods are capable of selecting measurements that are one-to-one with the quantities we wish to reconstruct. For instance, the measurements selected using these techniques on the shock-mixing layer flow are shown in Figure 5.3.1. In each case it is possible to reconstruct the underlying phase, and hence the full state of the system. The techniques we propose are also capable of selecting fundamental sets of eigenfunctions for spectral embedding (and possibly also for Koopman operators) as we discussed in Section 5.1.3. The main idea behind these



(a) secant detect. diffs., $\#(\mathcal{S}) = 2$ (b) secant detect. diffs., $\#(\mathcal{S}) = 3$ (c) secant amplification tol.

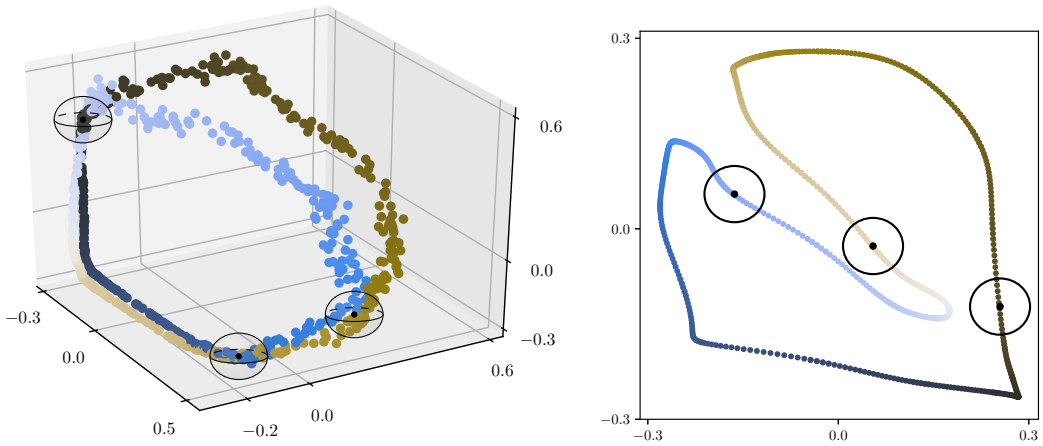
Figure 5.3.1: We show the measurements made by sensors selected using secant-based greedy optimization methods on the shock-mixing layer flow problem. Each dot indicates the values measured by the sensors and its color indicates the phase of the corresponding flowfield. In each case, the selected sensors make distinct measurements for distinct states, enabling reconstruction of the state from the measurements.

techniques is to consider pairs of sampled states, referred to as "secants", and to choose sensors that produce different measurements for pairs of states that have different values of the quantities we wish to reconstruct. This is essentially a sampled version of the vertical line test for the existence of a reconstruction function $\mathbf{\Phi}_{\mathcal{S}}$. We quantify how far $\mathbf{\Phi}_{\mathcal{S}}$ is from becoming multi-valued using three different submodular objectives for greedy optimization. We refer to the sampled approximation of the underlying set $\mathcal{X}$ as $\mathcal{X}_N = \{x_1, \ldots, x_N\} \subset \mathcal{X}$. In [195]** we provide a number of performance guarantees with respect to the underlying set $\mathcal{X}$ depending on the fineness of this sampling and the number of secants considered in the objectives. We shall not discuss these results in detail since they can be found in our paper.

Remark 5.3.1. *All of the optimization problems we discuss in this section may also be formulated as linear programs with the usual convex relaxation of the cardinality of the selected sensors — namely, by replacing* $|\mathcal{S}|$ *with* $s_1 + \cdots + s_M$ *and the constraint* $0 \leq s_j \leq 1$*. However, it is much more computationally expensive to solve such linear programs for large numbers of sensors and secants than it is to rely on greedy algorithms.*

5.3.1 Maximizing detectable differences

If two states produce nearby measurements, then the values to be reconstructed should also be close. Otherwise, a small disturbance of the measurements could result in a large reconstruction error. One way to quantify the performance of a given set of sensors is to minimize the distance between values to be reconstructed when the corresponding measurements are nearby. In particular, we may choose a distance or "detection threshold" $\gamma > 0$ in the measurement space according to the magnitude



(a) LASSO measurements

(b) secant detect. diffs. measurements

Figure 5.3.2: we show the measurements made by two different sets of sensors along with balls indicating a choice of detection threshold γ. We observe that the LASSO measurements have large sums of square differences between states at points lying within the same balls, whereas such fluctuations are small for the sensors chosen using the secant-based detectable differences method.

of noise or disturbances in the measurements that we want the reconstruction function to tolerate. We consider the sum of square differences between target variables from states with measurements closer together than the detection threshold. That is, we seek to minimize

$$F_\gamma(\mathcal{S}) = \frac{1}{N^2} \sum_{x,x' \in \mathcal{X}_N} \mathbb{1}\left\{\|\boldsymbol{m}_\mathcal{S}(x) - \boldsymbol{m}_\mathcal{S}(x')\|_2 < \gamma\right\} \|\boldsymbol{g}(x) - \boldsymbol{g}(x')\|_2^2, \tag{5.44}$$

where the function $\mathbb{1}\{A\} = 1$ if A is true and 0 if A is false. We illustrate this idea in Figure 5.3.2. Balls indicate a choice of detection threshold γ in the measurement spaces for sensors chosen using LASSO and the method we present here. The LASSO sensors have large differences among the phases of states within the same balls, whereas the sensors we choose using our "detectable differences" method have only small phase differences within each ball, resulting in a one-to-one mapping of the state.

Minimizing the sum of square undetectable differences Eq. 5.44 is equivalent to maximizing the sum of square differences between relevant quantities corresponding to states with measurements separated by at least γ, i.e., the sum of detectable differences

$$\tilde{f}_\gamma(\mathcal{S}) = \frac{1}{N^2} \sum_{x,x' \in \mathcal{X}_N} \mathbb{1}\left\{\|\boldsymbol{m}_\mathcal{S}(x) - \boldsymbol{m}_\mathcal{S}(x')\|_2 \geq \gamma\right\} \|\boldsymbol{g}(x) - \boldsymbol{g}(x')\|_2^2. \tag{5.45}$$

While Eq. 5.45 is normalized such that $\tilde{f}_\gamma(\emptyset) = 0$ and monotone increasing, it is unfortunately not submodular.

The sum of detectable differences given by Eq. 5.45 may be relaxed into a submodular function by relaxing the discontinuous detection threshold $\{\|\boldsymbol{m}_{\mathcal{S}}(x) - \boldsymbol{m}_{\mathcal{S}}(x')\|_2 \geq \gamma\}$ into a continuous weight function

$$w_{\gamma,x,x'}(\mathcal{S}) = \min\left\{\frac{1}{\gamma^2}\sum_{j\in\mathcal{S}}\|\boldsymbol{m}_j(x) - \boldsymbol{m}_j(x')\|_2^2,\ 1\right\}. \tag{5.46}$$

For each secant $x, x' \in \mathcal{X}_N$, Eq. 5.46 is a concave function composed with a modular function of $\mathcal{S}$, which implies that the weights and the resulting relaxed objective

$$\boxed{f_\gamma(\mathcal{S}) = \frac{1}{N^2}\sum_{x,x'\in\mathcal{X}_N} w_{\gamma,x,x'}(\mathcal{S})\|\boldsymbol{g}(x) - \boldsymbol{g}(x')\|_2^2.} \tag{5.47}$$

are submodular (for a proof, see [195]**). By greedily maximizing this relaxed objective under a fixed sensor budget $|\mathcal{S}| \leq K$, the classical performance guarantee described in Theorem 5.0.2 holds for Eq. 5.47.

Even though Eq. 5.47 may be larger than the original detectable differences objective in Eq. 5.45, we show in [195]** that the undetectable differences in Eq. 5.44 at a reduced threshold $\alpha\gamma$ for any $0 < \alpha < 1$ are bounded by

$$F_{\alpha\gamma}(\mathcal{S}) \leq \frac{F_\infty - f_\gamma(\mathcal{S})}{1-\alpha^2}, \qquad \text{where} \qquad F_\infty = \sum_{x,x'\in\mathcal{X}_N}\|\boldsymbol{g}(x) - \boldsymbol{g}(x')\|_2^2. \tag{5.48}$$

Putting our bounds together and letting $\tilde{\mathcal{S}}_K^* \subset \mathcal{M}$ denote a minimizer of Eq. 5.44 under the sensor budget $|\mathcal{S}| \leq K$, we find that the sequence of greedily chosen sets $\mathcal{S}_k$ satisfy a worst-case bound

$$F_{\alpha\gamma}(\mathcal{S}_k) \leq \min_{K=1,\ldots,M}\frac{1}{1-\alpha^2}\left[\left(1 - e^{-k/K}\right)F_\gamma(\tilde{\mathcal{S}}_K^*) + e^{-k/K}F_\infty\right], \tag{5.49}$$

for every $k = 1, \ldots, M$. This bound tells us how well the greedy algorithm must perform in comparison with the optimal solutions using different sensor budgets and larger detection thresholds. Perhaps a more useful lower bound on the optimal performance is given by

$$\boxed{F_\gamma(\tilde{\mathcal{S}}_K^*)) \geq \max_{k=1,\ldots,M}\left[F_\infty - \frac{e^{k/K}}{e^{k/K}-1}f_\gamma(\mathcal{S}_k)\right],} \tag{5.50}$$

which can be computed from the greedily chosen sets $\mathcal{S}_k$ after running the algorithm. We give the proofs of the bounds in Eq. 5.49 and Eq. 5.50 in Appendix 5.A.

Another useful property of the objective Eq. 5.47 (and Eq. 5.44) is that it can be accurately

approximated with high probability by normalized sums over much smaller collections of randomly chosen secants. This is essentially a result of the law of large numbers, which we discuss further in [195]**.

5.3.2 Minimal sensing to achieve measurement separation

The greedy optimization problem described in Section 5.3.1 is aimed at maximizing a reconstruction performance metric averaged over pairs of states. Therefore, it cannot produce guarantees about our ability to accurately reconstruct individual states given the selected measurements. In this section, we present a modification of the maximizing detectable differences approach, which is capable of choosing nearly the minimum possible number of sensors so that the relevant quantities can be recovered within a user-specified accuracy $\varepsilon > 0$. In particular, we choose sensors so that every pair of states producing measurements closer together than $\gamma > 0$ have relevant quantities differing by less than ε. Stated another way, if two states have relevant quantities differing by at least ε, the measurements must separate them by at least γ.

Let us assume that the above measurement separation condition can be met by using all of the available measurements taken together, i.e., by using $\mathcal{S} = \mathcal{M}$. Then, it is possible to encode the measurement separation condition for smaller subsets $\mathcal{S} \subset \mathcal{M}$ using the normalized, monotone non-decreasing, submodular function

$$\boxed{f_{\gamma,\varepsilon}(\mathcal{S}) = \frac{1}{N^2} \sum_{\substack{x,x' \in \mathcal{X}_N: \\ \|\boldsymbol{g}(x) - \boldsymbol{g}(x')\|_2 \geq \varepsilon}} w_{\gamma,x,x'}(\mathcal{S}) \|\boldsymbol{g}(x) - \boldsymbol{g}(x')\|_2^2.} \tag{5.51}$$

This function closely resembles the detectable differences objective Eq. 5.47, except that the sum is taken only over secants with relevant quantities separated by at least ε. Our assumption that the measurement separation condition is met using $\mathcal{S} = \mathcal{M}$ translates into condition that $w_{\gamma,x,x'}(\mathcal{M}) = 1$ for every $x, x' \in \mathcal{X}_N$ such that $\|\boldsymbol{g}(x) - \boldsymbol{g}(x')\|_2 \geq \varepsilon$. We observe that if there is even a single secant $x, x' \in \mathcal{X}_N$ such that $\|\boldsymbol{g}(x) - \boldsymbol{g}(x')\|_2 \geq \varepsilon$, but $\|\boldsymbol{m}_{\mathcal{S}}(x) - \boldsymbol{m}_{\mathcal{S}}(x')\|_2 < \gamma$, then we have $w_{\gamma,x,x'}(\mathcal{S}) < 1$ and so $f_{\gamma,\varepsilon}(\mathcal{S}) < f_{\gamma,\varepsilon}(\mathcal{M})$. On the other hand, if the measurement separation condition is met using a collection of sensors $\mathcal{S}$, then we have $f_{\gamma,\varepsilon}(\mathcal{S}) = f_{\gamma,\varepsilon}(\mathcal{M})$. Therefore, the measurement separation condition is equivalent to the condition that $f_{\gamma,\varepsilon}(\mathcal{S}) = f_{\gamma,\varepsilon}(\mathcal{M})$.

We may therefore seek to find the minimum possible number of sensors meeting the measurement

separation condition by solving the submodular set-covering problem

$$\boxed{\underset{\mathcal{S}\subset\mathcal{M}}{\text{minimize}} \quad |\mathcal{S}| \qquad \text{s.t.} \qquad f_{\gamma,\varepsilon}(\mathcal{S}) = f_{\gamma,\varepsilon}(\mathcal{M}).} \tag{5.52}$$

This problem is always feasible, but a solution will only produce the desired measurement separation if it is achieved with $\mathcal{S} = \mathcal{M}$. Thanks to the classical result by L. A. Wolsey [283], a greedy algorithm that maximizes $f_{\gamma,\varepsilon}$ at each step and stops when $f_{\gamma,\varepsilon}(\mathcal{S}) = f_{\gamma,\varepsilon}(\mathcal{M})$ will find, within a constant factor, the minimum possible number of sensors. In particular, let the "increment condition number" be the ratio of the largest and smallest increments in the objective,

$$\kappa = \frac{f_{\gamma,\varepsilon}(\mathcal{S}_1)}{f_{\gamma,\varepsilon}(\mathcal{S}_K) - f_{\gamma,\varepsilon}(\mathcal{S}_{K-1})}, \tag{5.53}$$

where $f_{\gamma,\varepsilon}(\mathcal{S}_K) = f_{\gamma,\varepsilon}(\mathcal{M})$ and $f_{\gamma,\varepsilon}(\mathcal{S}_{K-1}) < f_{\gamma,\varepsilon}(\mathcal{M})$. Then any optimal solution $\mathcal{S}^*$ of the set covering problem in Eq. 5.52 has at least as many elements as the greedy approximation $\mathcal{S}_K$ up to a constant factor given by

$$|\mathcal{S}^*| \geq \frac{|\mathcal{S}_K|}{1 + \ln \kappa}. \tag{5.54}$$

For the shock-mixing layer problem, we find that this approach selects the same sensors as the detectable differences method over a range of choices of ε and the same value of γ.

This optimization problem is more difficult to down-sample than the detectable differences problem we discussed in Section 5.3.1 since we are interested in a separation criterion that applies to every secant rather than an average. In [195]**, we discuss a down-sampling method in which we choose a collection of base points $\mathcal{B}$ at random from $\mathcal{X}_N$ and then formulate the objective in Eq. 5.51 over the smaller set of secants $\mathcal{B} \times \mathcal{X}_N$. With high probability over the collection of base points, this allows us to control the size of the "bad set" of points in $\mathcal{X}_N$ for which there is another point in $\mathcal{X}_N$ with relevant quantities separated by at least ε, but measurements closer together than γ.

5.3.3 Minimal sensing to meet an amplification tolerance

We may want the separations between measurements to grow in proportion to the differences between the quantities to be reconstructed, rather than saturating at the threshold γ as in the methods described above. The techniques discussed in Sections 5.3.1 and 5.3.2 also neglect the structure of measurements at scales smaller than γ, which may be important in reduced-order modeling applications. For instance, the measurements selected by the greedy detectable differences method for the shock-mixing layer problem display cusps (Figure 5.3.1a) where the time derivatives of the

measurements vanish. When building a reduced-order model for the system in the measurement space, these cusps would result in spurious fixed points where the modeled dynamics would get stuck.

To better capture the local and global structure of the data, we seek measurements that keep the Lipschitz constant

$$\|\boldsymbol{\Phi}_{\mathcal{S}}\|_{\mathrm{lip},\mathcal{X}_N} = \max_{\substack{x,x'\in\mathcal{X}_N:\\ \boldsymbol{m}_{\mathcal{S}}(x)\neq\boldsymbol{m}_{\mathcal{S}}(x')}} \frac{\|\boldsymbol{g}(x)-\boldsymbol{g}(x')\|_2}{\|\boldsymbol{m}_{\mathcal{S}}(x)-\boldsymbol{m}_{\mathcal{S}}(x')\|_2} \tag{5.55}$$

of the reconstruction below a user-specified threshold L. The Lipschitz constant in Eq. 5.55 measures the maximum amplification of measurement disturbances when reconstructing the desired quantities $\boldsymbol{g}$. We can formulate the problem of selecting the minimum number of sensors such that $\|\boldsymbol{\Phi}_{\mathcal{S}}\|_{\mathrm{lip},\mathcal{X}_N} \leq L$ as a submodular set-covering problem as in Eq. 5.52, except with an objective defined by

$$\boxed{f_L(\mathcal{S}) = \frac{1}{N^2} \sum_{\substack{x,x'\in\mathcal{X}_N:\\ \boldsymbol{g}(x)\neq\boldsymbol{g}(x')}} \min\left\{\sum_{j\in\mathcal{S}} \frac{\|\boldsymbol{m}_j(x)-\boldsymbol{m}_j(x')\|_2^2}{\|\boldsymbol{g}(x)-\boldsymbol{g}(x')\|_2^2},\ \frac{1}{L^2}\right\}.} \tag{5.56}$$

This function is: normalized so that $f_L(\emptyset) = 0$, monotone non-decreasing, and submodular. The last condition holds since each term composes a concave function with a modular function. We assume that when all of the sensors are used, we achieve the desired Lipschitz constant $\|\boldsymbol{\Phi}_{\mathcal{M}}\|_{\mathrm{lip},\mathcal{X}_N} \leq L$. Under this assumption, the condition $f_L(\mathcal{S}) = f_L(\mathcal{M})$ is equivalent to the desired condition $\|\boldsymbol{\Phi}_{\mathcal{S}}\|_{\mathrm{lip},\mathcal{X}_N} \leq L$ for any subset $\mathcal{S} \subset \mathcal{M}$.

The amplification tolerance optimization problem shares similar down-sampling properties to the problem described in Section 5.3.2. In particular, it is possible to bound the size of an analogous "bad set" with high probability over the same collection of randomly chosen base points. The measurements from the sensors we chose using this approach are shown in Figure 5.3.1c. Additional details can be found in [195]**.

5.4 Selection based on local linearization

In some applications, we may have prior information during the reconstruction process that reduces our uncertainty about the underlying state. Specifically, we may know $x \in \mathcal{U}_\alpha \subseteq \mathcal{X}$. For instance, we may wish to estimate the state of a system when the initial condition is known with some uncertainty. Another example is when we wish to choose sensors that can estimate the state of a system across

different known operating conditions or regimes. Mathematically, we suppose that the state space $\mathcal{X}$ is a smooth manifold covered by a collection of neighborhoods

$$\mathcal{X} \subset \bigcup_{\alpha} \mathcal{U}_{\alpha} \tag{5.57}$$

where we given the information that $x \in \mathcal{U}_{\alpha}$ during the reconstruction or estimation process. It is no longer necessary to reconstruct the relevant quantities based on measurements from any state, but rather based on measurements from states lying in each $\mathcal{U}_{\alpha}$. We shall focus on the case when the neighborhoods $\mathcal{U}_{\alpha}$ are infinitesimally small, allowing us to study the reconstruction problem using linearization. This approach has been used by S. Rao et al. in [215] to develop greedy algorithms for sensor placement based on composite submodular objectives by summing many local versions of the log determinant objective Eq. 5.39 obtained from the original problem's linearization at a collection of points. In this section we shall present some alternatives based on convex optimization and techniques based on generalizations of pivoted QR factorization resembling those employed by DEIM [60, 82].

Suppose that every linearized reconstruction problem can be solved, i.e., for every $x \in \mathcal{X}$ there is a matrix $\boldsymbol{A}_x$ so that $\mathrm{D}\,\boldsymbol{g}(x) = \boldsymbol{A}_x\,\mathrm{D}\,\boldsymbol{m}_{\mathcal{S}}(x)$. Then, Theorem 5.4.1, below guarantees that in a sufficiently small neighborhood $\mathcal{U}_{\alpha}$ of any $x \in \mathcal{X}$, there is a reconstruction function $\boldsymbol{\Phi}_{\mathcal{S},\alpha}$ so that $\boldsymbol{g} = \boldsymbol{\Phi}_{\mathcal{S},\alpha} \circ \boldsymbol{m}_{\mathcal{S}}$ on $\mathcal{U}_{\alpha}$. In the notation of Theorem 5.4.1, the relevant information is represented by the function $g = \boldsymbol{g}$, the measurements are represented by $f = \boldsymbol{m}_{\mathcal{S}}$, and the local reconstruction is given by the resulting $\boldsymbol{\Phi}_{\mathcal{S},\alpha} = h$.

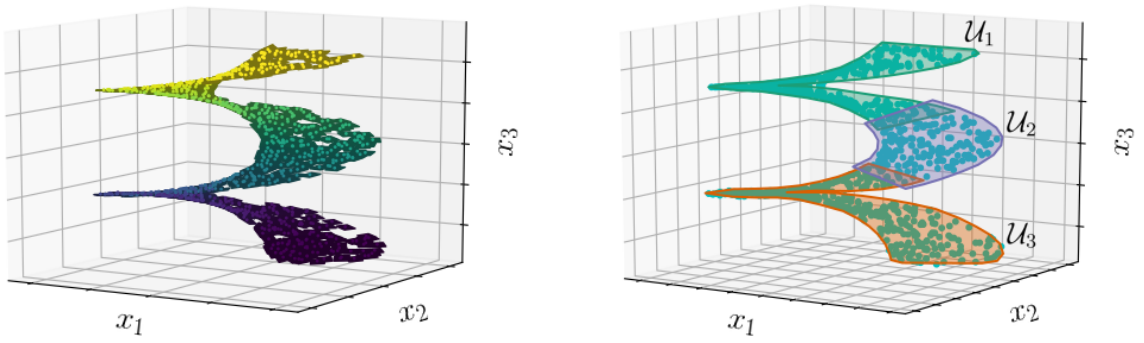
Theorem 5.4.1 (Existence of local reconstruction functions)**.** *Let $f : \mathcal{X} \to \mathcal{M}$ and $g : \mathcal{X} \to \mathcal{N}$ be smooth maps of manifolds $\mathcal{X}, \mathcal{M}, \mathcal{N}$ such that for every $x \in \mathcal{X}$, we have*

$$\ker \mathrm{D}\, f(x) \subset \ker \mathrm{D}\, g(x). \tag{5.58}$$

Then for every $x_0 \in \mathcal{X}$ there is an open neighborhood $\mathcal{U} \subset \mathcal{X}$ containing x_0 and a smooth function $h : f(\mathcal{U}) \to \mathcal{N}$ such that

$$g(x) = h \circ f(x) \qquad \forall x \in \mathcal{U}. \tag{5.59}$$

Proof. We use the rank theorem (Theorem 4.12 on p.81 in J. M. Lee [149]) and the fundamental theorem of calculus to show that g is constant on pre-image sets of f restricted to sufficiently small neighborhoods. We give the detailed proof in Appendix 5.A. □



(a) tangent planes

(b) neighborhoods

Figure 5.4.1: A spiral "parking garage" manifold where it is possible to reconstruct states locally in each tangent plane by measuring only x_1 and x_2, but it is not possible to reconstruct the full state from these measurements. It is possible to cover the manifold by neighborhoods on which the state can be reconstructed from x_1 and x_2.

It is important to note that local conditions based on solvability of linearized reconstruction problems cannot guarantee that the relevant quantities can be reconstructed globally. That is, a reconstruction function may only exist when the states are restricted to sufficiently small neighborhoods. We illustrate such a case in Figure 5.4.1 where the state space $\mathcal{X}$ is a spiral parking garage manifold in $\mathbb{R}^3$. We are interested in reconstructing the full state on this manifold and select measurements from among the coordinate functions $\boldsymbol{m}_1(\boldsymbol{x}) = x_1, \boldsymbol{m}_1(\boldsymbol{x}) = x_2$, and $\boldsymbol{m}_1(\boldsymbol{x}) = x_3$. The linearized problem consists of reconstructing states in each tangent plane to the spiral manifold shown in Figure 5.4.1a. This can be accomplished by measuring only $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) = (x_1, x_2)$. On the other hand, there are multiple "levels" of the parking garage that correspond to the same coordinates x_1, x_2. Hence, these coordinates are not sufficient to reconstruct the full state. This ambiguity is eliminated by restricting our attention to any one of the neighborhoods $\mathcal{U}_1$, $\mathcal{U}_2$, or $\mathcal{U}_3$ covering $\mathcal{X}$ in Figure 5.4.1b. That is, the state in any $\mathcal{U}_i$ can be reconstructed by measuring the coordinates x_1, x_2.

5.4.1 Convex optimization approaches

The convex optimization methods for linear measurement selection problems discussed by S. Joshi and S. Boyd in [125] extend to collections of linearized measurement selection problems near a collection of states. Let $\mathcal{X}_N = \{x_1, \ldots, x_N\} \subset \mathcal{X}$ be a collection of sampled states and define the linearized target variables $T_i = \mathrm{D}\,\boldsymbol{g}(x_i)$ and measurement functions $M_{i,\mathcal{S}} = \mathrm{D}\,\boldsymbol{m}_{\mathcal{S}}(x_i)$ about the sampled states. Our goal will be to choose a collection of sensors $\mathcal{S}$ so that the linearized target variables can be reconstructed from the linearized measurements at each x_i, i.e., there is a linear

map $\Phi_{i,\mathcal{S}}$ such that

$$T_i = \Phi_{i,\mathcal{S}} M_{i,\mathcal{S}}, \qquad \text{for every} \qquad i = 1, \ldots, N. \tag{5.60}$$

Without a prior distribution for tangent vectors in $T_{x_i}\mathcal{X}$, we may consider the average square error of a maximum likelihood linear estimator given by

$$J_i(\boldsymbol{s}) = \operatorname{Tr}\Big[T_i\Big(\sum_{j=1}^{M} s_j M_{i,j}^* \boldsymbol{C}_{\boldsymbol{n}_j}^{-1} M_{i,j}\Big)^{-1} T_i^*\Big], \qquad \boldsymbol{s} = (s_1, \ldots, s_M) \tag{5.61}$$

or the log-determinant of the reconstruction error covariance matrix

$$J_i(\boldsymbol{s}) = \log\det\Big[\boldsymbol{U}_i^* T_i\Big(\sum_{j=1}^{M} s_j M_{i,j}^* \boldsymbol{C}_{\boldsymbol{n}_j}^{-1} M_{i,j}\Big)^{-1} T_i^* \boldsymbol{U}_i\Big], \qquad \boldsymbol{s} = (s_1, \ldots, s_M), \tag{5.62}$$

where the columns of $\boldsymbol{U}_i$ provide an orthonormal basis for Range(T_i). We can find the sensors that maximize the average performance over the operating conditions x_i on a fixed sensor budget $|\mathcal{S}| \leq K$ by solving the relaxed convex optimization problem

$$\boxed{\underset{\boldsymbol{s}\in[0,1]^M}{\text{minimize}} \;\; \frac{1}{N}\sum_{i=1}^{N} J_i(\boldsymbol{s}) \qquad \text{s.t.} \qquad \sum_{i=1}^{M} s_i \leq K.} \tag{5.63}$$

Alternatively, we may have a specific level of desired performance $J_i(\boldsymbol{s}) \leq E_i$ at each sensor location, in which case we seek the minimum number of sensors needed to achieve this performance by solving the relaxed convex optimization problem

$$\boxed{\underset{\boldsymbol{s}\in[0,1]^M}{\text{minimize}} \;\; \sum_{i=1}^{M} s_i \qquad \text{s.t.} \qquad J_i(\boldsymbol{s}) \leq E_i, \quad i = 1, \ldots, N.} \tag{5.64}$$

5.4.2 Simultaneous QR pivoting

In the common special case in which we are seeking a collection of coordinate functions to reconstruct the full state (restricted to neighborhoods), we may perform the selection using an efficient greedy algorithm related to pivoted QR (PQR) factorization. This approach enables a generalization of DEIM (see Section 5.1.2) for building reduced-order models on nonlinear manifolds by choosing sample locations or state variables that can reconstruct the system's time derivative at each point on the underlying manifold. Rather than using pivoted QR factorization as in Eq. 5.21 to identify sensors based on a single POD subspace, the simultaneously pivoted QR factorization (SimPQR) method we present in [193]* identifies sensors based on a collection of subspaces simultaneously.

These subspaces may be tangent to an underlying manifold at a collection of sample points, or represent POD subspaces at a collection of operating conditions.

Let $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N \in \mathbb{R}^{n \times r}$ be a collection of matrices whose columns are orthonormal bases for the relevant subspaces at sample locations. Let $\boldsymbol{P}_{\mathcal{S}}$ denote the submatrix of the $n \times n$ identity matrix formed by the columns with indices in the set $\mathcal{S}$. In [193]* we seek a fixed set of sensors of the form

$$\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) = \boldsymbol{P}_{\mathcal{S}}^T \boldsymbol{x}, \tag{5.65}$$

that enable robust reconstruction in each subspace Range($\boldsymbol{U}_i$). In particular, we select $\mathcal{S}$ in such a way that every $\boldsymbol{P}_{\mathcal{S}}^T \boldsymbol{U}_1, \ldots, \boldsymbol{P}_{\mathcal{S}}^T \boldsymbol{U}_N$ has a left inverse with low amplification (operator norm). We select $\boldsymbol{P}_{\mathcal{S}}$ as in Eq. 5.21 using the leading columns of the permutation matrix $\boldsymbol{P} = \left[\begin{array}{c|c} \boldsymbol{P}_{\mathcal{S}} & \boldsymbol{P}_{\mathcal{S}^c} \end{array}\right]$ used to construct the SimPQR factorization

$$\boxed{\begin{bmatrix} \boldsymbol{U}_1^T \\ \vdots \\ \boldsymbol{U}_N^T \end{bmatrix} \left[\begin{array}{c|c} \boldsymbol{P}_{\mathcal{S}} & \boldsymbol{P}_{\mathcal{S}^c} \end{array}\right] = \begin{bmatrix} \boldsymbol{Q}_1 & & \\ & \ddots & \\ & & \boldsymbol{Q}_N \end{bmatrix} \left[\begin{array}{c|c} \boldsymbol{R}_1^{(1)} & \boldsymbol{R}_1^{(2)} \\ \vdots & \vdots \\ \boldsymbol{R}_N^{(1)} & \boldsymbol{R}_N^{(2)} \end{array}\right],} \tag{5.66}$$

The pivoting procedure aims to maximize the determinants of $r \times r$ upper-triangular sub-matrices $\tilde{\boldsymbol{R}}_i^{(1)}$ of each $\boldsymbol{R}_i^{(1)}$, which, thanks to Proposition 2.1 in [193]*, corresponds to minimizing an upper bound on the amplification

$$\|\boldsymbol{P}_{\mathcal{S}}^T \boldsymbol{U}_i\|_2 = \sigma_{\max}(\boldsymbol{P}_{\mathcal{S}}^T \boldsymbol{U}_1) \leq \frac{1}{|\det(\tilde{\boldsymbol{R}}_i^{(1)})|}. \tag{5.67}$$

During each step, the SimPQR pivoting procedure selects a sensor or pivot column $j^* \in \mathcal{M}$ along with the subset $\mathcal{U}^* \subset \{\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N\}$ on which the pivot j^* will be used to update the QR factorization. We do not require that the pivot j^* be used to update the QR factorizations of every $\boldsymbol{U}_i^T$ because the pivot column j^* may be a bad choice for some of these matrices. If $\mathcal{S}_i \subset \mathcal{S}$ denotes the set of pivot columns used in the factorization of $\boldsymbol{U}_i^T$, then we obtain the local pivoted QR factorizations

$$\boldsymbol{U}_i^T \left[\begin{array}{c|c|c} \boldsymbol{P}_{\mathcal{S}_i} & \boldsymbol{P}_{\mathcal{S} \setminus \mathcal{S}_i} & \boldsymbol{P}_{\mathcal{S}^c} \end{array}\right] = \boldsymbol{Q}_i \left[\begin{array}{c|c|c} \tilde{\boldsymbol{R}}_i^{(1)} & \tilde{\boldsymbol{R}}_i^{(2)} & \boldsymbol{R}_i^{(2)} \end{array}\right], \qquad i = 1, \ldots, N, \tag{5.68}$$

where $\tilde{\boldsymbol{R}}_i^{(1)}$ are the upper-triangular submatrices mentioned above.

In choosing the pivot column and the subset $\mathcal{U}$, there is a trade-off between reconstruction

robustness and choosing a small total number of sensors. If, on the one hand, we apply the selected pivot to the largest possible subset $\mathcal{U}$, then the factorization of every $\boldsymbol{U}_i^T$ will be completed using a small total number of pivots, i.e., sensors. On the other hand, if we apply the selected pivot to small collections $\mathcal{U}$ where it contributes the most to reconstruction robustness, then a large total number of pivots will be used to factorize every $\boldsymbol{U}_i^T$, i.e., a large total number of sensors. We manage this trade-off by solving an optimization problem for the pivot j^* and subset $\mathcal{U}^*$ selected during each step of SimPQR pivoting with a user-specified weighting $\gamma > 0$ that determines the relative importance of robustness and minimal sensing. The details of this optimization problem and how it can be efficiently solved are somewhat involved and can be found in [193]*. We give a brief summary here.

In ordinary pivoted QR factorization as described in G. H. Golub and C. F. Van Loan [97], we initialize a "redidual matrix" $\boldsymbol{A}$ with the matrix to be factored and keep track of the magnitudes of each column $c_j = \|\operatorname{col}_j(\boldsymbol{A})\|_2$. During the kth step of PQR, we select a pivot column j^* with the largest column magnitude c_{j^*}, which becomes the kth diagonal entry of the upper-triangular $\boldsymbol{R}$ matrix, i.e., $[\boldsymbol{R}]_{k,k} = c_{j^*}$. The remaining columns of $\boldsymbol{A}$ are orthogonalized with respect to the pivot column $\operatorname{col}_{j^*}(\boldsymbol{A})$ and the magnitudes $\boldsymbol{c}$ are updated. In SimPQR, we keep track of multiple residual matrices $\boldsymbol{A}_i$ initialized with $\boldsymbol{U}_i^T$ along with their column magnitudes $c_{i,j} = \|\operatorname{col}_j(\boldsymbol{A}_i)\|_2$. To inform our selection process for the pivot column j^* and the matrices $\mathcal{U}$ to which pivot is applied, we consider two extreme cases. On one hand, we could select a pivot column to maximize the number of matrices that can be factorized

$$U_{\max} = \max_{1 \le j \le n} \left| \{ i \in \{1, \dots, N\} \ : \ \|\operatorname{col}_j(\boldsymbol{A}_i)\|_2 > \varepsilon \} \right|, \tag{5.69}$$

where $\varepsilon > 0$ is a small constant. On the order hand, we could select a single matrix $\boldsymbol{A}_i$ and a pivot column j^* with maximum magnitude

$$C_i = \max_{1 \le j \le n} \|\operatorname{col}_j(\boldsymbol{A}_i)\|_2. \tag{5.70}$$

To manage the trade-off between these two extremes, we introduce a user-defined constant $\gamma > 0$ and solve the following optimization problem in [193]* for the pivot column and the subset of matrices to be factored during the kth step of SimPQR:

$$\boxed{(j^*, \mathcal{U}^*) = \operatorname*{argmax}_{\substack{1 \le j \le n, \\ \mathcal{U} \subset \{1, \dots, N\}}} \left(\frac{|\mathcal{U}|}{U_{\max}} + \gamma \min_{i \in \mathcal{U}} \frac{\|\operatorname{col}_j(\boldsymbol{A}_i)\|_2}{C_i} \right) \quad \text{s.t.} \quad \|\operatorname{col}_j(\boldsymbol{A}_i)\|_2 > \varepsilon \ \forall i \in \mathcal{U}.} \tag{5.71}$$

In [193]* we provide an efficient algorithm for solving Eq. 5.71 along with an efficient way to sweep out the complete set SimPQR factorizations, i.e., the "solution path" obtained for every value of γ. The following two propositions clarify the relationship between the user-defined parameter $\gamma > 0$ and the trade-off between robust and minimal pivoting.

Proposition 5.4.2 (SimPQR Robustness, Thm. 3.3 in [193]*)**.** *Let $\eta \in [0, 1)$. If $\gamma \geq \frac{N-1}{N(1-\eta)}$, then the solution of Eq. 5.71 satisfies*

$$\| \operatorname{col}_{j^*}(\boldsymbol{A}_i) \|_2 \geq \eta C_i \qquad \forall i \in \mathcal{U}^*. \tag{5.72}$$

Proof. See [193]*. □

Proposition 5.4.3 (SimPQR Minimality, Thm. 3.4 in [193]*)**.** *Let $\nu \in [0, 1)$. If $\gamma \leq 1 - \nu$, then the solution of Eq. 5.71 satisfies*

$$|\mathcal{U}^*| \geq \nu U_{max}. \tag{5.73}$$

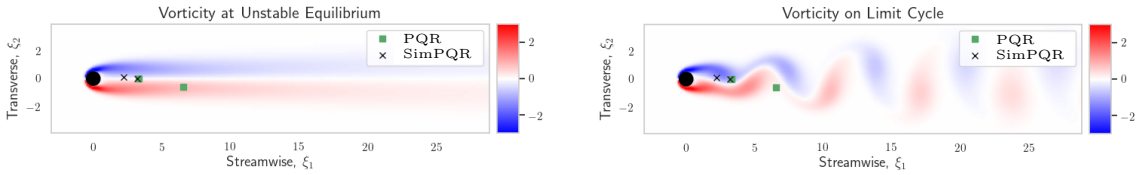
Moreover, if $\gamma \leq 1/N$, then $|\mathcal{U}^*| = U_{max}$.

Proof. See [193]*. □

In the limit $\gamma \to \infty$, SimPQR becomes equivalent to combining ordinary PQR factorizations of each $\boldsymbol{U}_i^T$. It should also be possible to extend SimPQR to vector measurements by working with SVDs of submatrices of each $\boldsymbol{U}_i^T$, but we shall not pursue this here.

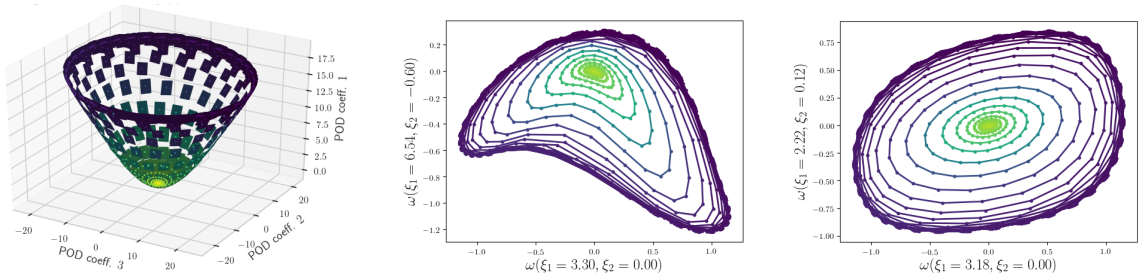
Among the examples in [193]*, we identify locations in the cylinder wake flow shown in Figure 5.4.2 at which to measure vorticity and reconstruct the system's state. This flow evolves along the two-dimensional unstable manifold at the unstable equilibrium up to a stable limit cycle. The tangent planes to this underlying manifold at a collection of points along the trajectory are shown in Figure 5.4.3a. We compare the SimPQR approach (with $\gamma = 1/N$ selected for minimal sensing) applied to a collection of local two-dimensional bases for each tangent plane against ordinary PQR using a global POD basis constructed from the snapshot data. The measurements made by the two sensors selected using each method are shown in Figure 5.4.3b and Figure 5.4.3c. Both sets of sensors provide an embedding of the underlying manifold. However, according to the distribution of reconstruction amplifications over the tangent planes plotted in Figure 5.4.4, the embedding provided by SimPQR is much more robust.

By expressing the tangent space bases $\boldsymbol{U}_k$ in the POD coordinate system, the SimPQR technique can be used to select the POD modes that enable local reconstruction of the state. In Figure 5.4.5,



(a) unstable equilibrium (b) limit cycle

Figure 5.4.2: Cylinder wake flow snapshots at the unstable equilibrium and on the stable limit cycle. The spatial sampling locations chosen by PQR on a global POD basis and SimPQR using a collection of local bases are also shown.



(a) tangent planes (b) PQR measurements (c) SimPQR measurements

Figure 5.4.3: Cylinder wake states along a trajectory plotted in different coordinate systems. In panel (a) we plot the states along with tangent planes to the underlying two-dimensional manifold in the leading 3 POD coordinates. Panels (b) and (c) show the vorticity measurements along the trajectory made by sensors chosen using PQR with a global POD basis and using SimPQR with bases for each tangent plane.

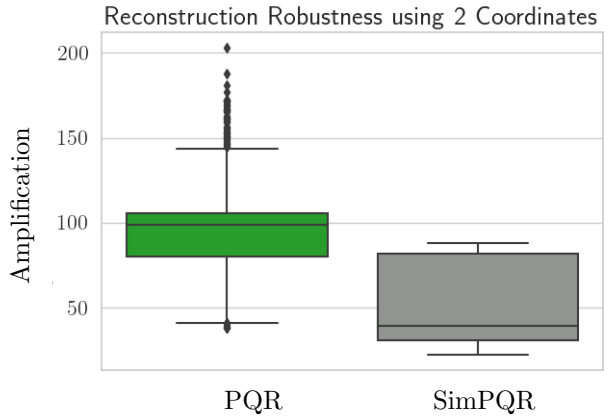
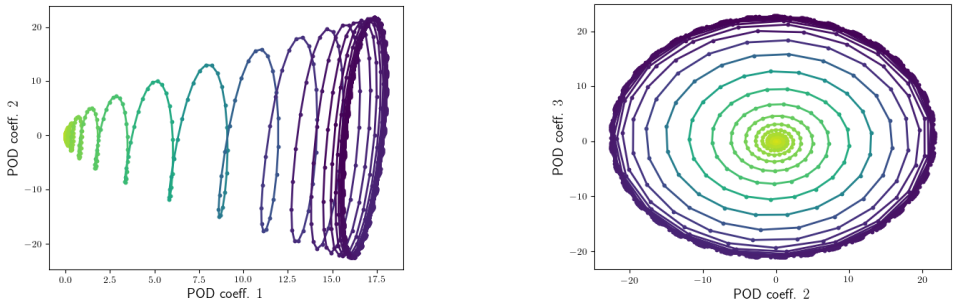


Figure 5.4.4: Distribution of maximum amplifications for reconstructing states in each tangent plane as measured by $1/\sigma_{\min}(\boldsymbol{P}_{\mathcal{S}}^T \boldsymbol{U}_i)$. We see that the coordinates chosen by SimPQR amplify disturbances less than those chosen by PQR on a global POD basis.



(a) Leading POD coordinates

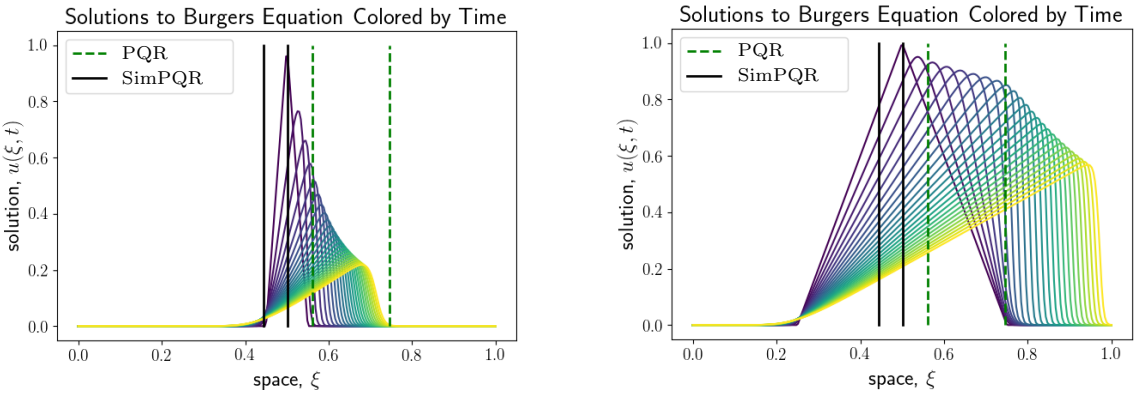
(b) SimPQR selection

Figure 5.4.5: Cylinder wake system trajectories plotted in the leading two POD coordinates as well as in the two POD coordinates selected using simultaneously pivoted QR (SimPQR) factorization. The POD coordinates selected by SimPQR provide an embedding of the underlying slow manifold on which the states lie, while the leading two POD coordinates do not.

we see that the leading two POD coefficients are insufficient to reconstruct the state of the cylinder wake flow — even locally. On the other hand, SimPQR factorization selects the second and third POD coefficients. Plotting the trajectory of the second and third POD coefficients in Figure 5.4.5b shows that these coordinates provide an embedding of the underlying slow manifold.

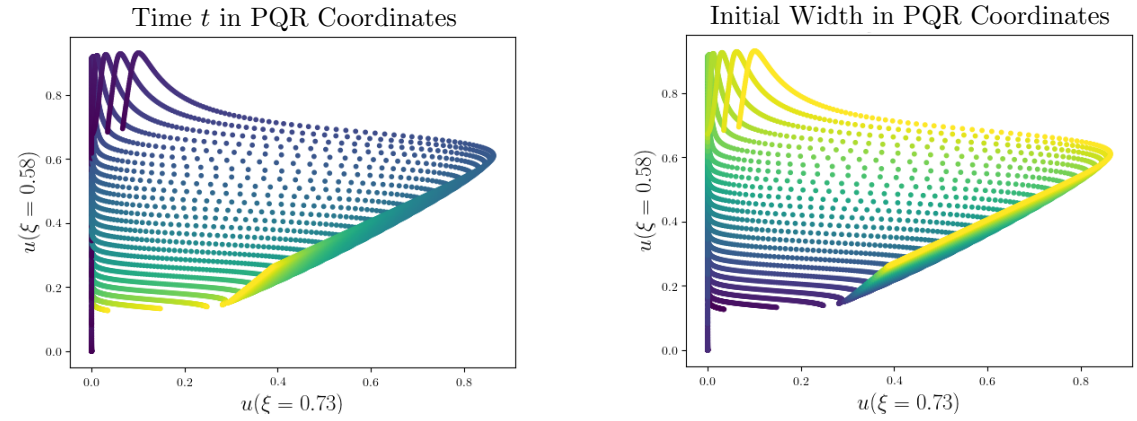
The advantages of simultaneous QR pivoting are made especially clear by considering another example, which is admittedly more contrived than the cylinder wake flow. In [193]*, we solve a viscous Burgers equation (a one-dimensional spatio-temporal PDE) from initial triangularly shaped profiles of varying widths. The evolution of the narrowest and widest profiles are shown in Figure 5.4.6. We observe that these states lie on a two-dimensional manifold parametrized by time and the width of the initial triangle. We aim to find a minimal collection of spatial sampling locations at which to measure the solution in order to recover the full state, i.e., to reconstruct the entire profile. Most of the variance of these solutions appears "down-stream" towards the right side of the spatial domain. Consequently the two sampling locations selected by DEIM with QR pivoting appear down-stream, yet are unable to reconstruct every profile, as one can see from the measurements shown in Figure 5.4.7.

On the other hand, we can approximate tangent planes to the underlying solution manifold by taking finite differences between profiles obtained at neighboring simulation times and from neighboring initial widths. Selecting the spatial sampling locations using SimPQR factorization of the resulting orthogonalized tangent space bases yields the "upstream" sampling locations shown in Figure 5.4.6. Remarkably the measurements from these sampling locations can be used to reconstruct every profile by inferring the corresponding simulation time and initial width as shown in Figure 5.4.8.



(a) (b)

Figure 5.4.6: Example simulations of the Burgers equation from the two extreme initial conditions. A collection of snapshots is plotted for each initial condition and colored according to the simulation time. The locations of the points selected by PQR on a global POD basis and SimPQR on local bases are indicated by the dashed green and solid black vertical lines respectively.



(a) (b)

Figure 5.4.7: Burgers equation snapshots plotted in the coordinates identified by PQR on a global POD basis and colored according to the time and initial width. We see that the underlying manifold folds back on itself when projected into these coordinates; hence the parametrizing coordinates are not single-valued functions of the coordinates selected by PQR.

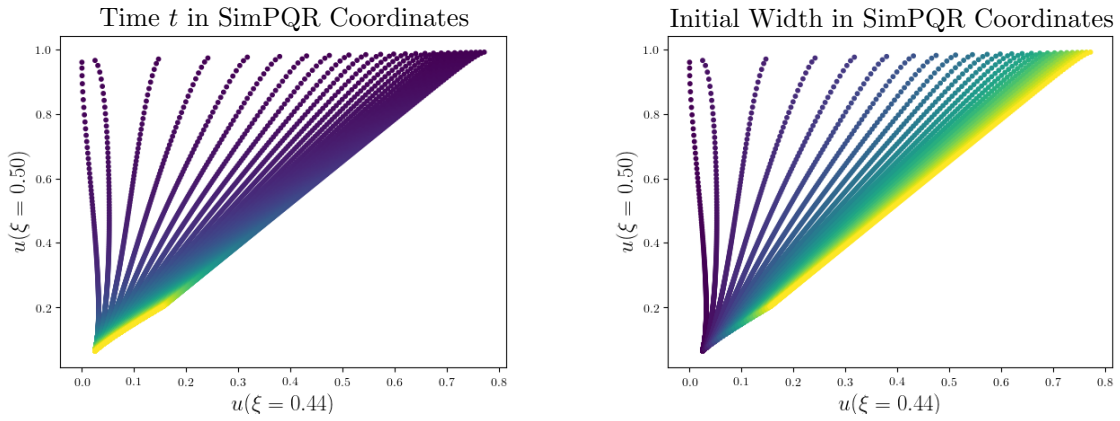


Figure 5.4.8: Burgers equation snapshots plotted in the coordinates identified by SimPQR factorization of local bases and colored according to the parametrizing coordinates time, t and initial width. We see that the parametrizing coordinates and hence the entire state are single-valued functions of the SimPQR coordinates.

5.4.3 Greedy performance for mean square error objectives

Two challenges arise when using greedy algorithms for sensor placement based on (local) least-squares objectives like Eq. 5.34. The first challenge is that Theorem 5.0.2 no longer provides a constant factor performance guarantee because the least-squares objective is not submodular (see Example 5.2.2). The second challenge is that even if we could obtain a similar constant factor performance guarantee, it would not translate into a particularly useful guarantee about the square error itself. It is important to overcome these challenges because greedy algorithms tend to be much more efficient and scalable to very large problems than the convex optimization approaches we described in Section 5.4.1. The solution we propose in this section involves applying modern non-submodular greedy performance guarantees to a logarithmic form of the square error objective.

Greedy optimization of non-submodular objectives including those arising from least-squares based sensor placement has been the subject of much research, including notable work by A. Das and D. Kempe in [72, 73] and A. A. Bian et al. in [24]. Here, the amount by which an optimization objective fails to be submodular is quantified, for instance using a "submodularity ratio" [72, 73] and/or "curvature" [24]. This yields generalized performance guarantees resembling the classical results of G. L. Nemhauser and L. A. Wolsey et al. [187, 283]. Bounds for these quantities (submodularity ratio and curvature) are usually obtained using arguments based on concavity properties of the objective. Because the greedy algorithm is so efficient, we shall focus on making a posteriori guarantees about the regret of greedy solutions — namely, how much worse is a greedy solution in

comparison with the true optimum we would find through an exhaustive search. In doing so, we can still recover a priori guarantees on the greedy performance that are similar to the ones involving the submodularity ratio [72, 73].

Constant-factor approximation for the naive square error objective in Eq. 5.34 does not always produce useful bounds on the square error of the optimal linear estimator. To see why, we denote the square error using the sensors $\mathcal{S}$ to construct an optimal linear estimator by $E(\mathcal{S}) = \operatorname{Tr} \boldsymbol{C}_e(\mathcal{S})$ (see Proposition 5.2.1). Recall that our greedy objective in Eq. 5.34 can be written as

$$\boxed{f(\mathcal{S}) = E(\emptyset) - E(\mathcal{S}), \qquad \text{where} \qquad E(\mathcal{S}) = g\left(\bar{\boldsymbol{P}}(\mathcal{S})\right) = \operatorname{Tr}\left(\boldsymbol{T}\bar{\boldsymbol{P}}(\mathcal{S})^{-1}\boldsymbol{T}^T\right).} \tag{5.74}$$

If $\mathcal{S}_K^*$ minimizes E under the sensor budget constraint $|\mathcal{S}| \leq K$ and $\mathcal{S}_K$ is a greedy solution, then a constant factor approximation guarantee of the form

$$f(\mathcal{S}_K) \geq C f(\mathcal{S}_K^*) \tag{5.75}$$

for some $C \in (0, 1)$ implies the bound

$$E(\mathcal{S}_K) \leq (1 - C)E(\emptyset) + CE(\mathcal{S}_K^*). \tag{5.76}$$

In the submodular case $C = 1 - e^{-1} \approx 0.63$, and we cannot hope to do better in the non-submodular case. At best Eq. 5.76 ensures that error can be reduced by a factor of $(1 - C)$ compared to the case with no sensors at all, and we would ideally hope to reduce the error to a value much less than $e^{-1} \approx 0.37$ of the error with no sensors. Said another way, if the greedily chosen sensors reduce the square error to less than $(1 - C)$ of the error with no sensors, then the best possible sensors $\mathcal{S}_K^*$ might reduce the error all the way to zero. Therefore, the constant factor approximation guarantee is essentially useless as an a posteriori bound when the sensors we select greedily achieve high performance.

It would be much better to obtain a constant factor approximation bound for an objective function using the log square error

$$\boxed{f(\mathcal{S}) = \log E(\emptyset) - \log E(\mathcal{S}) = -\log\left(\frac{E(\mathcal{S})}{E(\emptyset)}\right).} \tag{5.77}$$

Now, a constant factor guarantee in the form of Eq. 5.75 would yield

$$E(\mathcal{S}_K) \leq \exp\left[(1-C)\log E(\emptyset) + C\log E(\mathcal{S}_K^*)\right], \tag{5.78}$$

which, if the constant C remains unchanged, is strictly better than Eq. 5.76 thanks to the convexity of the exponential function. Moreover, Eq. 5.78 ensures that the error using the greedily chosen sensors $E(\mathcal{S}_K)$ approaches zero if the error using the optimal sensors $E(\mathcal{S}_K^*)$ approaches zero. Stated another way, if we greedily identify a set of sensors $\mathcal{S}_K$ achieving a very low error $E(\mathcal{S}_K)$, then Eq. 5.78 provides us with a non-trivial lower bound on the minimum possible error

$$\boxed{E(\mathcal{S}_K^*) \geq \exp\left[\frac{1}{C}\log E(\mathcal{S}_K) - (\frac{1}{C}-1)\log E(\emptyset)\right],} \tag{5.79}$$

that is, a bound on how much we may regret using the greedily chosen sensors. In practice, greedy approximation factors for the original and logarithmic square error objectives may be different, in which case, both the original and logarithmic bounds may be superior in different regimes. Regardless of the difference in greedy approximation factors, as long as the factor for the logarithmic objective Eq. 5.77 is nonzero, the bound in Eq. 5.79 will be superior in the limit of small error $E(\mathcal{S}_K) \to 0$.

Maximizing the logarithmic objective in Eq. 5.77 is equivalent to maximizing the original square error objective in Eq. 5.74 because Eq. 5.77 is a monotone increasing function of Eq. 5.74. Hence, greedy algorithms for Eq. 5.74 and Eq. 5.77 select precisely the same sequence of sensors. Both objectives are not submodular, and so we must resort to non-submodular constant factor approximation guarantees such as those proved by A. A. Bian et al. [24] for Eq. 5.77 in order to arrive at the superior bound in Eq. 5.79. In particular, A. A. Bian et al. defines the Greedy submodularity ratio and curvature as follows.

Definition 5.4.4 (greedy submodularity ratio [24])**.** *The greedy submodularity ratio is the largest scalar γ_G such that*

$$\sum_{\omega \in \Omega \setminus \mathcal{S}_k} \left[f(\mathcal{S}_k \cup \{\omega\}) - f(\mathcal{S}_k)\right] \geq \gamma_G \left[f(\mathcal{S}_k \cup \Omega) - f(\mathcal{S}_k)\right] \tag{5.80}$$

for every subset $\Omega \subset \mathcal{M}$ of size $|\Omega| = K$ and every $k = 0, \ldots, K-1$.

Definition 5.4.5 (greedy curvature [24])**.** *The greedy curvature is the smallest scalar $\alpha_G \geq 0$ such that*

$$f(\mathcal{S}_k \cup \Omega) - f(\mathcal{S}_{k-1} \cup \Omega) \geq (1-\alpha_G)\left[f(\mathcal{S}_k) - f(\mathcal{S}_{k-1})\right] \tag{5.81}$$

for every subset $\Omega \subset \mathcal{M} \setminus \{j_k\}$ *of size* $|\Omega| = K$ *and every* $k = 1, \ldots, K$.

Using these definitions, Theorem 1 in [24] shows that the greedy algorithm maximizing f has the following constant factor approximation guarantee

$$\boxed{f(\mathcal{S}_K) \geq \frac{1}{\alpha_{\mathrm{G}}} \left(1 - e^{-\alpha_{\mathrm{G}}\gamma_{\mathrm{G}}}\right) f(\mathcal{S}_K^*) \geq \left(1 - e^{-\gamma_{\mathrm{G}}}\right) f(\mathcal{S}_K^*).} \tag{5.82}$$

Relying on Definition 5.4.4 and Definition 5.4.5 to compute the greedy submodularity ratio and curvature of the logarithmic square error objective Eq. 5.77 requires us to perform exhaustive searches over every subset $\Omega \subset \mathcal{M}$ of size $|\Omega| = K$. Since doing such a search is as computationally costly as simply computing the optimal set $\mathcal{S}_K^*$ maximizing the objective, we must bound the submodularity ratio and curvature by quantities that don't involve Ω. To construct such bounds, we rely on a useful result by T. Ando and F. Hiai (Proposition 1.1 in [9]). By this result, the function $g : \boldsymbol{P} \mapsto \mathrm{Tr}(\boldsymbol{T}\boldsymbol{P}^{-1}\boldsymbol{T}^T)$ defined for positive-definite matrices $\boldsymbol{P}$ in Eq. 5.74 is operator log-convex, i.e.,

$$\log g((1-\lambda)\boldsymbol{P}_0 + \lambda \boldsymbol{P}_1) \leq (1-\lambda)\log g(\boldsymbol{P}_0) + \lambda \log g(\boldsymbol{P}_1) \qquad \forall \lambda \in [0,1] \qquad \forall \boldsymbol{P}_0, \boldsymbol{P}_1 \succ \boldsymbol{0}. \tag{5.83}$$

This follows by the operator monotonicity of matrix inversion (Lemma 5.A.1 in Appendix 5.A). For multiple linearized objectives defined by $g_i : \boldsymbol{P}_i \mapsto \mathrm{Tr}(\boldsymbol{T}_i \boldsymbol{P}_i^{-1} \boldsymbol{T}_i^T)$, $i = 1, \ldots N$, the same argument can be used to show that

$$g(\boldsymbol{P}_1, \ldots, \boldsymbol{P}_N) = \frac{1}{N}\sum_{i=1}^{N} g_i(\boldsymbol{P}_i) = \mathrm{Tr}\left(\begin{bmatrix} \boldsymbol{T}_1 & & \\ & \ddots & \\ & & \boldsymbol{T}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_1 & & \\ & \ddots & \\ & & \boldsymbol{P}_N \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{T}_1 & & \\ & \ddots & \\ & & \boldsymbol{T}_N \end{bmatrix}^T\right) \tag{5.84}$$

is operator log-convex. Here, we recall that

$$E(\mathcal{S}) = g(\bar{\boldsymbol{P}}_1(\mathcal{S}), \ldots, \bar{\boldsymbol{P}}_N(\mathcal{S})) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{Tr}(\boldsymbol{T}_i \bar{\boldsymbol{P}}_i(\mathcal{S})^{-1} \boldsymbol{T}_i^T), \qquad \bar{\boldsymbol{P}}_i(\mathcal{S}) = \bar{\boldsymbol{P}}_i(\emptyset) + \sum_{j \in \mathcal{S}} \boldsymbol{P}_i(\{j\}) \tag{5.85}$$

is the mean square error of optimal linear estimators across the operating conditions $i = 1, \ldots, N$. The log-convexity of Eq. 5.84 immediately yields upper and lower bounds, summarized in Lemma 5.4.6, for the differences in the logarithmic objective between any two subsets of sensors. Lemma 5.4.6 includes the case $N = 1$ where the objective is based on the average square error for a single optimal linear estimator.

Lemma 5.4.6 (Modular sandwich for logarithmic square error)**.** *Consider the logarithmic objective* $f(\mathcal{S}) = \log E(\emptyset) - \log E(\mathcal{S})$ *with the composite mean square error given by Eq. 5.85 and let*

$$\boldsymbol{G}_i(\mathcal{S}) = -\nabla g_i(\bar{\boldsymbol{P}}_i(\mathcal{S})) = \bar{\boldsymbol{P}}_i(\mathcal{S})^{-1}\boldsymbol{T}_i^T\boldsymbol{T}_i\bar{\boldsymbol{P}}_i(\mathcal{S})^{-1}, \qquad i = 1, \dots, N. \tag{5.86}$$

Then, for any $\mathcal{S}, \mathcal{S}' \subset \mathcal{M}$*, we have the following two-sided bound*

$$\frac{1}{N}\sum_{i=1}^{N} \operatorname{Tr}\left[\frac{\boldsymbol{G}_i(\mathcal{S}')}{E(\mathcal{S}')}\left(\boldsymbol{P}_i(\mathcal{S}') - \boldsymbol{P}_i(\mathcal{S})\right)\right] \leq f(\mathcal{S}') - f(\mathcal{S}) \leq \frac{1}{N}\sum_{i=1}^{N} \operatorname{Tr}\left[\frac{\boldsymbol{G}_i(\mathcal{S})}{E(\mathcal{S})}\left(\boldsymbol{P}_i(\mathcal{S}') - \boldsymbol{P}_i(\mathcal{S})\right)\right], \tag{5.87}$$

where we recall that $\boldsymbol{P}_i(\mathcal{S}') - \boldsymbol{P}_i(\mathcal{S}) = \sum_{j\in\mathcal{S}'\setminus\mathcal{S}} \boldsymbol{P}_i(\{j\}) - \sum_{j\in\mathcal{S}\setminus\mathcal{S}'} \boldsymbol{P}_i(\{j\})$ *is modular.*

Proof. We use the operator log-convexity result by T. Ando and F. Hiai (Proposition 1.1 in [9]) to bound Eq. 5.84 from below using its derivative. Bounding $E(\mathcal{S}')$ from below using the derivative evaluated at $\mathcal{S}$ yields the upper bound, while the lower bound is obtained by bounding $E(\mathcal{S})$ from below using the derivative evaluated at $\mathcal{S}'$. □

The differences between the values of the objective function appearing in Definition 5.4.4 and Definition 5.4.5 of the greedy submodularity ratio and curvature can now be bounded from both sides using Lemma 5.4.6. In particular, by reducing the differences involving the variable subset Ω into sums over its elements, we can obtain non-trivial bounds for the greedy submodularity ratio and curvature that do not require an exhaustive search over all subsets $\Omega \subset \mathcal{M}$. For instance, the greedy submodularity ratio is bounded in the following result.

Proposition 5.4.7 (bound on greedy submodularity ratio for logarithmic objective)**.** *Define the quantities*

$$\lambda_k = \min_{1\leq i\leq N}\left(\lambda_{min}\left[\bar{\boldsymbol{P}}_i(\emptyset)\right] + \min_{\substack{\mathcal{S}\subset\mathcal{M}\\|\mathcal{S}|=k}}\sum_{a\in\mathcal{S}}\lambda_{min}\left[\boldsymbol{P}_i(\{a\})\right]\right) \tag{5.88}$$

$$\Lambda_k = \max_{1\leq i\leq N}\left(\min\left\{\lambda_{max}\left[\bar{\boldsymbol{P}}_i(\emptyset)\right] + \max_{\substack{\mathcal{S}\subset\mathcal{M}\\|\mathcal{S}|=k}}\sum_{a\in\mathcal{S}}\lambda_{max}\left[\boldsymbol{P}_i(\{a\})\right],\ \lambda_{max}\left[\bar{\boldsymbol{P}}_i(\mathcal{M})\right]\right\}\right), \tag{5.89}$$

which may be computed a priori using a sorting procedure. Then the greedy submodularity ratio given in Definition 5.4.4 for the logarithmic objective Eq. 5.77 using the composite square error Eq. 5.85 is bounded below by

$$\gamma_G \geq \min_{\substack{0\leq k\leq K-1\\a\in\mathcal{M}}}\left(\frac{E(\mathcal{S}_k)}{E(\mathcal{S}_k\cup\{a\})}\right)\left(\frac{\sum_{i=1}^{N}\operatorname{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k\cup\{a\})\boldsymbol{P}_i(\{a\})\right]}{\sum_{i=1}^{N}\operatorname{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k)\boldsymbol{P}_i(\{a\})\right]}\right) \geq \min_{0\leq k\leq K-1}\left(\frac{\lambda_k}{\Lambda_{k+1}}\right)^2 \tag{5.90}$$

Proof. We rely primarily on Lemma 5.4.6 to bound the numerator and denominator in the definition of the greedy submodularity ratio. We provide the detailed proof in Appendix 5.A. □

A similar approach can be used to show that the greedy curvature is bounded above by

$$\alpha_{\mathrm{G}} \leq 1 - \min_{1\leq k\leq K} \left(\frac{E(\mathcal{S}_{k-1})}{E(\mathcal{S}_k)}\right) \frac{\sum_{i=1}^{N} \beta_i(\mathcal{S}_k)^{-2} \operatorname{Tr}\left[\boldsymbol{T}_i \boldsymbol{P}_i(\{j_k\}) \boldsymbol{T}_i^T\right]}{\sum_{i=1}^{N} \lambda_{\min}\left[\bar{\boldsymbol{P}}_i(\mathcal{S}_{k-1})\right]^{-2} \operatorname{Tr}\left[\boldsymbol{T}_i \boldsymbol{P}_i(\{j_k\}) \boldsymbol{T}_i^T\right]}$$
$$\leq 1 - \min_{0\leq k\leq K-1} \left(\frac{\lambda_k}{\Lambda_{k+1}}\right)^2, \quad (5.91)$$

where

$$\beta_i(\mathcal{S}) = \lambda_{\max}\left[\bar{\boldsymbol{P}}_i(\mathcal{S})\right] + \max_{\substack{\mathcal{A}\subset\mathcal{M}\setminus\mathcal{S} \\ |\mathcal{A}|\leq K}} \sum_{a\in\mathcal{A}} \lambda_{\max}\left[\boldsymbol{P}_i(\{a\})\right] \quad (5.92)$$

is easily computed by a sorting procedure. Using this method, we can obtain a posteriori or (pessimistic) a priori bounds for the greedy approximation factor appearing in Eq. 5.82. This approximation factor then yields non-trivial bounds on the mean square error using Eq. 5.79 or Eq. 5.78.

5.4.4 A modified greedy algorithm for non-submodular objectives

Another way to handle non-submodular objectives, such as the square error objective for an optimal linear estimator, is to modify the greedy algorithm itself to improve the resulting bound on performance. In this section, we describe a greedy algorithm that is based on maximizing a submodular upper bound on a non-submodular objective and bounding the performance based on how much the submodular approximation over-estimates the original objective. Because linearization automatically provides a modular upper bound, this approach is especially well-suited to objectives that have concave extensions on the cube $[0,1]^M$. The linearized objective can be also be modified, e.g., by composing with a concave function to improve the approximation of the original objective, while remaining submodular. For example, in set-covering problems (see Section 5.3.2), we often work with composite objectives where each component is truncated at the desired level of performance. If the components have concave extensions, then a submodular upper bound can be obtained by truncating the linearization of each component. We show that maximizing a submodular upper bound at each step instead of the original objective allows us to bound the performance directly in terms of the gap between the objectives. This gap is often easy to compute in practice a posteriori and to bound a priori. This may be advantageous compared to greedily maximizing the original objective because existing performance guarantees for non-submodular objectives, such as those in

A. A. Bian et al. [24], depend on quantities like the submodularity ratio and curvature that are difficult to compute in practice and admit only highly pessimistic bounds.

We consider a general set function $f : 2^{\mathcal{M}} \to \mathbb{R}$ (not necessarily the ones given by Eq. 5.34 or Eq. 5.77) and we try to solve the optimization problem

$$\underset{\mathcal{S}\subseteq\mathcal{M}}{\text{maximize}} \quad f(\mathcal{S}) \quad \text{s.t.} \quad c(\mathcal{S}) := \sum_{a\in\mathcal{S}} c(\{a\}) \leq C. \tag{5.93}$$

Here, $c(\mathcal{S})$ represents the cost of the sensors, which allows us to work with the case when some sensors are more expensive than others. In most cases, however, $c(\mathcal{S}) = |\mathcal{S}|$ simply counts the total number of sensors. We always assume that the objective function f satisfies the following

Assumption 5.4.8. *The objective function $f : 2^{\mathcal{M}} \to \mathbb{R}$ is normalized so that $f(\emptyset) = 0$ and monotone increasing, i.e., that*

$$\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{M} \quad \Rightarrow \quad f(\mathcal{S}) \leq f(\mathcal{S}'). \tag{5.94}$$

We will also always assume that there are no free measurements, i.e., $c(\mathcal{S}) > 0$ for all non-empty $\mathcal{S} \subseteq \mathcal{M}$.

The objective function in Eq. 5.34 based on the mean square error of the optimal linear estimator as well as its logarithmic version Eq. 5.77 satisfy all conditions of assumption 5.4.8. In particular, normalization is trivial and monotonicity follows from the Loewner order-reversing property of the matrix inverse shown in Lemma 5.A.1 of Appendix 5.A.

Given the possibly non-submodular objective function f, we define

Definition 5.4.9 (Submodular Upper Bound)**.** *A submodular upper bound for a set function $f : 2^{\mathcal{M}} \to \mathbb{R}$ at $\mathcal{S} \subseteq \mathcal{M}$ is a set function $\hat{f}_{\mathcal{S}} : 2^{\mathcal{M}\setminus\mathcal{S}} \to \mathbb{R}$ with the following properties:*

1. $\hat{f}_{\mathcal{S}}(\emptyset) = f(\mathcal{S})$ *(normalization)*
2. $\hat{f}_{\mathcal{S}}(\mathcal{A}) \geq f(\mathcal{S}\cup\mathcal{A})\ \forall \mathcal{A} \subseteq \mathcal{M}\setminus\mathcal{S}$ *(upper bound)*
3. $\mathcal{A} \subseteq \mathcal{A}' \subseteq (\mathcal{M}\setminus\mathcal{S})\setminus\{a\} \ \Rightarrow\ \hat{f}_{\mathcal{S}}(\mathcal{A}\cup\{a\}) - \hat{f}_{\mathcal{S}}(\mathcal{A}) \geq \hat{f}_{\mathcal{S}}(\mathcal{A}'\cup\{a\}) - \hat{f}_{\mathcal{S}}(\mathcal{A}')$ *(submodularity)*

For ease of notation, we also define $h_{\mathcal{S}}(\mathcal{A}) = \hat{f}_{\mathcal{S}}(\mathcal{A}) - f(\mathcal{S})$.

To quantify the amount by which the submodular upper bound over-estimates the original objective f, we define

Definition 5.4.10 (Submodularity Gap Ratio)**.** *The submodularity gap ratio associated with a nested sequence of sets* $\mathcal{S}_0 \subset \mathcal{S}_1 \subset \cdots \subset \mathcal{S}_K \subseteq \mathcal{M}$ *and corresponding submodular upper bounds is defined as*

$$\Gamma_K = \min_{0 \leq k < K} \frac{f(\mathcal{S}_{k+1}) - f(\mathcal{S}_k)}{h_{\mathcal{S}_k}(\mathcal{S}_{k+1} \setminus \mathcal{S}_k)}. \tag{5.95}$$

Rather than approximate the solution of Eq. 5.93 by greedy selection based on the objective f, we propose the Greedy Upper Selection (GUS) Algorithm 1, which performs greedy selection based on the subodular upper bound.

Algorithm 1 Greedy Upper Selection (GUS)

$\mathcal{S}_0 = \emptyset$
for $k = 1, 2, \ldots, K$ **do**
  $a_k = \operatorname{argmax}_{a \in \mathcal{M} \setminus \mathcal{S}_{k-1}} h_{\mathcal{S}_{k-1}}(\{a\})/c(\{a\})$
  $\mathcal{S}_k = \mathcal{S}_{k-1} \cup \{a_k\}$
end for
return $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K$

GUS has an approximation guarantee stated below in Theorem 5.4.11. This guarantee closely resembles the guarantee by A. Das and D. Kempe [72, 73] based on the submodularity ratio. In particular, the guarantees are identical when Γ_K is replaced by the submodularity ratio.

Theorem 5.4.11 (Greedy upper selection performance)**.** *Let* $\emptyset = \mathcal{S}_0 \subset \mathcal{S}_1 \subset \cdots \subset \mathcal{S}_K \subseteq \mathcal{M}$ *be the sequence of sets computed by Algorithm 1 and let* Γ_K *be the corresponding submodularity gap ratio. If* $\mathcal{S}^*$ *is the optimal solution of (5.93) then we have*

$$f(\mathcal{S}_k) > \left(1 - e^{-\Gamma_K c(\mathcal{S}_k)/C}\right) f(\mathcal{S}^*), \quad k = 1, \ldots, K. \tag{5.96}$$

Proof. We use a slight modification of the argument by G. L. Nemhauser et al. that we presented in the proof of Theorem 5.0.2. The details can be found in Appendix 5.A. □

Remark 5.4.12. *The proof of Theorem 5.4.11 only requires* $\hat{f}_{\mathcal{S}}(\mathcal{A})$ *to be a submodular upper bound for sets* $\mathcal{S} = \mathcal{S}_k$ *selected by GUS and* $\mathcal{A} \subset \mathcal{M} \setminus \mathcal{S}_k$ *with* $|\mathcal{A}| \leq K$ *where* $K = |\mathcal{S}^*|$.

Among the options for submodular upper bounds, there is always one one that can be used to reproduce the original greedy solution for the objective f as well as its performance guarantee based on the greedy submodularity ratio. In particular if γ_{G} is the greedy submodularity ratio given by Definition 5.4.9 and $\mathcal{S}_k$ are the greedily chosen sets, we may always define a modular upper bound

$\hat{f}_{\mathcal{S}_k}(\mathcal{A}) = f(\mathcal{S}_k) + h_{\mathcal{S}_k}(\mathcal{A})$ according to

$$h_{\mathcal{S}_k}(\mathcal{A}) = \frac{1}{\gamma_{\mathrm{G}}} \sum_{a \in \mathcal{A}} \left[f(\mathcal{S}_k \cup \{a\}) - f(\mathcal{S}_k) \right]. \tag{5.97}$$

Using this objective, we observe that GUS will select precisely the same sets $\mathcal{S}_k$ as the original greedy algorithm for the objective f. The function in Eq. 5.97 provides a submodular upper bound for every $\mathcal{A} \subset \mathcal{M} \setminus \mathcal{S}_k$ with $|\mathcal{A}| \leq K$ by definition of γ_{G}. Hence, thanks to Remark 5.4.12, Theorem 5.4.11 holds for GUS using this objective. Moreover, we have $\Gamma_K = \gamma_{\mathrm{G}}$ as a direct consequence of Definition 5.4.10. Therefore, the performance bound given by Theorem 5.4.11 for GUS using the submodular upper bound defined by Eq. 5.97 and cost $c(\mathcal{S}) = |\mathcal{S}|$ is the same as the bound $f(\mathcal{S}_k) \geq \left(1 - e^{-\gamma_{\mathrm{G}} k/K}\right) f(\mathcal{S}^*)$ using the greedy submodularity ratio in Eq. 5.82.

The main advantage of the submodularity gap ratio is that it can be computed exactly during execution of Algorithm 1 without increasing the computational or memory complexity. On the other hand, computing the (greedy) submodularity ratio (given by Definition 5.4.4) exactly would require an auxiliary combinatorial optimization step. Since this is computationally infeasible, one is forced into making highly pessimistic estimates, e.g. based on ratios of smallest and largest eigenvalues as in [72] and [24].

When we are trying to maximize the logarithmic square error objective in Eq. 5.77, we can use Lemma 5.4.6 to provide a submodular upper bound to be used by GUS. In particular, we define a submodular (in fact, modular) upper bound $\hat{f}_{\mathcal{S}}(\mathcal{A}) = f(\mathcal{S}) + h_{\mathcal{S}}(\mathcal{A})$ using Lemma 5.4.6 according to

$$h_{\mathcal{S}}(\mathcal{A}) = \frac{1}{NE(\mathcal{S})} \sum_{i=1}^{N} \sum_{a \in \mathcal{A}} \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}) \boldsymbol{P}_i(\{a\}) \right]. \tag{5.98}$$

The resulting submodularity gap ratio Γ_K can be computed directly during the execution of GUS simply be comparing the increment in the objective f to the increment predicted by the submodular upper bound for each of the selected elements. We also note that Lemma 5.4.6 yields a lower bound for the submodularity gap ratio using the objective in Eq. 5.98,

$$\Gamma_K \geq \min_{0 \leq k < K} \left(\frac{E(\mathcal{S}_k)}{E(\mathcal{S}_{k+1})} \right) \left(\frac{\sum_{i=1}^{N} \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_{k+1}) \boldsymbol{P}_i(\{j_{k+1}\}) \right]}{\sum_{i=1}^{N} \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k) \boldsymbol{P}_i(\{j_{k+1}\}) \right]} \right) \geq \min_{0 \leq k \leq K-1} \left(\frac{\lambda_k}{\Lambda_{k+1}} \right)^2, \tag{5.99}$$

where λ_k and Λ_{k+1} are defined as in Proposition 5.4.7 and may be computed a priori. We observe that the a posteriori lower bound given by the first inequality above has a superior form compared to the a posteriori lower bound for the greedy submodularity ratio (see Definition 5.4.4) given by

the first inequality of Eq. 5.90 in Proposition 5.4.7. However, the sets $\mathcal{S}_k$ above are the ones selected using GUS, which may be different from the greedily selected $\mathcal{S}_k$ appearing in Eq. 5.90.

Remark 5.4.13 (Computational advantages of the linearized upper bound Eq. 5.98)**.** *We note that optimizing Eq. 5.98 during each step of GUS is much less computationally costly than greedy optimization based on the original objective when the state space dimension n is large. In particular, each matrix $\boldsymbol{P}_i$ has dimension $n \times n$ and so computing each trace in Eq. 5.98 requires $\mathcal{O}(n^2)$ operations once $\boldsymbol{G}_i(\mathcal{S}_k)$ has been found. This must be done once for each candidate sensor in the collection $\mathcal{M} \setminus \mathcal{S}_k$. Computing $\boldsymbol{G}_i(\mathcal{S}_k)$ requires one-time work of $\mathcal{O}(n^3)$ during the kth step of GUS. On the other hand, evaluating the performance of each candidate sensor in the collection $\mathcal{M} \setminus \mathcal{S}_k$ using the original objective f requires $\mathcal{O}(n^3)$ operations because each $\bar{\boldsymbol{P}}_i(\mathcal{S}_k \cup \{a\})$ must be inverted.*

Our approach may also be used to approximate the solutions of non-submodular set-covering problems of the form

$$\underset{\mathcal{S} \subseteq \mathcal{M}}{\text{minimize}} \;\; c(\mathcal{S}) := \sum_{a \in \mathcal{S}} c(\{a\}) \quad \text{s.t.} \quad f(\mathcal{S}) = f(\mathcal{M}). \tag{5.100}$$

These problems arise, for instance, when we want to find the minimum number of sensors (or the cheapest collection of sensors) that achieve a specified level of performance at every operating condition. Suppose that the average square error of optimal linear estimators (see Proposition 5.2.1) using sensors $\mathcal{S}$ at a collection of operating conditions are given by

$$E_i(\mathcal{S}) = g_i(\bar{\boldsymbol{P}}_i(\mathcal{S})), \qquad g_i : \boldsymbol{P} \mapsto \operatorname{Tr}\left(\boldsymbol{T}_i \boldsymbol{P}^{-1} \boldsymbol{T}_i^T\right), \tag{5.101}$$

and we want to find $\mathcal{S} \subset \mathcal{M}$ of minimum cost $c(\mathcal{S})$ such that $E_i(\mathcal{S}) \leq \varepsilon_i$ for some thresholds $\varepsilon_i > 0$ at each $1 = 1, \ldots, N$. If it is possible to achieve $E_i(\mathcal{M}) \leq \varepsilon_i$ for each $1 = 1, \ldots, N$ using all of the sensors, then we can cast this problem in the form of Eq. 5.100 using the non-submodular objective

$$f(\mathcal{S}) = \frac{1}{N} \sum_{i=1}^{N} \min\left\{E_i(\emptyset) - E_i(\mathcal{S}),\; E_i(\emptyset) - \varepsilon_i\right\}. \tag{5.102}$$

To approximate the solution of Eq. 5.100, we construct a submodular upper bound (see Definition 5.4.9) for f that satisfies the following additional assumption:

Assumption 5.4.14. *We assume that the submodular upper bound for the objective function f in problem Eq. 5.100 shares the same maximum value, that is, it satisfies $\hat{f}_{\mathcal{S}}(\mathcal{A}) \leq f(\mathcal{M})$ for every $\mathcal{S} \subseteq \mathcal{M}$ and $\mathcal{A} \subseteq \mathcal{M} \setminus \mathcal{S}$.*

We then use the GUS Algorithm 1 to choose sensors maximizing f and stopping when $f(\mathcal{S}_K) = f(\mathcal{M})$ is achieved. Here, K is not specified before the algorithm is run. Theorem 5.4.15 says that GUS is guaranteed to achieve a certain minimum level of performance compared to the optimal solution of the non-submodular set covering problem in Eq. 5.100.

Theorem 5.4.15 (Near-Minimum Cost Selection)**.** *Let $\emptyset = \mathcal{S}_0 \subset \mathcal{S}_1 \subset \cdots \subset \mathcal{S}_K \subseteq \mathcal{M}$ be the sequence of sets computed by Algorithm 1 with $f(\mathcal{S}_{K-1}) < f(\mathcal{M})$ and $f(\mathcal{S}_K) = f(\mathcal{M})$. Let Γ_K be the corresponding submodularity gap ratio (see Definition 5.4.10) and let $\mathcal{S}^*$ be the optimal solution of Eq. 5.100 achieving cost $C^* = c(\mathcal{S}^*)$. If $K = 1$ then the greedy solution is optimal with $c(\mathcal{S}_1) = c(\mathcal{S}^*)$. Otherwise, when $K > 1$ the cost of the greedy solution satisfies*

$$\min_{a \in \mathcal{S}_K} c(\mathcal{S}_K \setminus \{a\}) < \left[1 + \ln\left(\frac{f(\mathcal{M})}{\rho_{min}} \cdot \frac{\Gamma}{C^*}\right)\right] \frac{C^*}{\Gamma}, \tag{5.103}$$

where

$$\rho_{min} = \min_{1 \leq k \leq K} \frac{f(\mathcal{S}_k) - f(\mathcal{S}_{k-1})}{c(\mathcal{S}_k) - c(\mathcal{S}_{k-1})}. \tag{5.104}$$

Proof. We provide the proof in Appendix 5.A. □

To provide some intuition for the result in Theorem 5.4.15, we restate the result below in the special case when the sensors each have unit cost $c(\mathcal{S}) = |\mathcal{S}|$.

Corollary 5.4.16. *When each measurement has unit cost $c(\mathcal{S}) = |\mathcal{S}|$ $\forall \mathcal{S} \subseteq \mathcal{M}$ then the conclusion of Theorem 5.4.15 reduces to*

$$|\mathcal{S}_K| \leq \left\lceil \left\{1 + \ln\left(\frac{f(\mathcal{M})}{\rho_{min}} \cdot \frac{\Gamma}{|\mathcal{S}^*|}\right)\right\} \frac{|\mathcal{S}^*|}{\Gamma} \right\rceil, \tag{5.105}$$

where $a \mapsto \lceil a \rceil = \min\{b \in \mathbb{Z} \ : \ a \leq b\}$ is the "ceiling" function that rounds a up to the smallest integer greater than or equal to a.

In most cases, the results of Theorem 5.4.15 and Corollary 5.4.16 provide pessimistic a priori bounds on the approximation ratio $c(\mathcal{S}_K)/C^*$. On the other hand, they provide much sharper a posteriori lower bounds on the optimal cost C^* once the greedy solution has been computed. One obtains the lower bound by finding the smallest value of C^* that satisfies the conclusion of Theorem 5.4.15.

Suppose we are using the objective function in Eq. 5.102 consisting of clipped square errors of optimal linear estimators. The square error of each component estimator has the upper and lower bounds stated in Lemma 5.4.17 thanks to the operator convexity of each function g_i.

Lemma 5.4.17 (Modular sandwich for mean square error)**.** *The change in the mean square error Eq. 5.101 for every* $\mathcal{S}, \mathcal{S}' \subseteq \mathcal{M}$ *is bounded above and below by*

$$\operatorname{Tr}\left[\boldsymbol{G}_i(\mathcal{S}')\left(\boldsymbol{P}_i(\mathcal{S}') - \boldsymbol{P}_i(\mathcal{S})\right)\right] \leq -E_i(\mathcal{S}') + E_i(\mathcal{S}) \leq \operatorname{Tr}\left[\boldsymbol{G}_i(\mathcal{S})\left(\boldsymbol{P}_i(\mathcal{S}') - \boldsymbol{P}(\mathcal{S})\right)\right], \tag{5.106}$$

where

$$\boldsymbol{G}_i(\mathcal{S}) := -\nabla g_i(\bar{\boldsymbol{P}}_i(\mathcal{S})) = \bar{\boldsymbol{P}}_i(\mathcal{S})^{-1}\boldsymbol{T}_i^T\boldsymbol{T}_i\bar{\boldsymbol{P}}_i(\mathcal{S})^{-1}. \tag{5.107}$$

Proof. This follows right away from the operator convexity of the matrix inverse shown in Lemma 5.A.2 in Appendix 5.A. □

This result allows us to construct a submodular upper bound for the objective Eq. 5.102 stated below in Proposition 5.4.18. In particular, we use Lemma 5.4.17 to provide a modular upper bound for each component, which is then clipped at the maximum possible value of each component.

Lemma 5.4.18 (Submodular Bound for Clipped MSE)**.** *The objective function Eq. 5.102 admits a submodular upper bound (see Definition 5.4.9) given by* $\hat{f}_{\mathcal{S}}(\mathcal{A}) = f(\mathcal{S}) + h_{\mathcal{S}}(\mathcal{A})$*, where*

$$h_{\mathcal{S}}(\mathcal{A}) = \frac{1}{N}\sum_{i=1}^{N} \min\left\{\operatorname{Tr}\left[\boldsymbol{G}_i(\mathcal{S})\boldsymbol{P}_i(\mathcal{A})\right],\ E_i(\mathcal{S}) - \varepsilon_i\right\}. \tag{5.108}$$

Proof. The fact that $\hat{f}_{\mathcal{S}}(\mathcal{A}) \geq f(\mathcal{S} \cup \mathcal{A})$ is an upper bound follows immediately from Lemma 5.4.17 and the fact that the ith component of f is bounded above by $E_i(\emptyset) - \varepsilon_i$. Furthermore, the ith component of $\hat{f}_{\mathcal{S}}(\mathcal{A})$ is a concave function $x \mapsto \min\left\{E_i(\emptyset) - E_i(\mathcal{S}) + x,\ E_i(\emptyset) - \varepsilon_i\right\}$ composed with a modular, monotone increasing function

$$\mathcal{A} \mapsto \operatorname{Tr}\left[\boldsymbol{G}_i(\mathcal{S})\boldsymbol{P}_i(\mathcal{A})\right] = \sum_{a\in\mathcal{A}} \operatorname{Tr}\left[\boldsymbol{G}_i(\mathcal{S})\boldsymbol{P}_i(\{a\})\right], \tag{5.109}$$

and is therefore submodular. □

As before, Lemma 5.4.17 can be used to construct an a priori bound for the submodularity gap ratio using the clipped square error objective and the submodular upper bound in Proposition 5.4.18.

To summarize, existing performance guarantees for greedy algorithms applied to non-submodular objectives are pessimistic and challenging to compute in practice. In this section, we presented an alternative greedy algorithm in which a local submodular upper bound on the original objective is maximized during each step. Such an upper bound is easy to compute for objectives like the square error of optimal linear estimators which involve operator convex components. The algorithm has

a posteriori guarantees that are less pessimistic than the original greedy algorithm and are much easier to compute.

5.5 Sensor placement guaranteeing ℓ^1-based recovery

An important type of nonlinear reconstruction function involves using ℓ^1 minimization to recover underling states in a set $\mathcal{X} \subset \mathbb{R}^n$ whose elements are sparse vectors. For instance, J. L. Callaham et al. [43] have used this approach to reconstruct flow fields in fluid dynamics problems by first coding them as sparse vectors in a dictionary and using ℓ^1 minimization to recover these vectors from small numbers of sensor measurements. The question we ask here is how to select the locations of such sensors in order to guarantee that underlying sparse states can always be recovered using the ℓ^1 minimization approach. In this section, we will develop an approach for minimal sensor placement that allows ℓ^1 minimization to exactly recover state vectors with any of a given set of sparsity patterns. These sparsity patterns can be determined via dictionary learning or sparse coding of a data set collected from the system under consideration [148, 232, 5, 251]. Our approach could be especially useful when there are a somewhat small number of known sparsity patterns that we wish to detect using our sensors, for example corresponding to the most probable configurations of a system or to anomalies.

Designing measurement matrices that enable recovery of sparse vectors is a central goal in the field of compressed sensing [93, 50, 81], which offers a variety of approaches and design criteria. For instance, many different kinds of randomly generated measurement matrices guarantee exact recovery of sparse vectors by ℓ^1 minimization with high probability [48, 49, 44, 45]. However, in our case, we are interested in selecting optimal measurements in a deterministic manner from among a given collection that guarantee recovery of sparse vectors. A complicating factor is that many conditions that guarantee exact recovery of sparse vectors such as the Restricted Isometry Property (RIP) for the measurement matrix [48, 44] are difficult to verify, making their use as optimization objectives impractical. Fortunately, the convex duality condition appearing, e.g., in [47, 48, 45] provides us with a practical optimization criteria that guarantees exact recovery of given sparsity patterns by ℓ^1 minimization. We propose a constellation of convex programming approaches for sensor placement based on satisfying this duality condition in order to ensure exact recovery of state vectors having particular sparsity patterns.

We consider the simplest version of the ℓ^1 minimization problem for recovering sparse states from a small collection of measurements. The states $\boldsymbol{x} \in \mathbb{R}^n$ of our system are assumed to live in a subset

$\mathcal{X} \subset \mathbb{R}^n$ consisting of "sparse" vectors having only a small number of nonzero entries. If $\boldsymbol{y} = \boldsymbol{M}_{\mathcal{S}}\boldsymbol{x}$ is a collection of linear measurements of the state coming from a collection of sensors $\mathcal{S} \subset \mathcal{M}$ where the dimension $d_{\mathcal{S}}$ of $\boldsymbol{y}$ is smaller than the dimension of $\boldsymbol{x}$, then the reconstruction problem for $\boldsymbol{x}$ is under-determined. When the state $\boldsymbol{x}$ is known to be sparse, one way to approximately reconstruct $\boldsymbol{x}$ is by solving the ℓ^1 minimization problem

$$\hat{\boldsymbol{x}} = \boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{y}) = \underset{\boldsymbol{z}\in\mathbb{R}^n}{\operatorname{argmin}} \ \|\boldsymbol{z}\|_1 \quad \text{s.t.} \quad \boldsymbol{M}_{\mathcal{S}}\boldsymbol{z} = \boldsymbol{y}. \tag{5.110}$$

The is a widely studied convex program for which there are variety of highly efficient solution methods. The main question is under what conditions on the measurement matrix $\boldsymbol{M}_{\mathcal{S}}$ can Eq. 5.110 be expected to exactly reconstruct the original sparse vector $\boldsymbol{x} \in \mathcal{X}$? The result in Theorem 5.5.1 answers this question by providing a sufficient condition for ℓ^1 minimization to recover $\boldsymbol{x}$ based on the existence of a vector $\boldsymbol{v}_{\mathcal{S}}$ in the measurement space referred to as a "dual certificate".

Theorem 5.5.1 (dual certificate of recovery, Theorem 9.8 in [62], Lemma 3.1 in [45])**.** *Assume the columns of $\boldsymbol{M}_{\mathcal{S}}$ corresponding to the nonzero entries of $\boldsymbol{x}$ are linearly independent. If*

$$\boxed{\exists \boldsymbol{v}_{\mathcal{S}} \in \mathbb{R}^{d_{\mathcal{S}}} \quad \textit{such that} \quad \begin{cases} \left[\boldsymbol{M}_{\mathcal{S}}^T \boldsymbol{v}_{\mathcal{S}}\right]_i = \operatorname{sgn}([\boldsymbol{x}]_i) & \textit{if } [\boldsymbol{x}]_i \neq 0 \\ -1 < \left[\boldsymbol{M}_{\mathcal{S}}^T \boldsymbol{v}_{\mathcal{S}}\right]_i < 1 & \textit{otherwise,} \end{cases}} \tag{5.111}$$

then $\boldsymbol{x}$ is the unique solution of ℓ^1 minimization (Eq. 5.110) with $\boldsymbol{y} = \boldsymbol{M}_{\mathcal{S}}\boldsymbol{x}$.

Proof. For completeness, we reproduce the proof given in the notes by Y. Chen [62] in Appendix 5.A. □

We observe that the condition provided by Theorem 5.5.1 only depends on the signed sparsity pattern of the vector $\boldsymbol{x}$, and not on the specific values of its entries. In other words, a dual certificate provided by Theorem 5.5.1 certifies that all vectors with the same signed sparsity pattern as $\boldsymbol{x}$ can be recovered by ℓ^1 minimization using the measurements provided by $\boldsymbol{M}_{\mathcal{S}}$.

The second key observation is that the existence of a dual certificate described by Theorem 5.5.1 is a linear feasibility problem. We can use this problem as a criterion for selecting sensors $\mathcal{S} \subset \mathcal{M}$. The main idea is to consider all of the available sensors forming the matrix $\boldsymbol{M} = \boldsymbol{M}_{\mathcal{M}}$ and to find maximally sparse dual certificates for the desired sparsity patterns. The nonzero entries of the resulting sparse dual certificates indicate the subset of measurements $\mathcal{S} \subset \mathcal{M}$ that we select to guarantee recovery of the given sparsity patterns. To illustrate, we begin by considering a single

signed sparsity pattern $\boldsymbol{s} \in \{-1, 0, 1\}^n$ and we try to find a sparse vector $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M)$ in the convex set

$$\mathcal{V}(\boldsymbol{s}) = \left\{ \boldsymbol{v} \in \mathbb{R}^{d_{\mathcal{M}}} \ : \ [\boldsymbol{M}^T \boldsymbol{v}]_i = [\boldsymbol{s}]_i \text{ if } [\boldsymbol{s}]_i \neq 0, \quad \text{and} \quad -1 < [\boldsymbol{M}^T \boldsymbol{v}]_i < 1 \text{ otherwise} \right\}. \tag{5.112}$$

Selecting the subset $\mathcal{S} = \{j_1, \ldots, j_K\} \subset \mathcal{M}$ containing the support (nonzero elements) of $\boldsymbol{v}$ automatically provides a dual certificate

$$\boldsymbol{v}_{\mathcal{S}} = (\boldsymbol{v}_{j_1}, \ldots, \boldsymbol{v}_{j_K}) \tag{5.113}$$

guaranteeing that any $\boldsymbol{x} \in \mathbb{R}^n$ with $\operatorname{sgn}(\boldsymbol{x}) = \boldsymbol{s}$ can be recovered by ℓ^1 minimization using the sensors $\boldsymbol{M}_{\mathcal{S}}$ according to Theorem 5.5.1.

Now suppose there are multiple sparsity patterns $\boldsymbol{s}^1, \ldots, \boldsymbol{s}^N$. In this case, we try to find multiple vectors $\boldsymbol{v}^1, \ldots, \boldsymbol{v}^N$ in the corresponding sets $\mathcal{V}(\boldsymbol{s}^i)$ so that the support of every $\boldsymbol{v}^i$ is contained in the same small subset $\mathcal{S} \subset \mathcal{M}$. Here, each $\boldsymbol{v}^i_{\mathcal{S}}$ provides a dual certificate that any vector $\boldsymbol{x} \in \mathbb{R}^n$ with signed sparsity pattern $\operatorname{sgn}(\boldsymbol{x}) = \boldsymbol{s}^i$ can be recovered by ℓ^1 minimization using the sensors $\boldsymbol{M}_{\mathcal{S}}$.

We may employ a variety of optimization approaches for sensor selection based on convex relaxation of the desired structured sparsity of the dual certificate vectors. These approaches bear resemblance to the convex optimization approaches we discussed in Section 5.4.1 as well as to the group LASSO method [296] described briefly at the end of Section 5.2.1. Letting

$$\boldsymbol{V}_j = \begin{bmatrix} \boldsymbol{v}_j^1 & \cdots & \boldsymbol{v}_j^N \end{bmatrix}, \qquad j \in \mathcal{M}, \tag{5.114}$$

any $\boldsymbol{V}_j$ containing a nonzero entry corresponds to a sensor that we include in the set $\mathcal{S}$. One way to promote sparsity among these matrices is by minimizing

$$\|\boldsymbol{V}_{\mathcal{M}}\|_{1,\infty} = \sum_{j \in \mathcal{M}} \|\boldsymbol{V}_j\|_\infty = \sum_{j \in \mathcal{M}} \max_{1 \leq i \leq N} \|\boldsymbol{v}_j^i\|_\infty, \tag{5.115}$$

subject to constraints $\boldsymbol{v}^i \in \mathcal{V}(\boldsymbol{s}^i)$, $i = 1, \ldots, N$. One wrinkle is that the constraints of the form $-1 < [\boldsymbol{M}\boldsymbol{v}]_i < 1$ are open and cannot be imposed directly. The simplest approach is to introduce a small constant $\varepsilon > 0$ and to impose constraints $\boldsymbol{v}^i \in \mathcal{V}_\varepsilon(\boldsymbol{s}^i)$ using the closed convex polyhedra

$$\mathcal{V}_\varepsilon(\boldsymbol{s}) = \left\{ \boldsymbol{v} \in \mathbb{R}^{d_{\mathcal{M}}} \ : \ [\boldsymbol{M}^T \boldsymbol{v}]_i = [\boldsymbol{s}]_i \text{ if } [\boldsymbol{s}]_i \neq 0, \text{ and } -1 + \varepsilon \leq [\boldsymbol{M}^T \boldsymbol{v}]_i \leq 1 - \varepsilon \text{ otherwise} \right\}. \tag{5.116}$$

This yields a relaxed optimization problem for minimum sensor placement,

$$\boxed{\underset{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^N\in\mathbb{R}^{d_{\mathcal{M}}}}{\text{minimize}} \|\boldsymbol{V}_{\mathcal{M}}\|_{1,\infty} \qquad \text{s.t.} \qquad \boldsymbol{v}^i \in \mathcal{V}_\varepsilon(\boldsymbol{s}^i), \quad \forall i = 1,\ldots,N,} \tag{5.117}$$

which can be solved (after introducing auxiliary variables) by linear programming! Another way to promote sparsity among the matrices $\boldsymbol{V}_i$ is to minimize the group LASSO penalty

$$\|\boldsymbol{V}_{\mathcal{M}}\|_{1,F} = \sum_{j\in\mathcal{M}} \|\boldsymbol{V}_j\|_F \tag{5.118}$$

in Eq. 5.117 instead of $\|\boldsymbol{V}_{\mathcal{M}}\|_{1,\infty}$. We can also use other convex, but nonlinear methods to impose the constraints $-1 < [\boldsymbol{M}\boldsymbol{v}]_i < 1$. For instance, we could add a convex logarithmic penalization term

$$\rho(\boldsymbol{V}_{\mathcal{M}}) = -\sum_{i=1}^{N} \sum_{\substack{1\le k\le d_{\mathcal{M}}:\\ [\boldsymbol{s}^i]_k=0}} \log\Big((1+[\boldsymbol{M}^T\boldsymbol{v}^i]_k)(1-[\boldsymbol{M}^T\boldsymbol{v}^i]_k)\Big), \tag{5.119}$$

with a small weight $\gamma > 0$ to the optimization objective. The result is a convex programming problem

$$\boxed{\underset{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^N\in\mathbb{R}^{d_{\mathcal{M}}}}{\text{minimize}} \|\boldsymbol{V}_{\mathcal{M}}\|_{1,F \text{ or } \infty} + \gamma\rho(\boldsymbol{V}_{\mathcal{M}}) \qquad \text{s.t.} \qquad [\boldsymbol{M}^T\boldsymbol{v}^i]_k = [\boldsymbol{s}^i]_k, \quad \forall i,k : [\boldsymbol{s}^i]_k \neq 0,} \tag{5.120}$$

which has only linear equality constraints.

Suppose that we have a set $\mathcal{X} \subset \mathbb{R}^n$ of sparse state vectors and a probability measure μ on $\mathcal{X}$. Suppose we draw an independent, identically distributed sample $\boldsymbol{x}_1,\ldots,\boldsymbol{m} \in \mathcal{X}$ under μ and choose sensors $\mathcal{S}$ that guarantee ℓ^1 minimization-based recovery of any $\boldsymbol{x} \in \mathbb{R}^n$ with signed sparsity among the samples $\operatorname{sgn}(\boldsymbol{x}) \in \{\operatorname{sgn}(\boldsymbol{x}_1),\ldots,\operatorname{sgn}(\boldsymbol{x}_m)\}$. Let $\mathcal{X}_{\mathcal{S}}$ denote the subset of states in $\mathcal{X}$ that can be recovered by ℓ^1 minimization using the sensors $\mathcal{S}$. How many samples m must we draw to ensure that most states can be recovered in the sense that $\mu(\mathcal{X}_{\mathcal{S}}) > 1-\delta$ with high probability? Theorem 5.5.2, below, answers this question provided with an a priori estimate L on the size of the selected set of sensors. The interesting aspect of Theorem 5.5.2 is that the required sample size m is much smaller than the number of sparsity patterns among vectors in $\mathbb{R}^n$, which has combinatorial growth.

Theorem 5.5.2 (Sampling to probably recover most states)**.** *Let $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$ be independent random vectors drawn from $\mathcal{X}$ according to the probability measure μ and let the set of sensors $\mathcal{S}$ be chosen so that ℓ^1 minimization is guaranteed to recover $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$. We assume that the selected set $\mathcal{S}$ always*

has at most L elements. If the number of sampled vectors is at least

$$m \geq \frac{1}{2\delta^2}\big(L\log(|\mathcal{M}|) - \log(p)\big), \tag{5.121}$$

then $\mu(\mathcal{X}_{\mathcal{S}}) > 1 - \delta$ with probability at least $1 - p$.

Proof. We use an argument based on the union bound and Hoeffding's inequality to control the probability that the fraction of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ recovered by ℓ^1 minimization differs from $\mu(\mathcal{X}_{\mathcal{S}'})$ by more than δ for any set of sensors $\mathcal{S}'$ with $|\mathcal{S}'| \leq L$. Since $\mathcal{S}$ is assumed to be among such sets and $\mathcal{S}$ guarantees recovery of each $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$, the stated result holds for $\mathcal{S}$. We give the detailed proof in Appendix 5.A. □

In the worst case, we can set $L = \min\{|\mathcal{M}|, n\}$ in Theorem 5.5.2 because we cannot select more sensors than are available, and any set of n linearly independent measurements will recover the state $\boldsymbol{x} \in \mathbb{R}^n$. More sophisticated arguments based on dimensionality considerations might also be used to provide superior bounds for the number of selected sensors. We also note that the proof of Theorem 5.5.2 fundamentally has nothing to do with the specific recovery procedure, in this case ℓ^1 minimization; the argument relies only on the fact that $\mathcal{S}$ is chosen so that the recovery procedure works for the sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$.

In conclusion, the dual certificate provides us with a convenient way to determine whether a given sparsity pattern can be recovered from sensor measurements by ℓ^1 minimization. We use this criterion as a basis for several proposed convex optimization approaches for minimal sensor placement to guarantee that a desired collection of sparsity patterns can be recovered. Interestingly, we have given a result that shows that we do not have to include an enormous number of sparsity patterns in this set to guarantee that most sparsity patterns in the underlying set of states $\mathcal{X}$ can be recovered by ℓ^1 minimization from the selected sensor measurements with high probability. Future work will include studying the robustness of the sensors selected using the proposed methods to noise and disturbances.

Appendix

5.A Chapter 5 Proofs

Proof of Proposition 5.2.1 (optimal linear estimator). We shall find an optimal linear estimator of the form

$$\hat{\boldsymbol{g}} = \boldsymbol{A}\boldsymbol{y}_{\mathbb{S}}, \tag{5.122}$$

where $\boldsymbol{y}_{\mathbb{S}} = \boldsymbol{M}_{\mathbb{S}}\boldsymbol{x} + \boldsymbol{n}_{\mathbb{S}}$, that minimizes the mean square error $\mathbb{E}\,\|\boldsymbol{g} - \hat{\boldsymbol{g}}\|_2^2$. Recalling the definition of $\boldsymbol{C}_{\boldsymbol{y}_{\mathbb{S}}}$, observe that $\boldsymbol{y}_{\mathbb{S}}$ has no variance outside $\operatorname{Range}\boldsymbol{C}_{\boldsymbol{y}_{\mathbb{S}}}$. Consider the eigen-decomposition the symmetric positive semi-definite covariance matrix

$$\boldsymbol{C}_{\boldsymbol{y}_{\mathbb{S}}} = \begin{bmatrix}\boldsymbol{W} & \boldsymbol{W}_0\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}^2 & \\ & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{W}^T \\ \boldsymbol{W}_0^T\end{bmatrix} = \boldsymbol{W}\boldsymbol{\Lambda}^2\boldsymbol{W}^T, \tag{5.123}$$

where $\boldsymbol{\Lambda}^2$ contains the strictly positive eigenvalues and let $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{W}$. The matrix $\boldsymbol{B}$ determines the estimate because

$$\hat{\boldsymbol{g}} = \boldsymbol{A}\boldsymbol{y}_{\mathbb{S}} = \boldsymbol{A}\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{y}_{\mathbb{S}} = \boldsymbol{B}\boldsymbol{W}^T\boldsymbol{y}_{\mathbb{S}}, \tag{5.124}$$

and we shall show that there is a unique optimal $\boldsymbol{B}$. The error to be minimized over all linear estimates is given by

$$E(\boldsymbol{B}) := \mathbb{E}\,\|\boldsymbol{g} - \boldsymbol{B}\boldsymbol{W}^T\boldsymbol{y}_{\mathbb{S}}\|_2^2 = \operatorname{Tr}\boldsymbol{C}_{\boldsymbol{g}} - 2\operatorname{Tr}\left(\boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_{\mathbb{S}}}\boldsymbol{W}\boldsymbol{B}^T\right) + \operatorname{Tr}\left(\boldsymbol{B}\boldsymbol{\Lambda}^2\boldsymbol{B}^T\right). \tag{5.125}$$

This is a strictly positive definite quadratic objective and is therefore uniquely minimized by

$$\boldsymbol{B}_* = \boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_{\mathbb{S}}}\boldsymbol{W}\boldsymbol{\Lambda}^{-2}, \tag{5.126}$$

yielding the optimal linear estimate

$$\hat{\boldsymbol{g}} = \boldsymbol{B}_*\boldsymbol{W}^T\boldsymbol{y}_{\mathbb{S}} = \boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_{\mathbb{S}}}\boldsymbol{W}\boldsymbol{\Lambda}^{-2}\boldsymbol{W}^T\boldsymbol{y}_{\mathbb{S}} = \boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_{\mathbb{S}}}\boldsymbol{C}^{+}_{\boldsymbol{y}_{\mathbb{S}}}\boldsymbol{y}_{\mathbb{S}}. \tag{5.127}$$

When the matrix $\boldsymbol{C}_{\boldsymbol{y}_{\mathbb{S}}}$ is not invertible, then there are an infinite number of matrices $\boldsymbol{A}$ such that $\boldsymbol{A}\boldsymbol{W} = \boldsymbol{B}_*$, of which $\boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_{\mathbb{S}}}\boldsymbol{C}^{+}_{\boldsymbol{y}_{\mathbb{S}}}$ is only one such choice. The optimal linear estimate is always unique because all optimal linear estimators agree on the range of $\boldsymbol{C}_{\boldsymbol{y}_{\mathbb{S}}}$. On the other hand, when $\boldsymbol{C}_{\boldsymbol{y}_{\mathbb{S}}}$ is invertible, then $\boldsymbol{W}$ is also invertible and $\boldsymbol{B}_*$ uniquely determines $\boldsymbol{A}$. Hence, the optimal

linear estimate is always unique, but the optimal linear estimator is unique if and only if $\boldsymbol{C}_{\boldsymbol{y}_\mathbb{s}}$ is invertible.

Notice that by a property of the Moore-Penrose pseudoinverse the covariance of the estimate is given by

$$\boldsymbol{C}_{\hat{\boldsymbol{g}}} = \boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_\mathbb{s}} \boldsymbol{C}_{\boldsymbol{y}_\mathbb{s}}^{+} \boldsymbol{C}_{\boldsymbol{y}_\mathbb{s}} \boldsymbol{C}_{\boldsymbol{y}_\mathbb{s}}^{+} \boldsymbol{C}_{\boldsymbol{y}_\mathbb{s},\boldsymbol{g}} = \boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_\mathbb{s}} \boldsymbol{C}_{\boldsymbol{y}_\mathbb{s}}^{+} \boldsymbol{C}_{\boldsymbol{y}_\mathbb{s},\boldsymbol{g}}. \tag{5.128}$$

Expanding the error covariance and substituting the estimator (5.127) and its covariance (5.128) yields the result

$$\boldsymbol{C}_{\boldsymbol{e}} = \boldsymbol{C}_{\boldsymbol{g}} - \boldsymbol{C}_{\boldsymbol{g},\boldsymbol{y}_\mathbb{s}} \boldsymbol{C}_{\boldsymbol{y}_\mathbb{s}}^{+} \boldsymbol{C}_{\boldsymbol{y}_\mathbb{s},\boldsymbol{g}} = \boldsymbol{C}_{\boldsymbol{g}} - \boldsymbol{C}_{\hat{\boldsymbol{g}}}. \tag{5.129}$$

□

Lemma 5.A.1 (Matrix Inversion Reverses Loewner Order)**.** *Let $\boldsymbol{A} \succ \boldsymbol{0}$ and $\boldsymbol{B} \succ \boldsymbol{0}$ be symmetric positive definite matrices with $\boldsymbol{A} \succeq \boldsymbol{B}$. Then the matrix inverses have the opposite Loewner order relation $\boldsymbol{A}^{-1} \preceq \boldsymbol{B}^{-1}$.*

Proof. By assumption, one readily checks that $\boldsymbol{B}^{-1/2}\boldsymbol{A}\boldsymbol{B}^{-1/2} \succeq \boldsymbol{I}$ since for every $\boldsymbol{x}$, we have

$$\left(\boldsymbol{x}^T \boldsymbol{B}^{-1/2}\right) \boldsymbol{A} \left(\boldsymbol{B}^{-1/2}\boldsymbol{x}\right) \geq \left(\boldsymbol{x}^T \boldsymbol{B}^{-1/2}\right) \boldsymbol{B} \left(\boldsymbol{B}^{-1/2}\boldsymbol{x}\right) = \boldsymbol{x}^T\boldsymbol{x}. \tag{5.130}$$

This means that all eigenvalues of $\boldsymbol{B}^{-1/2}\boldsymbol{A}\boldsymbol{B}^{-1/2}$ are at least 1, so the eigenvalues of the inverse $\boldsymbol{B}^{1/2}\boldsymbol{A}^{-1}\boldsymbol{B}^{1/2}$ lie in $(0, 1]$. Therefore $\boldsymbol{B}^{1/2}\boldsymbol{A}^{-1}\boldsymbol{B}^{1/2} \preceq \boldsymbol{I}$ and so for every $\boldsymbol{x}$ we have

$$\begin{aligned} \boldsymbol{x}^T\boldsymbol{A}^{-1}\boldsymbol{x} &= \left(\boldsymbol{x}^T \boldsymbol{B}^{-1/2}\right) \boldsymbol{B}^{1/2}\boldsymbol{A}^{-1}\boldsymbol{B}^{1/2} \left(\boldsymbol{B}^{-1/2}\boldsymbol{x}\right) \\ &\leq \left(\boldsymbol{x}^T \boldsymbol{B}^{-1/2}\right) \left(\boldsymbol{B}^{-1/2}\boldsymbol{x}\right) = \boldsymbol{x}^T\boldsymbol{B}^{-1}\boldsymbol{x}, \end{aligned} \tag{5.131}$$

completing the proof. □

Lemma 5.A.2 (Matrix Inversion is Convex in the Loewner Order)**.** *Let $\boldsymbol{A} \succ \boldsymbol{0}$ and $\boldsymbol{B} \succ \boldsymbol{0}$ be symmetric positive definite matrices and $\alpha \in [0, 1]$. Then, we have*

$$(\alpha\boldsymbol{A} + (1-\alpha)\boldsymbol{B})^{-1} \preceq \alpha\boldsymbol{A}^{-1} + (1-\alpha)\boldsymbol{B}^{-1}. \tag{5.132}$$

Proof. Our proof is based on the one found in [52], in fact, it is a special case of the Loewner-Heinz theorem. Let $\boldsymbol{C} = \boldsymbol{A}^{-1/2}\boldsymbol{B}\boldsymbol{A}^{-1/2}$ and factor out $\boldsymbol{A}^{-1/2}$ from the difference of the right and left

hand sides of the desired inequality

$$(\alpha \boldsymbol{A} + (1-\alpha)\boldsymbol{B})^{-1} - \alpha \boldsymbol{A}^{-1} - (1-\alpha)\boldsymbol{B}^{-1} = \\ \boldsymbol{A}^{-1/2}\left[(\alpha \boldsymbol{I} + (1-\alpha)\boldsymbol{C})^{-1} - \alpha \boldsymbol{I} - (1-\alpha)\boldsymbol{C}^{-1}\right]\boldsymbol{A}^{-1/2}. \quad (5.133)$$

Consider the eigen-decomposition $\boldsymbol{C} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ and observe that

$$(\alpha \boldsymbol{I} + (1-\alpha)\boldsymbol{C})^{-1} - \alpha \boldsymbol{I} - (1-\alpha)\boldsymbol{C}^{-1} = \boldsymbol{V}\left[(\alpha \boldsymbol{I} + (1-\alpha)\boldsymbol{\Lambda})^{-1} - \alpha \boldsymbol{I} - (1-\alpha)\boldsymbol{\Lambda}^{-1}\right]\boldsymbol{V}^T. \quad (5.134)$$

If $\lambda > 0$ is an eigenvalue of $\boldsymbol{C}$ then $(\alpha + (1-\alpha)\lambda)^{-1} \leq \alpha - (1-\alpha)\lambda^{-1}$ by convexity of the function $x \mapsto x^{-1}$ on the positive real numbers. Therefore, we have

$$(\alpha \boldsymbol{I} + (1-\alpha)\boldsymbol{\Lambda})^{-1} - \alpha \boldsymbol{I} - (1-\alpha)\boldsymbol{\Lambda}^{-1} \preceq \boldsymbol{0} \quad (5.135)$$

from which it immediately follows that

$$(\alpha \boldsymbol{A} + (1-\alpha)\boldsymbol{B})^{-1} - \alpha \boldsymbol{A}^{-1} - (1-\alpha)\boldsymbol{B}^{-1} \preceq \boldsymbol{0}, \quad (5.136)$$

which proves convexity. □

Proof. (Proof of Eq. 5.49) Letting $\mathcal{S}_K^*$ denote a maximizer of the relaxed objective in Eq. 5.47 under the sensor budget constraint $|\mathcal{S}| \leq K$, and combining Eq. 5.48 with Theorem 5.0.2 we obtain

$$F_{\alpha\gamma}(\mathcal{S}_k) \leq \frac{F_\infty - f_\gamma(\mathcal{S}_k)}{1-\alpha^2} \leq \frac{1}{1-\alpha^2}\left[F_\infty - \left(1 - e^{-k/K}\right) f_\gamma(\mathcal{S}_K^*)\right]. \quad (5.137)$$

Since $f_\gamma(\mathcal{S}_K^*) \geq f_\gamma(\mathcal{S})$ for every $\mathcal{S}$ with $|\mathcal{S}| \leq K$ and $|\tilde{\mathcal{S}}_K^*| \leq K$, we have $f_\gamma(\mathcal{S}_K^*) \geq f_\gamma(\tilde{\mathcal{S}_K}^*)$. Finally, because $f_\gamma \geq \tilde{f}_\gamma$, we obtain

$$F_{\alpha\gamma}(\mathcal{S}_k) \leq \frac{1}{1-\alpha^2}\left[F_\infty - \left(1 - e^{-k/K}\right) \tilde{f}_\gamma(\tilde{\mathcal{S}}_K^*)\right], \quad (5.138)$$

which can be rearranged and minimized over K to give Eq. 5.49. □

Proof. (Proof of Eq. 5.50) Proceeding in the same way as in the proof of Eq. 5.49, we find

$$f_\gamma(\mathcal{S}_k) \geq \left(1 - e^{-k/K}\right) f_\gamma(\mathcal{S}_K^*) \tag{5.139}$$

$$\geq \left(1 - e^{-k/K}\right) f_\gamma(\tilde{\mathcal{S}}_K^*) \tag{5.140}$$

$$\geq \left(1 - e^{-k/K}\right) \tilde{f}_\gamma(\tilde{\mathcal{S}}_K^*) = \left(1 - e^{-k/K}\right) (F_\infty - F_\gamma(\tilde{\mathcal{S}}_K^*)). \tag{5.141}$$

The above inequality can be rearranged to provide a lower bound on $F_\gamma(\tilde{\mathcal{S}}_K^*)$, which is then maximized over k, yielding Eq. 5.50. □

Proof of Theorem 5.4.1 (Existence of local reconstruction). Suppose that $\mathcal{X}$, $\mathcal{M}$, and $\mathcal{N}$ have dimensions $d_\mathcal{X}$, $d_\mathcal{M}$, and $d_\mathcal{N}$ respectively. Choose any $x_0 \in \mathcal{X}$ and let $\mathcal{U}' \subset \mathcal{X}$ be an open neighborhood of x_0 over which $\mathrm{D}\, f$ has constant rank r. By the rank theorem (Theorem 4.12 on p.81 in J. M. Lee [149]), there is an open neighborhood $\mathcal{U} \subset \mathcal{U}'$ of x_0 and a neighborhood $\tilde{\mathcal{V}} \subset \mathcal{M}$ of $f(x_0)$ and local parametrizations $\phi : \mathbb{R}^{d_\mathcal{X}} \to \mathcal{U}$, $\tilde{\psi} : \mathbb{R}^{d_\mathcal{M}} \to \tilde{\mathcal{V}}$ such that $f(\mathcal{U}) \subset \tilde{\mathcal{V}}$ and

$$\psi^{-1} \circ f \circ \phi(z_1, \ldots, z_r, z_{r+1}, \ldots, z_\mathcal{X}) = (z_1, \ldots, z_r, 0, \ldots, 0). \tag{5.142}$$

Taking any $p \in f(\mathcal{U})$, we observe that $p = \tilde{\psi}(\bar{z}_1, \ldots, \bar{z}_r, 0, \ldots, 0)$ for some $\bar{z}_1, \ldots, \bar{z}_r$ and so

$$f^{-1}(p) \cap \mathcal{U} = \left\{\phi(\bar{z}_1, \ldots, \bar{z}_r, z_{r+1}, \ldots z_{d_\mathcal{M}}) \ : \ (z_{r+1}, \ldots, z_{d_\mathcal{M}}) \in \mathbb{R}^{d_\mathcal{M} - r}\right\} \tag{5.143}$$

is a smooth, connected submanifold of $\mathcal{U}$ with codimension r.

For each $x \in \mathcal{U}$, we define the linear map $A_x : T_{f(x)}\mathcal{M} \to T_{g(x)}\mathcal{N}$ according to

$$A_x \xi = \mathrm{D}\, g(x)\eta \quad \text{for any } \eta \in T_x\mathcal{X} \text{ such that } \mathrm{D}\, f(x)\eta = \xi. \tag{5.144}$$

The map A_x is well-defined thanks to the assumption stated in Eq 5.58, for if we choose another $\eta' \in T_x\mathcal{X}$ with $\mathrm{D}\, f(x)\eta' = \xi$ then $\eta - \eta' \in \ker \mathrm{D}\, f(x) \subset \ker \mathrm{D}\, g(x)$, then we have $\mathrm{D}\, g(x)\eta' = \mathrm{D}\, g(x)\eta$. We use this map to show that $g(x)$ is constant for every $x \in f^{-1}(p) \cap \mathcal{U}$, allowing us to define $h(p) = g(x)$ for any $x \in f^{-1}(p) \cap \mathcal{U}$. If $r = d_\mathcal{X}$ then $f^{-1}(p) \cap \mathcal{U}$ consists of a single point and the statement is trivial. If $r < d_\mathcal{X}$, we choose any distinct $x, x' \in f^{-1}(p) \cap \mathcal{U}$ and join them by a smooth path $\gamma : [0, 1] \to f^{-1}(p) \cap \mathcal{U}$ with $\gamma(0) = x$ and $\gamma(1) = x'$. By the fundamental theorem of calculus

and the definition of A_x, we have

$$\begin{aligned} g(x') - g(x) &= \int_0^1 \mathrm{D}\, g(\gamma(t))\, \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, \gamma(t)\, \mathrm{d}\, t \\ &= \int_0^1 A_{\gamma(t)}\, \mathrm{D}\, f(\gamma(t))\, \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, \gamma(t)\, \mathrm{d}\, t \\ &= \int_0^1 A_{\gamma(t)}\, \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} (\underbrace{f \circ \gamma}_{p})(t)\, \mathrm{d}\, t = 0. \end{aligned} \tag{5.145}$$

Therefore, the function h is well-defined on $f(\mathcal{U})$ and we have $g(x) = h(f(x))$ for every $x \in \mathcal{U}$. Moreover, the derivative of h is given by

$$\mathrm{D}\, h(y) = A_x \quad \text{for any } x \in \mathcal{X} \text{ such that } f(x) = y, \tag{5.146}$$

because $\mathrm{D}\, g(x)\eta = \mathrm{D}\, h(y)\, \mathrm{D}\, f(x)\eta = A_x\eta$ for every $\eta \in T_x\mathcal{X}$. Repeated differentiation of $g \circ \phi = h \circ f \circ \phi$ by the chain rule shows that derivatives of h exist up to any order, and so h is smooth. $\square$

Proof of Proposition 5.4.7 (greedy submodularity ratio for logarithmic objective). Consider the greedily chosen set $\mathcal{S}_k$ and any $\Omega \subset \mathcal{M}$ with $|\Omega| = K$. Applying Lemma 5.4.6 yields a lower bound for the ratio

$$R = \frac{\sum_{\omega \in \Omega \setminus \mathcal{S}_k} [f(\mathcal{S}_k \cup \{\omega\}) - f(\mathcal{S}_k)]}{f(\mathcal{S}_k \cup \Omega) - f(\mathcal{S}_k)} \geq \frac{\sum_{\omega \in \Omega \setminus \mathcal{S}_k} \left(\frac{E(\mathcal{S}_k)}{E(\mathcal{S}_k \cup \{\omega\})} \right) \sum_{i=1}^N \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k \cup \{\omega\}) \boldsymbol{P}_i(\{\omega\}) \right]}{\sum_{\omega \in \Omega \setminus \mathcal{S}_k} \sum_{i=1}^N \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k) \boldsymbol{P}_i(\{\omega\}) \right]}. \tag{5.147}$$

This equation may be re-written as a weighted average

$$R \geq \sum_{\omega \in \Omega \setminus \mathcal{S}_k} w_\Omega(\omega) \left(\frac{E(\mathcal{S}_k)}{E(\mathcal{S}_k \cup \{\omega\})} \right) \left(\frac{\sum_{i=1}^N \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k \cup \{\omega\}) \boldsymbol{P}_i(\{\omega\}) \right]}{\sum_{i=1}^N \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k) \boldsymbol{P}_i(\{\omega\}) \right]} \right) \tag{5.148}$$

with weights

$$w_\Omega(\omega) = \frac{\sum_{i=1}^N \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k) \boldsymbol{P}_i(\{\omega\}) \right]}{\sum_{\omega' \in \Omega \setminus \mathcal{S}_k} \sum_{i=1}^N \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k) \boldsymbol{P}_i(\{\omega'\}) \right]} \geq 0 \tag{5.149}$$

that sum to 1. All of the dependence on Ω for each term in the bound on R is captured by these weights. We can remove this dependence by bounding the weighted average from below by its smallest term

$$R \geq \min_{\omega \in \Omega \setminus \mathcal{S}_k} \left(\frac{E(\mathcal{S}_k)}{E(\mathcal{S}_k \cup \{\omega\})} \right) \left(\frac{\sum_{i=1}^N \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k \cup \{\omega\}) \boldsymbol{P}_i(\{\omega\}) \right]}{\sum_{i=1}^N \mathrm{Tr}\left[\boldsymbol{G}_i(\mathcal{S}_k) \boldsymbol{P}_i(\{\omega\}) \right]} \right). \tag{5.150}$$

Minimizing over all subsets $\Omega \subset \mathcal{M}$ with $|\Omega| = K$ and $k = 0, \ldots, K-1$, we obtain the first inequality

in Eq. 5.90.

To obtain the a priori bound given by the second inequality in Eq. 5.90, we must remove the dependence on the greedily chosen sets. To do this, we observe that if $\bar{\boldsymbol{P}} = \sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T$ is a symmetric positive semi-definite eigen-decomposition with $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ taken to be orthonormal and $\boldsymbol{G} = \bar{\boldsymbol{P}}^{-1} \boldsymbol{T}^T \boldsymbol{T} \bar{\boldsymbol{P}}^{-1}$, then

$$\operatorname{Tr}(\boldsymbol{G}\boldsymbol{H}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j \operatorname{Tr}\left(\boldsymbol{v}_i \boldsymbol{v}_i^T \boldsymbol{T}^T \boldsymbol{T} \boldsymbol{v}_j \boldsymbol{v}_j^T \boldsymbol{H}\right) \tag{5.151}$$

for any $n \times n$ matrix $\boldsymbol{H}$. Assuming that $\boldsymbol{H}$ is symmetric and positive semi-definite, we exploit the rearrangement property and invariance of the trace under similarity transformation to obtain the upper and lower bounds,

$$\lambda_{\max}\left(\bar{\boldsymbol{P}}\right)^{-2} \operatorname{Tr}\left(\boldsymbol{T}\boldsymbol{H}\boldsymbol{T}^T\right) \leq \operatorname{Tr}(\boldsymbol{G}\boldsymbol{H}) \leq \lambda_{\min}\left(\bar{\boldsymbol{P}}\right)^{-2} \operatorname{Tr}\left(\boldsymbol{T}\boldsymbol{H}\boldsymbol{T}^T\right). \tag{5.152}$$

Letting γ denote the middle quantity in Eq. 5.90, we apply the above result to each term in the numerator and denominator to obtain

$$\gamma \geq \min_{\substack{0 \leq k \leq K-1 \\ a \in \mathcal{M}}} \left(\frac{E(\mathcal{S}_k)}{E(\mathcal{S}_k \cup \{a\})}\right) \left(\frac{\sum_{i=1}^{N} \lambda_{\max}\left[\bar{\boldsymbol{P}}_i(\mathcal{S}_k \cup \{a\})\right]^{-2} \operatorname{Tr}\left[\boldsymbol{T}_i \boldsymbol{P}_i(\{a\}) \boldsymbol{T}_i^T\right]}{\sum_{i=1}^{N} \lambda_{\min}\left[\bar{\boldsymbol{P}}_i(\mathcal{S}_k)\right]^{-2} \operatorname{Tr}\left[\boldsymbol{T}_i \boldsymbol{P}_i(\{a\}) \boldsymbol{T}_i^T\right]}\right). \tag{5.153}$$

Bounding the sums from above and below gives

$$\gamma \geq \min_{\substack{0 \leq k \leq K-1 \\ a \in \mathcal{M}}} \left(\frac{E(\mathcal{S}_k)}{E(\mathcal{S}_k \cup \{a\})}\right) \left(\frac{\min_{1 \leq i \leq N} \lambda_{\min}\left[\bar{\boldsymbol{P}}_i(\mathcal{S}_k)\right]}{\max_{1 \leq i \leq N} \lambda_{\max}\left[\bar{\boldsymbol{P}}_i(\mathcal{S}_k \cup \{a\})\right]}\right)^2. \tag{5.154}$$

By Weyl's inequality, we observe that

$$\min_{1 \leq i \leq N} \lambda_{\min}\left[\bar{\boldsymbol{P}}_i(\mathcal{S}_k)\right] \geq \lambda_k, \quad \text{and} \tag{5.155}$$

$$\max_{1 \leq i \leq N} \lambda_{\max}\left[\bar{\boldsymbol{P}}_i(\mathcal{S}_k \cup \{a\})\right] \leq \Lambda_{k+1}, \tag{5.156}$$

which, together with the fact that $E(\mathcal{S}_k) \geq E(\mathcal{S}_k \cup \{a\})$ for every $a \in \mathcal{M}$ thanks to monotonicity, completes the proof of the second inequality in Eq. 5.90. □

Proof of Theorem. 5.4.11 (Greedy Upper Selection Performance). Consider the sets $\mathcal{S}_k$ selected by

Algorithm 1. By monotonicity of f, we have

$$f(\mathcal{S}^*) - f(\mathcal{S}_k) \leq f(\mathcal{S}^* \cup \mathcal{S}_k) - f(\mathcal{S}_k). \tag{5.157}$$

Since $\hat{f}_{\mathcal{S}_k}$ is a submodular upper bound, it follows that

$$f(\mathcal{S}^*) - f(\mathcal{S}_k) \leq h_{\mathcal{S}_k}(\mathcal{S}^* \setminus \mathcal{S}_k) \leq \sum_{a \in \mathcal{S}^* \setminus \mathcal{S}_k} h_{\mathcal{S}_k}(\{a\}). \tag{5.158}$$

By definition of Algorithm 1, we choose $a_{k+1} \in \mathcal{M} \setminus \mathcal{S}_k$ and form $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{a_{k+1}\}$ so that $h_{\mathcal{S}_k}(\{a\})/c(\{a\}) \leq h_{\mathcal{S}_k}(\{a_{k+1}\})/c(\{a_{k+1}\})$ for all $a \in \mathcal{M} \setminus \mathcal{S}_k$, therefore

$$f(\mathcal{S}^*) - f(\mathcal{S}_k) \leq \sum_{a \in \mathcal{S}^* \setminus \mathcal{S}_k} \frac{c(\{a\})}{c(\{a_{k+1}\})} h_{\mathcal{S}_k}(\{a_{k+1}\}) \leq \frac{C}{c(\{a_{k+1}\})} h_{\mathcal{S}_k}(\{a_{k+1}\}). \tag{5.159}$$

Using the definition of the submodularity gap ratio for the sequence of greedily chosen sets, we relate the increment of the submodular upper bound to the increment of the objective, yielding

$$f(\mathcal{S}^*) - f(\mathcal{S}_k) \leq \frac{C}{\Gamma_K c(\{a_{k+1}\})} \left(f(\mathcal{S}_{k+1}) - f(\mathcal{S}_k) \right). \tag{5.160}$$

Re-arranging we obtain the recursive relationship

$$\begin{aligned} f(\mathcal{S}^*) - f(\mathcal{S}_{k+1}) &\leq \left(1 - \Gamma_K \frac{c(\{a_{k+1}\})}{C} \right) \left(f(\mathcal{S}^*) - f(\mathcal{S}_k) \right) \\ &< e^{-\Gamma_K c(\{a_{k+1}\})/C} \left(f(\mathcal{S}^*) - f(\mathcal{S}_k) \right), \end{aligned} \tag{5.161}$$

where the second inequality follows from convexity of the exponential function and the fact that no measurement is free, i.e., $c(\{a\}) > 0 \ \forall a \in \mathcal{M}$. Iterating this bound starting from $f(\mathcal{S}^*) - f(\mathcal{S}_0) = f(\mathcal{S}^*)$ and recalling $\sum_{i=1}^{k} c(\{a_i\}) = c(\mathcal{S}_k)$ we conclude that

$$f(\mathcal{S}^*) - f(\mathcal{S}_k) < e^{-\Gamma_K c(\mathcal{S}_k)/C} f(\mathcal{S}^*), \quad \forall k \geq 1 \tag{5.162}$$

which can be re-arranged to produce the stated result. □

Proof of Theorem. 5.4.15 (Near-Minimum Cost Selection). The case when $K = 1$ is trivial. We start by observing that scaling f by any positive constant does not change the optimization problem Eq. 5.100 or the sequence of sets obtained by Algorithm 1. Therefore, let us work with the new

function defined by $\tilde{f} := \frac{1}{\rho_{\min}} f$ which satisfies

$$\frac{\tilde{f}(\mathcal{S}_k) - \tilde{f}(\mathcal{S}_{k-1})}{c(\mathcal{S}_k) - c(\mathcal{S}_{k-1})} \geq 1, \qquad \forall k \geq 1. \tag{5.163}$$

For any $\phi > 0$ and $1 \leq k \leq K$, the conclusion of Theorem 5.4.11 with $C = C^*$ can be used to show that

$$c(\mathcal{S}_k) \geq \frac{C^*}{\Gamma} \ln\left(\frac{\tilde{f}(\mathcal{M})}{\phi}\right) \quad \Rightarrow \quad \phi > \tilde{f}(\mathcal{M}) - \tilde{f}(\mathcal{S}_k) \quad \Rightarrow \quad \phi > c(\mathcal{S}_K) - c(\mathcal{S}_k), \tag{5.164}$$

where the second implication holds because

$$\tilde{f}(\mathcal{M}) - \tilde{f}(\mathcal{S}_k) = \sum_{i=k+1}^{K} \left(\tilde{f}(\mathcal{S}_i) - \tilde{f}(\mathcal{S}_{i-1})\right) \geq \sum_{i=k+1}^{K} (c(\mathcal{S}_i) - c(\mathcal{S}_{i-1})) = c(\mathcal{S}_K) - c(\mathcal{S}_k). \tag{5.165}$$

If

$$c(\mathcal{S}_K) \geq \frac{C^*}{\Gamma} \ln\left(\frac{\tilde{f}(\mathcal{M})}{\phi}\right) \tag{5.166}$$

then there is an index k with $1 \leq k \leq K$ such that

$$c(\mathcal{S}_k) \geq \frac{C^*}{\Gamma} \ln\left(\frac{\tilde{f}(\mathcal{M})}{\phi}\right) > c(\mathcal{S}_{k-1}), \tag{5.167}$$

hence

$$c(\mathcal{S}_k) < \frac{C^*}{\Gamma} \ln\left(\frac{\tilde{f}(\mathcal{M})}{\phi}\right) + c(\mathcal{S}_k) - c(\mathcal{S}_{k-1}) \leq \frac{C^*}{\Gamma} \ln\left(\frac{\tilde{f}(\mathcal{M})}{\phi}\right) + \max_{a \in \mathcal{S}_K} c(\{a\}). \tag{5.168}$$

By Eq. 5.164 we therefore obtain

$$c(\mathcal{S}_K) < \phi + \frac{C^*}{\Gamma} \ln\left(\frac{\tilde{f}(\mathcal{M})}{\phi}\right) + \max_{a \in \mathcal{S}_K} c(\{a\}) \tag{5.169}$$

for all $\phi > 0$. Choosing the minimizing value $\phi = C^*/\Gamma$ completes the proof. □

Proof of Theorem 5.5.1 (dual certificate of recovery). Here, we provide the proof exactly as it is given in [62]. For simplicity of notation, we shall drop the subscript $\mathcal{S}$ since it will not change throughout the proof. Suppose that $\boldsymbol{x}+\boldsymbol{h}$ is an optimizer for the ℓ^1 minimization problem Eq. 5.110, then it suffices to prove that $\boldsymbol{h} = \mathbf{0}$. First, we suppose that $\boldsymbol{h}$ has a nonzero element $h_i \neq 0$ with

$i \notin \text{supp}(\boldsymbol{x})$. We observe that the vector $\boldsymbol{w} \in \mathbb{R}^n$ defined element-wise by

$$\begin{cases} w_i = \text{sgn}(x_i) & \text{if } x_i \neq 0 \\ w_i = \text{sgn}(h_i) & \text{otherwise} \end{cases} \tag{5.170}$$

is a member of the sub-gradient set

$$\partial \|\boldsymbol{x}\|_1 = \{\boldsymbol{g} \in \mathbb{R}^n \ : \ \|\boldsymbol{x} + \boldsymbol{h}\|_1 \geq \|\boldsymbol{x}\|_1 + \langle \boldsymbol{g}, \ \boldsymbol{h} \rangle \quad \forall \boldsymbol{h} \in \mathbb{R}^n\}. \tag{5.171}$$

By definition of the sub-gradient and the fact that $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} = \boldsymbol{M}(\boldsymbol{x} + \boldsymbol{h})$, i.e., $\boldsymbol{M}\boldsymbol{h} = \boldsymbol{0}$, we obtain

$$\|\boldsymbol{x}\|_1 \geq \|\boldsymbol{x} + \boldsymbol{h}\|_1 \geq \|\boldsymbol{x}\|_1 + \langle \boldsymbol{w}, \ \boldsymbol{h} \rangle = \|\boldsymbol{x}\|_1 + \langle \boldsymbol{w} - \boldsymbol{M}^T \boldsymbol{v}, \ \boldsymbol{h} \rangle, \tag{5.172}$$

and so $\langle \boldsymbol{w} - \boldsymbol{M}^T \boldsymbol{v}, \ \boldsymbol{h} \rangle \leq 0$. However, letting $\boldsymbol{u} = \boldsymbol{M}^T \boldsymbol{v}$, we find

$$\begin{aligned} \langle \boldsymbol{w} - \boldsymbol{M}^T \boldsymbol{v}, \ \boldsymbol{h} \rangle &= \sum_{i \notin \text{supp}(\boldsymbol{x})} (\text{sgn}(h_i) h_i - u_i h_i) \\ &= \sum_{i \notin \text{supp}(\boldsymbol{x})} (|h_i| - u_i h_i) \geq \sum_{i \notin \text{supp}(\boldsymbol{x})} (1 - |u_i|) \, |h_i| > 0, \end{aligned} \tag{5.173}$$

which is a contradiction. Therefore, $h_i = 0$ for every $i \notin \text{supp}(\boldsymbol{x})$. Let $S = \text{supp}(\boldsymbol{x})$ and let $\boldsymbol{h}_S$ denote the restriction of $\boldsymbol{h}$ to the subset of its elements in S. We also let $\boldsymbol{M}_S$ denote the sub-matrix of $\boldsymbol{M}$ formed by retaining the columns in S. Moreover, since $\text{supp}(\boldsymbol{h}) \subset S$, we have $\boldsymbol{M}_S \boldsymbol{h}_S = \boldsymbol{0}$, which implies that $\boldsymbol{h}_S = \boldsymbol{0}$ since $\boldsymbol{M}_S$ is injective by assumption. Therefore, we conclude that $\boldsymbol{h} = 0$. $\square$

Proof of Theorem 5.5.2 (sampling to probably recover most states). For any subset $\mathcal{S}' \subset \mathcal{M}$ of sensors, we have

$$\mu(\mathcal{X}'_{\mathcal{S}}) = \mathbb{E}\left[\mathbb{1}\left\{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}'}\right\}\right] \tag{5.174}$$

We observe that for our selected set of sensors $\mathcal{S}$, we have $\boldsymbol{x}_i \in \mathcal{X}_{\mathcal{S}}$ for every $i = 1, \ldots, m$, and so the empirical average

$$A_m(\mathcal{S}) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left\{\boldsymbol{x}_i \in \mathcal{X}_{\mathcal{S}}\right\} = 1. \tag{5.175}$$

If we can bound the amount by which $A_m(\mathcal{S}')$ over-estimates $\mu(\mathcal{X}'_{\mathcal{S}})$ for every $\mathcal{S}' \subset \mathcal{M}$ of size at most L with high probability, then we automatically obtain a lower bound for $\mu(\mathcal{X}_{\mathcal{S}})$ using the optimal sensors $\mathcal{S}$ that holds with high probability. Fixing $\mathcal{S}' \subset \mathcal{M}$, we recognize that $\mathbb{1}\left\{\boldsymbol{x}_i \in \mathcal{X}_{\mathcal{S}}\right\}$ are independent, identically distributed Bernoulli random variables, and so Hoeffding's inequality

yields

$$\mathbb{P}\left\{A_m(\mathcal{S}') - \mu(\mathcal{X}'_{\mathcal{S}}) \geq \delta\right\} \leq e^{-2m\delta^2}. \tag{5.176}$$

Unfixing $\mathcal{S}'$ among sets with at most L elements using the union bound, we obtain

$$\mathbb{P} \bigcup_{\substack{\mathcal{S}' \subset \mathcal{M}: \\ |\mathcal{S}'| \leq L}} \left\{A_m(\mathcal{S}') - \mu(\mathcal{X}'_{\mathcal{S}}) \geq \delta\right\} \leq \sum_{\substack{\mathcal{S}' \subset \mathcal{M}: \\ |\mathcal{S}'| \leq L}} e^{-2m\delta^2}. \tag{5.177}$$

If $|\mathcal{M}| = M$, then the number of subsets of $\mathcal{M}$ with at most L elements is bounded by

$$|\{\mathcal{S} \subseteq \mathcal{M} \ : \ |\mathcal{S}| \leq L\}| = \sum_{k=1}^{L} \binom{M}{k} \leq \sum_{k=1}^{L} \frac{M^k}{k!} \leq L\frac{M^L}{L!} = \frac{M^L}{(L-1)!} \leq M^L. \tag{5.178}$$

Substituting this estimate into the union bound, we obtain

$$\mathbb{P} \bigcup_{\substack{\mathcal{S}' \subset \mathcal{M}: \\ |\mathcal{S}'| \leq L}} \left\{A_m(\mathcal{S}') - \mu(\mathcal{X}'_{\mathcal{S}}) \geq \delta\right\} \leq e^{L\log(M) - 2m\delta^2} < p. \tag{5.179}$$

when

$$m \geq \frac{1}{2\delta^2}\big(L\log(M) - \log(p)\big). \tag{5.180}$$

Recalling that $A_m(\mathcal{S}) = 1$ proves the desired result, namely that $\mu(\mathcal{X}_{\mathcal{S}}) \leq 1 - \delta$ with probability less than p. $\square$

Chapter 6

Conclusion and Outlook

6.1 Conclusion

This thesis presents several advances in data-driven modeling and sensing techniques for high-dimensional nonlinear systems such as those arising from discretized fluid flow simulations. There are three key takeaways from this work. The first is that reduced-order models (ROMs) based on nonlinear manifolds can provide significant dimensionality reduction and improved representational power when compared to models based on linear subspaces. The reduction is especially significant when modeling coherent structures that translate in space, such as advecting vortices in fluid dynamics. In Section 3.4 we developed an autoencoder for projecting dynamical systems onto such nonlinear manifolds. In Chapter 5, we demonstrated how low-dimensional nonlinear structure can be leveraged to identify minimal collections of sensors, and how linear techniques fail to do so. The second key takeaway is that nonlinear coherent structures for dimensionality reduction can be identified using simulation or experimental data. This is especially useful for systems operating in regimes that are too far away from an equilibrium to apply perturbative techniques.

The final takeaway is that while simulation and experimental data is useful, it does not provide the required sensitivity information about the system when the state dimension exceeds the number of snapshots collected from the system. In Chapter 3.3 we describe how improperly modeling the system's sensitivity can result in poor predictions for systems in which low-energy features play a dynamically significant role. We conclude that the linearized adjoint of the original system is an indispensable tool for modeling the system's sensitivity. We describe several ways in which the adjoint can be leveraged for reduced-order modeling. The simplest approach is to construct a

projection into a truncated coordinate system where a forward and an adjoint covariance matrix are balanced. This generalization of balanced POD [225] may serve as a standard pre-processing step for nearly any subsequent data-driven modeling technique such as (E/K)DMD, SINDy, RNNs, nonlinear Galerkin projection, etc.. Using these key takeaways, a variety of data-driven reduced-order modeling and sensing techniques beyond what we have presented may be conceived.

6.2 Outlook and future work

The most important next step will be to use the data-driven modeling techniques developed here for feedback control. However, using a data-driven model for feedback control will likely present its own challenges. This is because feedback control can significantly change the distribution of state and input time histories away from the distribution on which the original model was trained. The learned model may no longer be valid near the closed loop trajectories if they do not resemble the training data. One solution is to use enough training data and a model of sufficiently high complexity so that every relevant open or closed-loop trajectory can be modeled accurately. However, this is unsatisfying and possibly very costly as systems with complex dynamics will have to be exhaustively sampled. We anticipate that ROMs designed specifically for the closed-loop setting will achieve higher performance in terms of accuracy and dimension reduction using less data. Some ways we can begin to design these models are described below.

One approach to improve the accuracy of nonlinear ROMs in the actuated setting is to constrain them in the neighborhood of a fixed point or trajectory to be stabilized. For instance, we can constrain the model to agree with the $\mathcal{H}_2$-optimal projection or with balanced truncation of the linearized system about a fixed point. In the case of the nonlinear projection method described in Section 3.4, this may entail imposing additional constraints on the autoencoder's weights, or designing a different architecture that provides only high-order corrections to a given linear projection. In addition to constraining the model near a fixed point, it may also be helpful to constrain the model so that it respects known symmetries and conservation laws of the original system.

Another class of approaches for data-driven closed-loop reduced-order modeling entails optimizing the parameters of the ROM in the closed-loop setting. One approach is to obtain a sequence of models, where the next model in the sequence is trained on data collected from the closed-loop system with feedback provided based on the previous model. If one has access to the full-order model (FOM) and its linearized adjoint, then another option is to directly optimize the parameters defining the ROM based on its performance in the closed-loop setting. In particular, the optimal

open loop input signal computed for the ROM can be differentiated with respect to the parameters of the ROM by differentiating the Euler-Lagrange optimality condition and solving the resulting linear system. Since the gradient of the cost function may be computed with respect to the input of the FOM, the chain rule can be applied to find the gradient of the cost function with respect to the parameters of the ROM when its optimal input signal is fed into the FOM.

Finally, in settings where the governing equations are inaccessible, it may be possible to leverage tools from reinforcement learning to construct optimal ROMs and the resulting control policies. In particular, optimal policies for simple ROMs may be a useful parametric class of policies for reinforcement learning. Here, the parameters defining a given policy are the parameters of a ROM. Alternatively, the optimal value function (or Q function) may be parametrized using the optimal value function (or Q function) associated with a ROM. Here, one would optimize the parameters of the model so that its optimal value function (or Q function) closely approximates that of the FOM. Of course both approaches are made possible via differentiation of the optimal input signal for the ROM with respect to the parameters that define the ROM.

Part II

Selected Papers

Chapter 7

Overview

In this part we reproduce selected papers by the author and collaborators that provide additional details and results pertaining to the methods summarized in Part I. The papers are:

- **Chapter 8:** S. E. Otto, A. Padovan, C. W. Rowley, "Optimizing Oblique Projections for Nonlinear Systems using Trajectories", submitted to SISC, 2021 [192]
- **Chapter 9:** S. E. Otto and C. W. Rowley, "Linearly-Recurrent Autoencoder Networks for Learning Dynamics", published in SIADS, 2019 [194]
- **Chapter 10:** S. E. Otto and C. W. Rowley, "Inadequacy of Linear Methods for Minimal Sensor Placement and Feature Selection in Nonlinear Systems; a New Approach Using Secants", submitted to JNLS, 2021 [195]

These papers have been selected because of their relevance to the discussion in Part I and because their main contributions are primarily attributable to S. E. Otto. Other papers by the author that are not reproduced here include

- S. E. Otto and C. W. Rowley, "A Discrete Empirical Interpolation Method for Interpretable Immersion and Embedding of Nonlinear Manifolds", arXiv pre-print, 2019 [193]
- S. Peitz, S. E. Otto, and C. W. Rowley, "Data-driven model predictive control using interpolated Koopman generators", published in SIADS, 2020 [202]
- A. Padovan, S. E. Otto, and C. W. Rowley, "Analysis of amplification mechanisms and cross-frequency interactions in nonlinear flows via the harmonic resolvent", published in JFM, 2020 [197]

- S. E. Otto and C. W. Rowley, "Koopman operators for estimation and control of dynamical systems", published in Ann. Rev. Contr. Robot. Aut. Sys., 2021. [196]

7.1 Author contributions

Here, we summarize the specific contributions by S. E. Otto and co-authors to each of the papers selected for reproduction in this part.

7.1.1 Optimizing Oblique Projections for Nonlinear Systems using Trajectories

- S. E. Otto had the idea to optimize oblique projections for nonlinear systems based on data. He initially developed the biorthogonal manifold machinery presented in Section 3.4.2 for this purpose.
- C. W. Rowley suggested working with the Grassmann manifold rather than the biorthogonal manifold, which provided a more elegant description of the projection operators and enabled the use of existing machinery for this manifold.
- S. E. Otto conceptualized and proved all of the theoretical results, including convergence theorems for the geometric conjugate gradient algorithm. He also wrote the sections of the paper concerning the Grassmann manifold and optimization on it.
- C. W. Rowley proposed the toy model and S. E. Otto produced the results for this example.
- A. Padovan was responsible for all aspects concerning the jet flow, including producing the results on this example, writing the corresponding section of the paper, and appendices about the adjoint Navier stokes equations.
- C. W. Rowley provided useful feedback and revisions of the paper that greatly improved its overall clarity.

7.1.2 Linearly-Recurrent Autoencoder Networks for Learning Dynamics

- S. E. Otto had the idea to use autoencoders to approximate the Koopman operator and came up with the resulting LRAN architecture and training strategy.

- S. E. Otto had the idea to use balanced truncation to reduce the dimension of over-specified E/KDMD-based models, and to reconstruct the state nonlinearly.
- C. W. Rowley guided the selection of examples to be included in the paper and contributed to the paper's overall organization
- C. W. Rowley edited several drafts of the paper and helped enormously to improve the clarity.
- William Eggert (mentioned in the acknowledgements) worked with S. E. Otto to write an initial version of the LRAN code for a class project. S. E. Otto later built upon this code and prepared the final examples used in the paper.
- C. W. Rowley mentioned trying to make the encoder a left-inverse of the decoder, which was initially attempted by S. E. Otto without success and not included in the paper. Later on in 2021, S. E. Otto realized that this could be accomplished by optimizing on biorthogonal manifolds, see Section 3.4.1 in Part I.

7.1.3 Inadequacy of Linear Methods for Minimal Sensor Placement and Feature Selection in Nonlinear Systems; a New Approach Using Secants

- S. E. Otto had the idea to base greedy sensor placement techniques on secants and developed the three submodular objectives presented in the paper.
- S. E. Otto conceptualized and proved all of the theoretical results in the paper, including the results pertaining to down-sampling.
- S. E. Otto selected the examples for the paper and produced the results.
- C. W. Rowley provided conversations and suggestions that helped to sharpen the examples and organization of the paper. For instance, he pointed out the connection with period-doubling.
- C. W. Rowley read and revised several drafts of the paper and provided suggestions which greatly improved the clarity of the presentation.

Chapter 8

Optimizing Oblique Projections for Nonlinear Systems using Trajectories

Samuel E. Otto, Alberto Padovan, and Clarence W. Rowley

Reduced-order modeling techniques, including balanced truncation and $\mathcal{H}_2$-optimal model reduction, exploit the structure of linear dynamical systems to produce models that accurately capture the dynamics. For nonlinear systems operating far away from equilibria, on the other hand, current approaches seek low-dimensional representations of the state that often neglect low-energy features that have high dynamical significance. For instance, low-energy features are known to play an important role in fluid dynamics where they can be a driving mechanism for shear-layer instabilities. Neglecting these features leads to models with poor predictive accuracy despite being able to accurately encode and decode states. In order to improve predictive accuracy, we propose to optimize the reduced-order model to fit a collection of coarsely sampled trajectories from the original system. In particular, we optimize over the product of two Grassmann manifolds defining Petrov-Galerkin projections of the full-order governing equations. We compare our approach with existing methods such as proper orthogonal decomposition and balanced truncation-based Petrov-Galerkin projection, and our approach demonstrates significantly improved accuracy both on a nonlinear toy model and on an incompressible (nonlinear) axisymmetric jet flow with $69,000$ states.

8.1 Introduction

Accurate low-dimensional models of physical processes enable a variety of important scientific and engineering tasks to be carried out. Such models can be used to make real-time forecasts as well as to shed light on the underlying physics through detailed analysis of the resulting dynamical system. The models can also serve as a building block for filters that estimate the state of the system from incomplete measurements and to design control laws to achieve desired behaviors from the system. However, many real-world systems like complex fluid flows in the atmosphere as well as around and inside aircraft are governed by extremely high-dimensional nonlinear systems — properties that make tasks like real-time forecasting, state estimation, and control computationally prohibitive using the original governing equations. Fortunately, the behavior of these systems is frequently dominated by coherent structures and patterns [32] that may be modeled with equations whose dimension is much smaller [250, 114]. The goal of "reduced-order modeling" is to obtain simplified models that are suitable for forecasting, estimation, and control from the vastly more complicated governing equations provided by physics. For reviews of modern techniques, see [14], [20] and [228]. For a striking display of coherent structures in turbulence, see the shadowgraphs in G. L. Brown and A. Roshko [32].

When the system of interest is operating close to an equilibrium point, the governing equations are accurately approximated by their linearization about the equilibrium. In this case, a variety of sophisticated and effective reduced-order modeling techniques can be applied with guarantees on the accuracy of the resulting low-dimensional model [10, 20]. Put simply, linearity provides an elegant and complete characterization of the system's trajectories in response to inputs, disturbances, and initial conditions that can be exploited to build simplified models whose trajectories closely approximate the ones from the original system. For instance, the balanced truncation method introduced by B. Moore [182] yields a low-dimensional projection of the original system that simultaneously retains the most observable and controllable states of the system and provides bounds on various measures of reduced-order model error [10]. A computationally efficient approximation called Balanced Proper Orthogonal Decomposition (BPOD) [225] is suitable for high-dimensional fluid flow applications. Another approach is to find a stable reduced-order model (ROM) that is as close as possible to a stable full-order model (FOM) with respect to the $\mathcal{H}_2$ norm. Algorithms like the Iterative Rational Krylov Algorithm (IRKA) [105] are based on satisfying necessary conditions for $\mathcal{H}_2$-optimality.

Various generalizations of linear model reduction techniques have also been developed for bilinear

[14, 17, 88] and quadratic bilinear systems [18, 19] based on truncated Volterra series expansion of the output. If enough terms are retained in the series expansions, these methods can yield reduced systems that approximate the response to arbitrary input signals. However, the computational cost increases with the number of terms retained, making them difficult to apply to fluid flows whose state dimensions can easily exceed 10^5.

One commonality among the above model reduction approaches based on direct input-output relationships is that they lead to reduced-order models that capture the most energetic features as well as any low-energy features that nonetheless significantly influence the dynamics at future times [20, 225]. These small, but dynamically significant features are known to play an important role in driving the growth of instabilities in "shear flows" such as mixing layers and jets. Linearizations of these shear flows often result in non-normal systems, which can exhibit large transient growth in response to low-energy perturbations [264, 242]. Some successful approaches [13, 6, 118, 120] have involved oblique projections of the nonlinear dynamics onto subspaces identified from the dynamics linearized about an equilibrium. However, this approach is often not satisfactory since the linearized dynamics become inaccurate as the state moves away from the equilibrium and nonlinear effects become significant. In this paper we illustrate how such nonlinear effects can cause reduced-order models obtained using the above approach to perform poorly, for instance on a simple three-dimensional system as well as on a high-dimensional axisymmetric jet flow.

When dealing with nonlinear systems operating far away from equilibria, nonlinear model reduction approaches tend to follow a two-step process: first identify a set, typically a smooth manifold or a subspace, near which the state of the system is known to lie, then model the dynamics in this set either by a projection of the governing equations or by a black-box data-driven approach. The most common approach to identify a candidate subspace is Proper Orthogonal Decomposition (POD), whose application to the study of complex fluid flows was pioneered by J. L. Lumley [163]. The dynamics may also be projected onto nonlinear manifolds using "nonlinear Galerkin" methods [168, 219]. Recently, more sophisticated manifold learning techniques like deep convolutional autoencoders have also been used [150].

The main obstacle encountered by the manifold-learning-based approaches described above is the presence of dynamically-significant low-energy features. Since POD and even sophisticated nonlinear manifold learning techniques like convolutional autoencoders aim to accurately reduce and reconstruct states, they will neglect features whose contribution to the overall state has a sufficiently small magnitude. But, as we mentioned earlier, we should not neglect all of the low-energy features since some can have a large influence on the dynamics. In fact, in our jet flow example, we shall

see that a model with 50 POD modes that together capture 99.6% of the energy still yields poor predictions that rapidly diverge from the full-order model.

In order to capture significant low-energy features, while remaining tractable for very large-scale systems like fluid flows, we shall optimize an oblique projection operator defining the reduced-order model with the objective of reproducing a collection of trajectories sampled from the original system. In particular, we seek to minimize the sum of squared errors between the trajectories predicted by the model and those collected from the full system. In this framework, oblique projection operators of a fixed dimension are identified with pairs of subspaces that meet a transversality condition. Recalling that the collection of all subspaces of a given dimension can be endowed with the structure of a Riemannian manifold called the Grassmann manifold [1, 16], we show that the pairs of subspaces that define oblique projection operators are an open, dense, and connected subset of the product of two such Grassmann manifolds and we provide conditions for the existence of a minimizer. The optimization is performed using the Riemannian conjugate gradient algorithm introduced by H. Sato [235] with retraction and vector transport defined in Absil et al. [1], and we provide general conditions under which the algorithm is guaranteed to converge to a local optimum.

Related techniques based on optimizing projection subspaces have been used to produce $\mathcal{H}_2$-optimal reduced-order models for linear and bilinear systems. Most approaches focus on optimizing orthogonal projection operators over a single Grassmann manifold [288, 238, 122] or an orthogonal Stiefel manifold [290, 238, 274, 291, 287]. On the other hand, an alternating minimization technique over the two Grassmann manifolds defining an oblique projection is proposed by T. Zeng and C. Lu [298] for $\mathcal{H}_2$-optimal reduction of linear systems. For systems with quadratic nonlinearities, Y.-L. Jiang and K.-L. Xu [122] present an approach to optimize orthogonal projection operators based on the same truncated generalization of the $\mathcal{H}_2$ norm used by P. Benner et al. [19]. Our approach differs from the ones mentioned above in that it may be used to find optimal reduced-order models based on oblique projections for general very high-dimensional nonlinear systems based on sampled trajectories.

8.2 Projection-Based Reduced-Order Models

Consider a physical process, modeled by an input-output dynamical system

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, x &= f(x,u), \qquad x(t_0) = x_0 \\ y &= g(x) \end{aligned} \tag{8.1}$$

on a finite-dimensional real inner product space $\mathcal{X} = (\mathbb{R}^n, \langle \cdot,\ \cdot \rangle)$ with outputs y in $\mathbb{R}^m$ equipped with the usual inner product. We shall often refer to Eq. 8.1 as the full-order model (FOM). Our goal is to use one or more discrete-time histories of observations $y_l = y(t_l)$ at sample times $t_0 < \cdots < t_{L-1}$ in order to learn the key dynamical features of Eq. 8.1 and produce a reduced-order model (ROM) that captures these effects. Throughout the paper we assume that

Assumption 8.2.1. *The functions $(x,t) \mapsto f(x,u(t))$ and $x \mapsto g(x)$ in Eq. 8.1, along with their first-order partial derivatives with respect to x, are continuous.*

We shall use our observation data to learn an r-dimensional subspace V of $\mathbb{R}^n$ in which to represent the state of the system Eq. 8.1. Since $f(x,u)$ might not lie in V when $x \in V$, we shall also learn another r-dimensional subspace W of $\mathbb{R}^n$ that, together with V, uniquely defines an oblique projection operator $P_{V,W} : \mathbb{R}^n \to V$ according to

$$\langle w,\ P_{V,W} x \rangle = \langle w,\ x \rangle\,, \quad \forall w \in W, \qquad \forall x \in \mathbb{R}^n \tag{8.2}$$

when no nonzero element of W is orthogonal to V. We let $\mathcal{P}$ denote the set of such subspaces pairs (V,W) with the property that no nonzero element of W is orthogonal to V and give some equivalent definitions of this set in Proposition 8.2.2.

Proposition 8.2.2 (Subspaces that Define Oblique Projections)**.** *Let V and W be subspaces of $\mathbb{R}^n$ with $\dim V = \dim W = r$, and let $\Phi, \Psi \in \mathbb{R}^{n \times r}$ be matrices such that $V = \operatorname{Range} \Phi$ and $W = \operatorname{Range} \Psi$. Let Ψ^* denote the adjoint of Ψ, viewed as a linear operator from $\mathbb{R}^r$ with the Euclidean inner product into the state space $\mathcal{X}$ with its own inner product. Then the following are equivalent:*

1. *no nonzero element of W is orthogonal to V;*
2. *no nonzero element of V is orthogonal to W;*
3. $\det(\Psi^* \Phi) \neq 0$*;*
4. *for every $x \in \mathbb{R}^n$ there exists a unique $\hat{x} \in V$ such that*

$$\langle w,\ x \rangle = \langle w,\ \hat{x} \rangle \qquad \forall w \in W. \tag{8.3}$$

Proof. The proof is an exercise in linear algebra, so we give it in Appendix 8.F. □

Applying the projection defined by $(V, W) \in \mathcal{P}$ to the full-order model Eq. 8.1, we obtain a Petrov-Galerkin reduced-order model whose state $\hat{x} \in V$ evolves according to

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\hat{x} = P_{V,W} f(\hat{x}, u), \qquad \hat{x}(0) = P_{V,W} x_0, \tag{8.4}$$

with observations given by $\hat{y} = g(\hat{x})$. The two subspaces V, W uniquely define the projection $P_{V,W}$ and the reduced-order model Eq. 8.4.

Let $L_y : \mathbb{R}^m \to [0, +\infty)$ be a smooth penalty function for the difference between each observation y_l and the model's prediction $\hat{y}(t_l)$. Let us also introduce a smooth nonnegative-valued function $\rho(V, W)$, to be defined precisely in Section 8.3, that will serve as regularization by preventing minimizing sequences of subspaces (V, W) from approaching points outside the set $\mathcal{P}$ in which valid Petrov-Galerkin projections can be defined. Using this regularization with a weight $\gamma > 0$ allows us to seek a minimum of the cost defined by

$$J(V, W) = \frac{1}{L} \sum_{l=0}^{L-1} L_y \left(\hat{y}(t_l) - y_l\right) + \gamma \rho(V, W) \tag{8.5}$$

over all pairs of r-dimensional subspaces (V, W), subject to the reduced-order dynamics Eq. 8.4. Here we shall consider the case when there is a single trajectory generated from a known initial condition since it will be easy to handle multiple trajectories from multiple known initial conditions once we understand the single trajectory case. The cost function (8.5) defines an optimization problem, and in the following section we define a suitable regularization function ρ and develop a technique for iteratively solving this problem.

8.3 Optimization Domain, Representatives, and Regularization

The set containing all r-dimensional subspaces of $\mathbb{R}^n$ can be endowed with the structure of a compact Riemannian manifold called the Grassmann manifold, which has dimension $nr - r^2$ and is denoted $\mathcal{G}_{n,r}$. Therefore, our optimization problem entails minimizing the cost given by Eq. 8.5 over the subset $\mathcal{P}$ of the product manifold $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ on which oblique projection operators are defined according to Proposition 8.2.2. The goal of this section will be to characterize the topology of the set $\mathcal{P}$ and to introduce an appropriate regularization function ρ so that we may instead consider the unconstrained minimization of Eq. 8.5 over $\mathcal{M}$. We also describe how to work with matrix

representatives of the relevant subspaces that can be stored in a computer.

8.3.1 Grassmann Manifold and Representatives of Subspaces

First we describe some basic properties of the Grassmann manifold that can be found in Absil et al. [1]. If $\mathbb{R}_*^{n,r}$ denotes the smooth manifold of $n \times r$ matrices with linearly independent columns, then $\mathcal{G}_{n,r}$ can be identified with the quotient manifold of $\mathbb{R}_*^{n,r}$ defined by identifying matrices with the same span. That is, two matrices $X, Y \in \mathbb{R}_*^{n,r}$ are defined to be equivalent if $\operatorname{Range} X = \operatorname{Range} Y$, i.e., there is an invertible matrix $M \in GL_r$, the general linear group, such that $X = YM$. The equivalence class of a matrix $X \in \mathbb{R}_*^{n,r}$ is defined by

$$[X] = \{Y \in \mathbb{R}_*^{n,r} \ : \ \operatorname{Range} X = \operatorname{Range} Y\}, \tag{8.6}$$

and the set of these equivalence classes is the quotient space $\mathbb{R}_*^{n,r}/GL_r$. Since the action of GL_r on $\mathbb{R}_*^{n,r}$ defining a change of basis $GL_r \times \mathbb{R}_*^{n,r} \to \mathbb{R}_*^{n,r} : (M, X) \mapsto XM$ is free and proper, it follows from the quotient manifold theorem (Theorem 21.10 in [149]) that $\mathbb{R}_*^{n,r}/GL_r$ is a smooth manifold and the quotient map $X \mapsto [X]$ is a smooth submersion. The Grassmann manifold can be identified with $\mathbb{R}_*^{n,r}/GL_r$ since each subspace $V \in \mathcal{G}_{n,r}$ corresponds to the unique equivalence class $[X] \in \mathbb{R}_*^{n,r}/GL_r$ whose elements all span V and vice-versa.

The identification of the Grassmann manifold with $\mathbb{R}_*^{n,r}/GL_r$ is very useful since we wish to optimize the subspaces V and W using a computer that can't store abstract subspaces in memory. Instead, we have to work with representatives of these subspaces given by pairs of $n \times r$ matrices $\Phi, \Psi \in \mathbb{R}_*^{n,r}$ such that $V = \operatorname{Range} \Phi$ and $W = \operatorname{Range} \Psi$. That is, we aim to optimize over the product manifold $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ by relying on representatives in the so called "structure space" $\bar{\mathcal{M}} = \mathbb{R}_*^{n,r} \times \mathbb{R}_*^{n,r}$. The "canonical projection" map $\pi : \bar{\mathcal{M}} \to \mathcal{M}$ is defined by

$$\pi : (\Phi, \Psi) \mapsto ([\Phi], [\Psi]), \tag{8.7}$$

and it is clear that any representatives $(\Phi, \Psi) \in \bar{\mathcal{M}}$ of $(V, W) \in \mathcal{M}$ must satisfy $(V, W) = \pi(\Phi, \Psi)$. For a pair of subspaces $(V, W) \in \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$, the set of all representatives $(\Phi, \Psi) \in \mathbb{R}_*^{n,r} \times \mathbb{R}_*^{n,r}$ such that $\pi(\Phi, \Psi) = (V, W)$ is given by the pre-image $\pi^{-1}(V, W)$. The canonical projection map is a surjective submersion since its component maps $\Phi \mapsto [\Phi]$ and $\Psi \mapsto [\Psi]$ are surjective submersions. This key fact yields the following useful result:

Lemma 8.3.1 (Regularity via Representatives)**.** *Let $\mathcal{N}$ be another smooth manifold and let F :*

$\mathcal{M} \to \mathcal{N}$ be a function. Then F is continuous if and only if $F \circ \pi$ is continuous. Furthermore, F is smooth if and only if $F \circ \pi$ is smooth.

Proof. The "only if" directions are obvious since the composition of continuous (resp. smooth) functions is continuous (resp. smooth). The converse statements follow immediately from the local submersion theorem [106] for π, which provides local charts on which F can be expressed by restricting $F \circ \pi$ to a coordinate slice. □

As before, we let $\mathcal{P}$ denote the set of subspace pairs that satisfy the conditions in Proposition 8.2.2, and suppose that $(V, W) \in \mathcal{P}$ and $(\Phi, \Psi) \in \pi^{-1}(V, W)$ are a choice of representatives. We recall that Ψ^* denotes the adjoint of Ψ viewed as a linear operator from $\mathbb{R}^r$ into the state space $\mathcal{X}$. Then it is clear from Proposition 8.2.2 that the $r \times r$ matrix $\Psi^*\Phi$ is invertible and the oblique projection operator corresponding to (V, W) is given by

$$P_{V,W} = \Phi(\Psi^*\Phi)^{-1}\Psi^*. \tag{8.8}$$

We also observe that Eq. 8.8 is independent of the choice of representatives $(\Phi, \Psi) \in \pi^{-1}(V, W)$ — as it should be, given that $P_{V,W}$ was originally defined by Eq. 8.2 in terms of abstract subspaces alone. Using the representatives and an r-dimensional state z defined by $\hat{x} = \Phi z \in V$, we obtain a representative of the reduced-order model Eq. 8.4 given by

$$\boxed{\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, z &= (\Psi^*\Phi)^{-1}\Psi^* f(\Phi z, u) =: \tilde{f}\left(z, u;\ (\Phi, \Psi)\right), \qquad z(0) = (\Psi^*\Phi)^{-1}\Psi^* x_0 \\ \hat{y} &= g(\Phi z) =: \tilde{g}\left(z;\ (\Phi, \Psi)\right), \end{aligned}} \tag{8.9}$$

that can be simulated on a computer. Observe that the output $\hat{y}(t)$ of *Eq.* 8.9 depends only on the subspaces (V, W), and not on the representatives (Φ, Ψ) we choose.

Consequently, any function of (Φ, Ψ) that depends only on the output $\hat{y}(t)$ of Eq. 8.9 can be viewed as a function on $\mathcal{M}$ composed with the canonical projection π. Hence, we can evaluate our cost function Eq. 8.5 for a subspace pair (V, W) by computing

$$\bar{J}(\Phi, \Psi) = J(\pi(\Phi, \Psi)) \tag{8.10}$$

for any choice of represenatives $(\Phi, \Psi) \in \pi^{-1}(V, W)$, that is, by evaluating the sum in Eq. 8.5 using the output $\hat{y}(t)$ generated by Eq. 8.9. Moreover, Lemma 8.3.1 tells us that J is smooth if and only if $\bar{J}$ is smooth.

Remark 8.3.2 (Convenient Representatives)**.** *One choice of representatives that is especially convenient to work with are biorthonormal pairs, that is,* $(\Phi, \Psi) \in \pi^{-1}(V, W)$ *such that* $\Phi^*\Phi = I_r$ *and* $\Psi^*\Phi = I_r$*. Of course such representatives may always be found by first choosing any representatives* $\tilde{\Phi}, \tilde{\Psi}$ *of the subspaces* V, W *and then letting* $\Phi = \mathrm{qf}(\tilde{\Phi})$*, where* qf *denotes orthogonalization by QR factorization, and letting* $\Psi = \tilde{\Psi}(\Phi^*\tilde{\Psi})^{-1}$*.*

8.3.2 Topology of the Optimization Problem Domain

The main result of this section is the following:

Theorem 8.3.3 (Topology of Subspaces that Define Oblique Projections)**.** *Let* $\mathcal{P}$ *denote the pairs of subspaces* $(V, W) \in \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ *that define oblique projection operators according to Proposition 8.2.2. Then* $\mathcal{P}$ *is open, dense, and connected in* $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$*. Moreover,* $\mathcal{P}$ *is diffeomorphic to the set of rank-r projection operators*

$$\mathbb{P} = \left\{ P \in \mathbb{R}^{n \times n} \ : \ P^2 = P \quad \textit{and} \quad \mathrm{rank}(P) = r \right\}. \tag{8.11}$$

Proof. See Appendix 8.A. □

The openness of $\mathcal{P}$ in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ means that it is a submanifold of $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ with the same dimension $\dim \mathcal{P} = 2nr - 2r^2$. The connectedness result is especially important since it means than an optimization routine can access any point in the set $\mathcal{P}$ by a smooth path from any initial guess without ever encountering the "bad set" $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r} \setminus \mathcal{P}$. In other words the bad set doesn't cut off access to any region of $\mathcal{P}$ by an optimizer that progresses along a smooth path, e.g., a gradient flow.

The reduced-order model Eq. 8.4 may not have a solution over the desired time interval $[t_0, t_{L-1}]$ for every projection operator defined by $(V, W) \in \mathcal{P}$. The following result characterizes the appropriate domain $\mathcal{D} \subset \mathcal{P}$ over which the ROM has a unique solution as well as the key properties of solutions when they exist.

Proposition 8.3.4 (Properties of ROM Solutions)**.** *When the reduced-order model Eq. 8.4 has a solution over the time interval* $[t_0, t_{L-1}]$*, it is unique. Let* $\mathcal{D} \subset \mathcal{P}$ *denote the set of subspace pairs* (V, W) *for which the resulting reduced-order model Eq. 8.4 has a unique solution over the time interval* $[t_0, t_{L-1}]$*. Then*

1. $\mathcal{D}$ *is open in* $\mathcal{P}$*, and hence* $\mathcal{D}$ *is also open in* $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$*.*

2. *When* $\frac{\partial}{\partial x} f(x, u(t))$ *is bounded then* $\mathcal{D} = \mathcal{P}$*.*

3. *If $\hat{x}(t;(V,W))$ denotes the solution of Eq. 8.4 with $(V,W) \in \mathcal{D}$, then $(V,W) \mapsto \hat{x}(t;(V,W))$ is continuously differentiable on $\mathcal{D}$ for every $t \in [t_0, t_{L-1}]$.*

4. *If $\{(V_k,W_k)\}_{k=1}^{\infty} \subset \mathcal{D}$ is a sequence approaching $(V_k,W_k) \to (V_0,W_0) \in \mathcal{P}\backslash\mathcal{D}$ and $\hat{x}(t;(V_k,W_k))$ are the corresponding solutions of Eq. 8.4, then*

$$\max_{t\in[t_0,t_{L-1}]} \|\hat{x}(t;(V_k,W_k))\| \to \infty \quad as \quad k \to \infty. \tag{8.12}$$

Proof. The claims follow from standard results in the theory of ordinary differential equations that can be found in W. G. Kelly A. C. Peterson [134]. We give the detailed proof in Appendix 8.B. □

In particular, Proposition 8.3.4 shows that the solutions produced by the reduced-order model are continuously differentiable over $\mathcal{D}$ and blow up as points outside of $\mathcal{D}$ are approached. In the special case when the governing equations Eq. 8.1 have a bounded Jacobian, we may dispense with $\mathcal{D}$ entirely since we find that the reduced-order model always has a unique, differentiable solution.

8.3.3 Regularization and Existence of a Minimizer

Without regularization, we cannot guarantee a priori that a sequence of subspace pairs with decreasing cost doesn't approach a point outside of the set $\mathcal{P}$ where projection operators are defined. That is, a minimizer for the cost function Eq. 8.5 may not even exist in $\mathcal{P}$, in which case our optimization problem would have no solution. In order to address this issue, we introduce a regularization function $\rho(V,W)$ into the cost Eq. 8.5 that "blows up" to $+\infty$ as the subspaces (V,W) approach any point outside of $\mathcal{P}$, and nowhere else. In order to do this, we use the fact that $(V,W) \in \mathcal{P}$ if and only if all representatives $(\Phi,\Psi) \in \pi^{-1}(V,W)$ have $\det(\Psi^*\Phi) \neq 0$, as shown in Proposition 8.2.2. While this condition characterizes the set $\mathcal{P}$, we cannot use $\det(\Psi^*\Phi)$ directly since its nonzero value depends on the choice of representatives. But this problem is easily solved by an appropriate normalization, leading us to define the regularization of Eq. 8.5 in terms of representatives according to

$$\boxed{\rho \circ \pi(\Phi,\Psi) = -\log\left(\frac{\det(\Psi^*\Phi)^2}{\det(\Phi^*\Phi)\det(\Psi^*\Psi)}\right).} \tag{8.13}$$

We observe that the function $\rho : \mathcal{P} \to \mathbb{R}$ in Eq. 8.13 is well-defined because $\rho \circ \pi(\Phi,\Psi)$ does not depend on the representatives (Φ,Ψ) thanks to the product rule for determinants.

The following theorem shows that the regularization defined by Eq. 8.13 has the desirable properties that it vanishes when $V = W$ and "blows up" as (V,W) escapes the set $\mathcal{P}$. When $V = W$,

the resulting projection operator $P_{V,V}$ is the orthogonal projection onto V.

Theorem 8.3.5 (Regularization)**.** *The minimum value of ρ defined by Eq. 8.13 over $\mathcal{P}$ is zero, and this minimum value $\rho(V,W)=0$ is attained if and only if $V=W$. On the other hand, if $(V_0,W_0)\in \mathcal{G}_{n,r}\times\mathcal{G}_{n,r}\setminus\mathcal{P}$ and $\{(V_n,W_n)\}_{n=1}^{\infty}$ is a sequence of subspaces in $\mathcal{P}$ such that $(V_n,W_n)\to(V_0,W_0)$ as $n\to\infty$, then $\lim_{n\to\infty}\rho(V_n,W_n)=\infty$.*

Proof. See Appendix 8.C. □

We must also rule out the possibility that a sequence of subspace pairs with decreasing cost approaches a point where the reduced-order model does not have a unique solution. By Proposition 8.3.4, we do not have this problem when the full-order model has a bounded Jacobian since the reduced-order model always has a unique solution, i.e., $\mathcal{D}=\mathcal{P}$. On the other hand, when $\mathcal{D}\neq\mathcal{P}$ we may accomplish this by choosing a cost function that blows up if the states of the reduced-order model blow up. In particular, we assume the following:

Assumption 8.3.6. *Let $\mathcal{D}$ be as in Proposition 8.3.4 and $\mathcal{P}$ be the set defined by Proposition 8.2.2. If $\mathcal{D}\neq\mathcal{P}$ and $\{(V_k,W_k)\}_{k=1}^{\infty}\subset\mathcal{D}$ is any sequence producing solutions $\hat{x}(t;(V_k,W_k))$ of the reduced-order model Eq. 8.4 such that*

$$\max_{t\in[t_0,t_{L-1}]}\|\hat{x}(t;(V_k,W_k))\|\to\infty \quad as \quad k\to\infty, \tag{8.14}$$

then we assume that $J(V_k,W_k)\to\infty$. Furthermore, we make the convention that $J(V,W)=\infty$ whenever $(V,W)\in\mathcal{P}\setminus\mathcal{D}$.

In practice, this is a reasonable assumption if $g(x)\to\infty$ as $\|x\|\to\infty$ and $L_y(y)\to\infty$ as $\|y\|\to\infty$. Alternatively, one could add a new regularization term to the cost function Eq. 8.5 that penalizes reduced-order model states with large magnitudes. In Corollary 8.C.1 we show that a minimizer of the cost function Eq. 8.5 exists in the valid set $\mathcal{D}\subset\mathcal{P}$ when Assumption 8.3.6 holds and we use the regularization described by Eq. 8.13 with any positive weight $\gamma>0$.

8.4 Computing the Gradient

With the reduced-order model representative Eq. 8.9 in hand, the following results (Theorem 8.4.1 and Corollary 8.4.2) allow us to compute the derivative of the cost Eq. 8.10 with respect to the matrices (Φ,Ψ) in the structure space $\bar{\mathcal{M}}=\mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{n\times r}$. In particular, we treat the matrices (Φ,Ψ) as general model parameters C whose values we wish to optimize. When the initial condition

x_0 for the full-order model is known, the initial condition for the reduced-order model representative Eq. 8.9 is given by $z_0 = (\Psi^*\Phi)^{-1}\Psi^* x_0$ and Corollary 8.4.2 provides a correction for the gradient provided by Theorem 8.4.1 due to the dependence of z_0 on (Φ, Ψ).

Theorem 8.4.1 (Gradient with Respect to Model Parameters)**.** *Suppose we have observation data* $\{y_1, \ldots, y_L\}$ *generated by a dynamical system at sample times* $t_0 < \cdots < t_{L-1}$ *and a parametric model for the system given by*

$$\frac{d}{d\mathrm{t}}\, z = \tilde{f}(z, u;\ C) \tag{8.15}$$

where $u(t)$ *is a known input signal,* C *are unknown parameters in a Riemannian manifold* $\bar{\mathcal{M}}$*, and the initial condition* $z(t_0) = z_0$ *is unknown. Furthermore, suppose that we aim to fit the unknown parameters by minimizing a cost function of the form*

$$\bar{J}_0(C, z_0) := \sum_{i=0}^{L-1} L_y(\tilde{g}(z(t_i);\ C) - y_i). \tag{8.16}$$

Let $F(t) = \frac{\partial}{\partial z}\tilde{f}(z(t), u(t);\ C)$*,* $S(t) = \frac{\partial}{\partial C}\tilde{f}(z(t), u(t);\ C)$*,* $H(t) = \frac{\partial}{\partial z}\tilde{g}(z(t);\ C)$*, and* $T(t) = \frac{\partial}{\partial C}\tilde{g}(z(t);\ C)$ *denote the linearized dynamics and observation functions around a solution* $z(t)$ *of Eq. 8.15 and define an adjoint variable* $\lambda(t)$ *that satisfies*

$$-\frac{d}{d\mathrm{t}}\,\lambda(t) = F(t)^*\lambda(t), \quad t \in (t_i, t_{i+1}], \quad 0 \le i < L-1, \tag{8.17a}$$

$$\lambda(t_i) = \lim_{t \to t_i^+} \lambda(t) + H(t_i)^*\,\nabla\, L_y(\tilde{g}(z(t_i);\ C) - y_i), \tag{8.17b}$$

$$\lambda(t_{L-1}) = H(t_{L-1})^*\,\nabla\, L_y(\tilde{g}\,(z(t_{L-1});\ C) - y_{L-1}). \tag{8.17c}$$

Here $(\cdot)^*$ *denotes the adjoint of a linear operator. Then the gradients of the cost function Eq. 8.16 with respect to the unknown variables* C *and* z_0 *subject to the dynamics Eq. 8.15 are given by*

$$\nabla_{z_0}\, \bar{J}_0(C, z_0) = \lambda(t_0) \tag{8.18}$$

$$\boxed{\nabla_C\, \bar{J}_0(C, z_0) = \int_{t_0}^{t_{L-1}} S(t)^*\lambda(t)\ dt + \sum_{i=0}^{L-1} T(t_i)^*\,\nabla\, L_y(\tilde{g}(z(t_i);\ C) - y_i).} \tag{8.19}$$

Proof. See Appendix 8.D. □

Corollary 8.4.2 (Gradient with Parameter-Dependent Initial Condition)**.** *When the initial condition* $z_0 = z_0(C)$ *in Theorem 8.4.1 is also a function of the parameters, then the gradient of*

$\bar{J}(C) = \bar{J}_0(C, z_0(C))$ *with respect to the parameters* C *is given by*

$$\nabla_C \bar{J}(C) = \nabla_C \bar{J}_0(C, z_0) + \left(\frac{\partial}{\partial C} z_0(C)\right)^* \nabla_{z_0} \bar{J}_0(C, z_0). \tag{8.20}$$

Proof. This is little more than the chain rule. For details, see Appendix 8.D. □

In order to optimize on $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$, we must make sense of the gradient computed using Theorem 8.4.1 and Corollary 8.4.2 with respect to (Φ, Ψ) in the structure space $\bar{\mathcal{M}} = \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}$ in terms of gradients that are tangent to the quotient manifold $\mathcal{M}$. The key idea is to endow $\bar{\mathcal{M}}$ with a special structure called a "horizontal distribution" that allows one to uniquely identify tangent vectors to $\bar{\mathcal{M}}$ that represent tangent vectors to $\mathcal{M}$. Together with the horizontal distribution, we also must define a Riemannian metric on the structure space that is unaffected by the choice of representative within an equivalence class [1]. In what follows, we summarize some useful results from Absil et al. [1], Chapter 3, which should be consulted for more details.

Since we do not have direct access to $T_{(V,W)}\mathcal{M}$ using a computer, it is necessary to work with representatives of the gradient that are tangent to the structure space $\bar{\mathcal{M}}$. But for any $\xi \in T_p\mathcal{M}$ and representative $\bar{p} \in \bar{\mathcal{M}}$ such that $p = \pi(\bar{p})$, there are an infinite number of possible $\bar{\xi} \in T_{\bar{p}}\bar{\mathcal{M}}$ that could serve as representatives of ξ in the sense that $\xi = \mathrm{D}\,\pi(\bar{p})\bar{\xi}$. A unique representative of ξ is identified by observing that the pre-image $\pi^{-1}(p)$ of any $p \in \mathcal{M}$ is a smooth submanifold of $\bar{\mathcal{M}}$ yielding a decomposition of the tangent space $T_{\bar{p}}\bar{\mathcal{M}}$ into a direct sum of the "vertical space" defined by $\mathcal{V}_{\bar{p}} = T_{\bar{p}}\pi^{-1}(p)$ and the "horizontal space" defined as its orthogonal complement $\mathcal{H}_{\bar{p}} = \mathcal{V}_{\bar{p}}^{\perp}$. Following Example 3.6.4 in [1], it can be shown that the orthogonal projection onto the horizontal space is given by

$$P^h_{(\Phi,\Psi)}(X, Y) = \left(X - \Phi(\Phi^*\Phi)^{-1}\Phi^* X,\ Y - \Psi(\Psi^*\Psi)^{-1}\Psi^* Y\right). \tag{8.21}$$

Using this horizontal distribution on the structure space, we have the following:

Definition 8.4.3 (horizontal lift [1])**.** *Given* $\xi \in T_p\mathcal{M}$ *and a representative* $\bar{p} \in \pi^{-1}(p)$*, there is always a unique element* $\bar{\xi}_{\bar{p}} \in \mathcal{H}_{\bar{p}}$ *called the "horizontal lift" of* ξ *such that* $\xi = \mathrm{D}\,\pi(\bar{p})\bar{\xi}_{\bar{p}}$*.*

An important consequence for optimization is that the horizontal lift of the gradient of a function $J : \mathcal{M} \to \mathbb{R}$ is given by the gradient of $\bar{J} = J \circ \pi$ [1], that is,

$$\overline{\nabla\, J(\pi(\bar{p}))}_{\bar{p}} = \nabla\, \bar{J}(\bar{p}), \qquad \forall \bar{p} \in \bar{\mathcal{M}}. \tag{8.22}$$

In particular, this means that the gradient computed using Theorem 8.4.1 and Corollary 8.4.2 is the "correct" representative in the sense that it is the horizontal lift at (Φ, Ψ) of the (inaccessible) gradient of the cost function tangent to $\mathcal{M}$ at (V, W). Moreover, given $(V, W) \in \mathcal{M}$ and invertible matrices $S, T \in GL_r$, then the unique horizontal lifts of a tangent vector $(\xi, \zeta) \in T_{(V,W)}\mathcal{M}$ at the representatives (Φ, Ψ) and $(\Phi S, \Psi T)$ are related by

$$(\bar{\xi}_{\Phi S},\ \bar{\zeta}_{\Psi T}) = (\bar{\xi}_{\Phi} S,\ \bar{\zeta}_{\Psi} T) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r}. \tag{8.23}$$

This can also be shown via a trivial adaptation of Example 3.6.4 in [1].

In order for the Riemannian metric on the structure space $\bar{\mathcal{M}}$ to induce a compatible Riemannian metric on the quotient manifold $\mathcal{M}$, we must have

$$\langle \xi,\ \zeta \rangle_p := \langle \bar{\xi}_{\bar{p}},\ \bar{\zeta}_{\bar{p}} \rangle_{\bar{p}} = \langle \bar{\xi}_{\bar{q}},\ \bar{\zeta}_{\bar{q}} \rangle_{\bar{q}}, \qquad \forall \xi, \zeta \in T_p\mathcal{M}, \quad \forall \bar{p}, \bar{q} \in \pi^{-1}(p), \tag{8.24}$$

so that the metric on $\bar{\mathcal{M}}$ is independent of the representative $\bar{p}$. The Riemannian metric we adopt for the structure space $\bar{\mathcal{M}}$ is given by

$$\langle (X_1, Y_1),\ (X_2, Y_2) \rangle_{(\Phi,\Psi)} = \operatorname{Tr}\left[(\Phi^*\Phi)^{-1} X_1^* X_2\right] + \operatorname{Tr}\left[(\Psi^*\Psi)^{-1} Y_1^* Y_2\right], \tag{8.25}$$

which clearly satisfies Eq. 8.24 thanks to Eq. 8.23.

Using the Riemannian metric Eq. 8.25 on the structure space, we next obtain an explicit form of each term required to compute the horizontal lift of the gradient using Theorem 8.4.1 and Corollary 8.4.2. Proposition 8.4.4 below also provides us with the gradient of the regularization function Eq. 8.13. In order to simplify these expressions, we have assumed that the particular representatives are chosen to satisfy $\Psi^*\Phi = I_r$, which is always possible thanks to Remark 8.3.2. The horizontal lift of the gradient computed at any equivalent point $(\Phi S, \Psi T)$ with $S, T \in GL_r$ can be readily obtained from the horizontal lift of the gradient computed at (Φ, Ψ) via Eq. 8.23.

Proposition 8.4.4 (Required Terms for Gradient)**.** *We assume that the representatives Φ and Ψ with $V = \operatorname{Range}\Phi$ and $W = \operatorname{Range}\Psi$ have been chosen such that $\Psi^*\Phi = I_r$. Then the terms required to compute the gradient of the cost function using the model Eq. 8.9 with respect to the representatives in the structure space via Theorem 8.4.1 and Corollary 8.4.2 are given by*

$$F(t)^* = \left(\frac{\partial}{\partial z} \tilde{f}(z(t), u(t);\ (\Phi, \Psi)) \right)^T \tag{8.26}$$

$$S(t)^* v = \Bigg(\left[\left(\frac{\partial}{\partial x} f(\Phi z(t), u(t)) \right)^* \Psi v z(t)^T - \Psi v \tilde{f}(z(t), u(t);\ (\Phi, \Psi))^T \right] \Phi^* \Phi, \\ \left[f(\Phi z(t), u(t)) - \Phi \tilde{f}(z(t), u(t);\ (\Phi, \Psi)) \right] v^T \Psi^* \Psi \Bigg) \in T_{(\Phi,\Psi)} \bar{\mathcal{M}} \qquad \forall v \in \mathbb{R}^r, \tag{8.27}$$

$$H(t)^* = \left(\frac{\partial}{\partial z} \tilde{g}(z(t);\ (\Phi, \Psi))) \right)^T, \tag{8.28}$$

$$T(t)^* w = \left(\left(\frac{\partial}{\partial x} g(\Phi z(t)) \right)^* w z(t)^T \Phi^* \Phi,\ 0 \right) \in T_{(\Phi,\Psi)} \bar{\mathcal{M}} \qquad \forall w \in \mathbb{R}^{\dim y}, \tag{8.29}$$

$$\left(\frac{\partial}{\partial (\Phi, \Psi)} z_0(\Phi, \Psi) \right)^* v \\ = \Big(- \Psi v (\Psi^* x_0)^T \Phi^* \Phi,\ (x_0 - \Phi \Psi^* x_0)\, v^T \Psi^* \Psi \Big) \in T_{(\Phi,\Psi)} \bar{\mathcal{M}} \qquad \forall v \in \mathbb{R}^r. \tag{8.30}$$

The gradient of the regularization function Eq. 8.13 in terms of representatives (Φ, Ψ) *satisfying* $\Psi^* \Phi = I_r$ *is given by*

$$\nabla(\rho \circ \pi)(\Phi, \Psi) = 2\Big(\Phi - \Psi(\Phi^* \Phi),\ \Psi - \Phi(\Psi^* \Psi) \Big) \in T_{(\Phi,\Psi)} \bar{\mathcal{M}}. \tag{8.31}$$

Proof. See Appendix 8.D. □

Below, we present Algorithm 2 to compute the gradient according to Theorem 8.4.1 and Corollary 8.4.2, with the appropriate terms given in Proposition 8.4.4.

Algorithm 2 Compute the cost function gradient with respect to (Φ, Ψ)

1: **input**: biorthogonal representatives $(\Phi, \Psi) \in \pi^{-1}(V, W)$, initial condition x_0, observations $\{y_l\}_{l=0}^{L-1}$ at times $\{t_l\}_{l=0}^{L-1}$, regularization weight γ.
2: Assemble and simulate the ROM representative Eq. 8.9 from initial condition $z_0 = \Psi^* x_0$, storing the trajectory $z(t)$ and predicted outputs $\{\hat{y}_l\}_{l=0}^{L-1}$.
3: Initialize the gradient: $\nabla \bar{J} \leftarrow T(t_{L-1})^* \nabla L_y(\hat{y}_{L-1} - y_{L-1})$.
4: Compute adjoint variable at final time: $\lambda(t_{L-1}) = H(t_{L-1})^* \nabla L_y(\hat{y}_{L-1} - y_{L-1})$.
5: **for** $l = L-2, L-3, \ldots, 0$ **do**
6: Solve the adjoint equation Eq. 8.17a backwards in time over the interval $[t_l, t_{l+1}]$ using the linearized ROM dynamics Eq. 8.26 and store $\lambda(t)$ on this interval.
7: Compute the integral component of Eq. 8.19 over the interval $[t_l, t_{l+1}]$: $\nabla \bar{J} \leftarrow \nabla \bar{J} + \int_{t_l}^{t_{l+1}} S(t)^* \lambda(t) \mathrm{d}t$ using Gauss-Legendre quadrature.
8: Add lth element of the sum in Eq. 8.19: $\nabla \bar{J} \leftarrow \nabla \bar{J} + T(t_l)^* \nabla L_y(\hat{y}_l - y_l)$.
9: Add "jump" Eq. 8.17b to the adjoint variable: $\lambda(t_l) \leftarrow \lambda(t_l) + H(t_l)^* \nabla L_y(\hat{y}_l - y_l)$.
10: **end for**
11: Add gradient due to initial condition: $\nabla \bar{J} \leftarrow \nabla \bar{J} + \left(\frac{\partial}{\partial (\Phi, \Psi)} z_0(\Phi, \Psi) \right)^* \lambda(t_0)$.
12: Normalize by trajectory length: $\nabla \bar{J} \leftarrow \nabla \bar{J} / L$.
13: Add regularization: $\nabla \bar{J} \leftarrow \nabla \bar{J} + \gamma \nabla(\rho \circ \pi)(\Phi, \Psi)$.
14: **return** $\nabla \bar{J}$

8.5 Optimization using a Conjugate Gradient Algorithm

In this section we provide the necessary tools to implement most gradient-based optimization algorithms [1] including stochastic gradient descent [28, 237], quasi-Newton methods [222, 117], and conjugate gradient methods [222, 235]. Here we use a conjugate gradient algorithm to illustrate the key ingredients, such as the retraction and transport of tangent vectors on the Grassmann manifold. We also provide convergence guarantees for the algorithm on broad classes of sufficiently smooth full-order models, including all linear systems as a special case.

Line search optimization techniques in Euclidean space entail choosing a search direction η_k and a step size α_k in order to produce the next iterate of the optimization process according to

$$p_{k+1} = p_k + \alpha_k \eta_k. \tag{8.32}$$

The step size α_k is usually chosen using a backtracking or bisection approach in order to meet a sufficient decrease condition like the one proposed by P. Wolfe [282]. In the steepest descent approach, one searches along the direction of the gradient $\eta_k = -\nabla J(p_k)$. Yet in poorly conditioned problems, the steepest descent method may converge slowly, and it can be greatly improved by choosing a search direction that incorporates second-order information about the cost function. However, in high-dimensional applications like finding optimal projection subspaces for large-scale dynamical systems, we cannot efficiently evaluate the second derivatives of the cost function. Conjugate gradient algorithms provide efficient alternatives by computing a search direction that combines the gradient at the current iterate with the previous search direction

$$\eta_k = -\nabla J(p_k) + \beta_k \eta_{k-1}. \tag{8.33}$$

There are many choices for the coefficient β_k; for instance, the one due to Y.-H. Dai and Y. Yuan [70] is given by

$$\beta_k^{(DY)} = \frac{\langle \nabla J(p_k),\ \nabla J(p_k) \rangle}{\langle \nabla J(p_k),\ \eta_{k-1} \rangle - \langle \nabla J(p_{k-1}),\ \eta_{k-1} \rangle}. \tag{8.34}$$

This coefficient guarantees convergence of the Euclidean conjugate gradient algorithm in the sense that the limiting infimum of the gradients at the iterates is zero when the line search satisfies the Wolfe conditions and the gradient of J is Lipschitz continuous on sub-level sets [70].

8.5.1 Retraction of Tangent Vectors onto the Manifold

In the setting of optimization on Riemanniann manifolds two problems with the Euclidean formulation of conjugate gradient algorithms must be addressed. First, we must have a suitable generalization of what it means to search along a "line" on the manifold. A natural, but computationally expensive choice, is the exponential map on the Riemanniann manifold. The idea of a "retraction" was introduced by [4] as a computationally efficient alternative to the exponential map that retains only the properties that are needed for the purpose of optimization. In particular, [1] gives the following Definition 8.5.1 for a retraction.

Definition 8.5.1 (Retraction [1])**.** *A retraction on a manifold $\mathcal{M}$ is a smooth mapping R from the tangent bundle $T\mathcal{M}$ onto $\mathcal{M}$ with the following properties. Let R_p denote the restriction of R to $T_p\mathcal{M}$.*

1. *(Base point preservation) $R_p(0) = p$.*

2. *(Local rigidity) R_p satisfies*

$$\mathrm{D}\, R_p(0)\xi = \xi \qquad \forall \xi \in T_p\mathcal{M} \tag{8.35}$$

(with the canonical identification $T_0T_p\mathcal{M} \simeq T_p\mathcal{M}$).

The retraction R_p parameterizes a neighborhood of $p \in \mathcal{M}$ using elemets of the tangent space, allowing us to pull our optimization problem back to the Euclidean space $T_p\mathcal{M}$ while the local rigidity condition ensures that the search direction is preserved. Line search may be performed on $\mathcal{M}$ using the retraction and a search direction $\eta_k \in T_{p_k}\mathcal{M}$ by defining the next iterate according to

$$p_{k+1} = R_{p_k}(\alpha_k \eta_k). \tag{8.36}$$

In our case, $\mathcal{M}$ is a quotient manifold of a structure space $\bar{\mathcal{M}}$ whose elements we must use as representatives. Proposition 4.1.3 in [1] says that a retraction $\bar{R}$ on the structure space $\bar{\mathcal{M}}$ induces a retraction R on the quotient manifold $\mathcal{M}$ when $\pi \circ \bar{R}$ does not depend on the choice of representatives. In particular, Example 4.1.5 in [1] shows that $R_{[\Phi]}(\xi) = [\Phi + \bar{\xi}_\Phi]$ defines a retraction on the Grassmann manifold. Therefore, for our problem on a product of Grassmann manifolds $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ we have a retraction defined by

$$R_{(V,W)}(\xi, \zeta) = \pi\left(\Phi + \bar{\xi}_\Phi,\ \Psi + \bar{\zeta}_\Psi\right), \qquad \text{for any } (\Phi, \Psi) \in \pi^{-1}(V, W). \tag{8.37}$$

In particular, $\left(\Phi + \bar{\xi}_\Phi, \Psi + \bar{\zeta}_\Psi\right)$ is a representative of $R_{(V,W)}(\xi, \zeta)$. In order to avoid ill-conditioning, we work with biorthogonalized representatives obtained by applying the procedure described in Remark 8.3.2.

8.5.2 Transporting Tangent Vectors to New Points

The second problem that must be solved in order to implement conjugate gradient algorithms on Riemannian manifolds is how to make sense of the search direction defined by Eq. 8.33, which combines elements from two different tangent spaces. In particular, the previous gradient $\nabla J(p_{k-1})$ and the previous search direction η_{k-1} lie in the tangent space $T_{p_{k-1}}\mathcal{M}$, which is different from the tangent space $T_{p_k}\mathcal{M}$ in which the current search direction η_k and gradient $\nabla J(p_k)$ lie. To solve this problem, vectors in $T_{p_{k-1}}\mathcal{M}$ must be "transported" to the new tangent space $T_{p_k}\mathcal{M}$. The most natural notion of vector transport on a Riemannian manifold is parallel translation along geodesics; yet computing parallel translations can be expensive. The following Definition 8.5.2 of "vector transport" given by Absil et al. [1] retains only the properties that are essential in the context of optimization.

Definition 8.5.2 (Vector Transport [1])**.** *Let the "Whitney sum"*

$$T\mathcal{M} \oplus T\mathcal{M} = \{(\eta_p, \xi_p) \ : \ \eta_p, \xi_p \in T_p\mathcal{M}, \ p \in \mathcal{M}\} \tag{8.38}$$

denote pairs of tangent vectors sharing the same root points. A vector transport on the manifold $\mathcal{M}$ is a smooth mapping

$$T\mathcal{M} \oplus T\mathcal{M} \to T\mathcal{M} \ : \ (\eta_p, \xi_p) \mapsto \mathcal{T}_{\eta_p}(\xi_p) \tag{8.39}$$

satisfying the following properties:

1. *(Associated retraction) There exists a retraction R, called the retraction associated with $\mathcal{T}$, such that $\mathcal{T}_{\eta_p}(\xi_p) \in T_{R_p(\eta_p)}\mathcal{M}$ for every $(\eta_p, \xi_p) \in T\mathcal{M} \oplus T\mathcal{M}$.*
2. *(Consistency) $\mathcal{T}_{0_p}(\xi_p) = \xi_p$ for all $\xi_p \in T\mathcal{M}$*
3. *(Linearity) $\mathcal{T}_{\eta_p}(a\xi_p + b\zeta_p) = a\mathcal{T}_{\eta_p}(\xi_p) + b\mathcal{T}_{\eta_p}(\zeta_p)$ for all $a, b \in \mathbb{R}$, $\eta_p, \xi_p, \zeta_p \in T_p\mathcal{M}$, and every $p \in \mathcal{M}$.*

As pointed out in [1], if one has a retraction, then a vector transport can be obtained by differ-

entiating it and letting

$$\mathcal{T}_{\eta_p}(\xi_p) := \mathrm{D}\, R_p(\eta_p)\xi_p = \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, R_p(\eta_p + t\xi_p)\bigg|_{t=0}. \tag{8.40}$$

Following Example 8.1.10 in [1] and differentiating our retraction Eq. 8.37 on $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$, we obtain the following vector transport defined in terms of its horizontal lift

$$\overline{\mathcal{T}_{(\xi,\eta)_{(V,W)}}\left((\omega,\zeta)_{(V,W)}\right)}_{(\Phi+\bar{\xi}_\Phi,\Psi+\bar{\eta}_\Psi)} = P^h_{(\Phi+\bar{\xi}_\Phi,\Psi+\bar{\eta}_\Psi)}\left(\bar{\omega}_\Phi, \bar{\zeta}_\Psi\right), \tag{8.41}$$

for any $(\Phi, \Psi) \in \pi^{-1}(V, W)$. We recall that P^h is the orthogonal projection onto the horizontal space given by Eq. 8.21. The horizontal lifts of transported vectors at the biorthogonalized representatives computed using the procedure in Remark 8.3.2 are found by applying the transformation Eq. 8.23.

8.5.3 Geometric Conjugate Gradient Algorithm

The search directions for Riemannian conjugate gradient algorithms are computed by transporting the previous search direction according to

$$\eta_k = -\nabla J(p_k) + \beta_k \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1}). \tag{8.42}$$

We use the scaled Riemannian Dai-Yuan coefficient proposed by H. Sato [235] since it guarantees (under Lipschitz assumptions on $\mathrm{D}(J \circ R)$) that the resulting conjugate gradient algorithm with line search based on Wolfe-type conditions always converges, though not necessarily to a global minimum of J. In particular, [235] defines the scaling factor

$$\sigma_k = \min\left\{1, \frac{\|\eta_{k-1}\|_{p_{k-1}}}{\|\mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})\|_{p_k}}\right\}, \tag{8.43}$$

and lets the coefficient β_k be defined by a scaled generalization of Eq. 8.34 given by

$$\beta_k^{(sRDY)} = \frac{\sigma_k \left\langle \nabla J(p_k),\ \nabla J(p_k)\right\rangle_{p_k}}{\sigma_k \left\langle \nabla J(p_k),\ \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\eta_{k-1})\right\rangle_{p_k} - \left\langle \nabla J(p_{k-1}),\ \eta_{k-1}\right\rangle_{p_{k-1}}}. \tag{8.44}$$

The Wolfe conditions to be satisfied during the line search are

$$J\left(R_{p_k}(\alpha_k \eta_k)\right) \leq J(p_k) + c_1 \alpha_k \left\langle \nabla J(p_k),\ \eta_k \right\rangle_{p_k} \tag{8.45}$$

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} J\left(R_{p_k}(t\eta_k)\right)\Big|_{t=\alpha_k} \geq c_2 \left\langle \nabla J(p_k),\ \eta_k \right\rangle_{p_k}, \tag{8.46}$$

with $0 < c_1 < c_2 < 1$ and the bisection method described in [39] is used. We now have all of the machinery needed to implement a geometric conjugate gradient method, detailed below in Algorithm 3, to minimize the cost function Eq. 8.5 over pairs of subspaces $\mathcal{P} \subset \mathcal{M}$ defining projection-based reduced-order models. In Appendix 8.E we provide convergence guarantees for this algorithm

Algorithm 3 Geometric conjugate gradient algorithm for model reduction

Input: biorthogonal representatives (Φ_0, Ψ_0) of the initial subspaces, stopping threshold $\varepsilon > 0$, and Wolfe condition coefficients $0 < c_1 < c_2 < 1$
Compute cost $\bar{J}(\Phi_0, \Psi_0)$ and gradient $\nabla \bar{J}_0$ using Algorithm 2
Initialize the search direction $(X_0, Y_0) = \nabla \bar{J}_0$ and set $k = 0$
while $\left\langle \nabla \bar{J}_k, \nabla \bar{J}_k \right\rangle_{(\Phi_k, \Psi_k)} > \varepsilon$, given by Eq. 8.25, **do**
    Define line-search objective $J_k(\alpha) = \bar{J}(\Phi_k + \alpha X_k, \Psi_k + \alpha Y_k)$ via retraction
    Compute step size α_k using the bisection method in [39], so that $J_k(\alpha_k)$ satisfies the Wolfe conditions, namely $J_k(\alpha_k) \leq J_k(0) + c_1 \alpha_k J_k'(0)$ and $J_k'(\alpha_k) \geq c_2 J_k'(0)$
    Compute next iterate $(\Phi_{k+1}, \Psi_{k+1}) = (\Phi_k + \alpha_k X_k, \Psi_k + \alpha_k Y_k)$ via retraction
    Transport search direction $(\tilde{X}_k, \tilde{Y}_k) = P^h_{(\Phi_{k+1}, \Psi_{k+1})}(X_k, Y_k)$ using Eq. 8.21
    Compute slim QR factorization $\Phi_{k+1} = QR$ and biorthogonalizing transformation matrices $S = R^{-1}$ and $T = (Q^* \Psi_{k+1})^{-1}$
    Biorthogonalize representatives $(\Phi_{k+1}, \Psi_{k+1}) \leftarrow (\Phi_{k+1} S, \Psi_{k+1} T)$ and transform the search direction $(\tilde{X}_k, \tilde{Y}_k) \leftarrow (\tilde{X}_k S, \tilde{Y}_k T)$ via Eq. 8.23
    Compute cost $\bar{J}(\Phi_{k+1}, \Psi_{k+1})$ and gradient $\nabla \bar{J}_{k+1}$ using Algorithm 2
    Using Eq. 8.25, compute Riemannian Dai-Yuan scaling factor and coefficient

$$\sigma_{k+1} = \min\left\{1, \sqrt{\frac{\langle (X_k, Y_k), (X_k, Y_k) \rangle_{(\Phi_k, \Psi_k)}}{\langle (\tilde{X}_k, \tilde{Y}_k), (\tilde{X}_k, \tilde{Y}_k) \rangle_{(\Phi_{k+1}, \Psi_{k+1})}}}\right\},$$

$$\beta_{k+1} = \frac{\sigma_{k+1} \langle \nabla \bar{J}_{k+1}, \nabla \bar{J}_{k+1} \rangle_{(\Phi_{k+1}, \Psi_{k+1})}}{\sigma_{k+1} \langle \nabla \bar{J}_{k+1}, (\tilde{X}_k, \tilde{Y}_k) \rangle_{(\Phi_{k+1}, \Psi_{k+1})} + \langle \nabla \bar{J}_k, (X_k, Y_k) \rangle_{(\Phi_k, \Psi_k)}}$$

    Compute next search direction $(X_{k+1}, Y_{k+1}) = \nabla \bar{J}_{k+1} + \beta_{k+1}(\tilde{X}_k, \tilde{Y}_k)$.
    Update $k \leftarrow k + 1$
end while
return biorthogonal representatives (Φ_K, Ψ_K) of the optimized projection subspaces and the final cost $\bar{J}(\Phi_K, \Psi_K)$

applied to our optimal model reduction problem under modest conditions on the problem's setup (see Theorem 8.E.1, Corollary 8.E.7, and Corollary 8.E.8). In particular, these guarantees say that for any threshold $\varepsilon > 0$ on the gradient, Algorithm 3 will eventually stop.

8.6 Simple Nonlinear System with an Important Low-Energy Feature

In this section, we illustrate our method on a simple example system for which existing approaches to nonlinear model reduction perform poorly. In particular, we consider the system

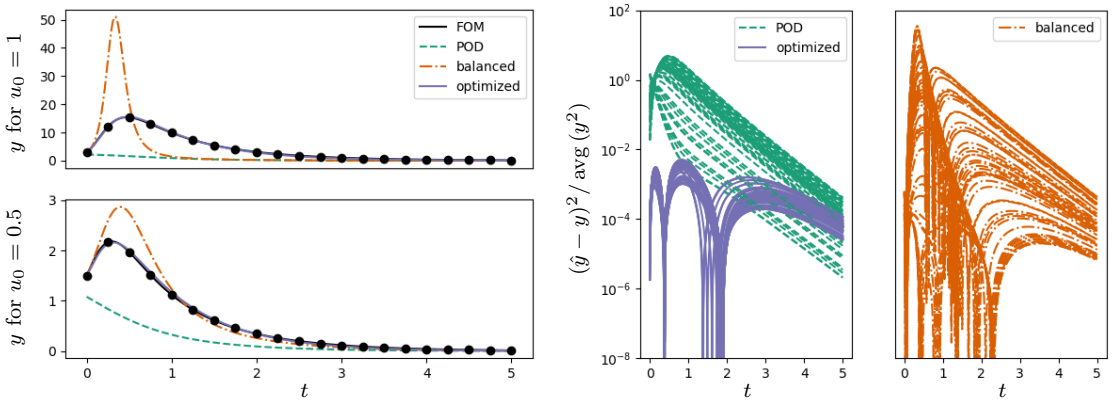
$$\begin{aligned} \dot{x}_1 &= -x_1 + 15x_1x_3 + u \\ \dot{x}_2 &= -2x_2 + 15x_2x_3 + u \\ \dot{x}_3 &= -5x_3 + u \\ y &= x_1 + x_2 + x_3, \end{aligned} \tag{8.47}$$

and we compare our method with POD/Galerkin projection onto the most energetic modes, and with Petrov-Galerkin projection onto subspaces determined by balanced truncation of the linearized system. We confine our attention to nonlinear impulse-responses with magnitudes $u_0 \in [0, 1]$. These responses can be obtained by considering the output of Eq. 8.47 with $u \equiv 0$ and known initial condition $x(0) = u_0(1,\ 1,\ 1)$. Two such responses with $u_0 = 0.5$ and $u_0 = 1$ are shown in Figure 8.6.1a.

The key feature of Eq. 8.47 is that that state x_3 plays a very important role in the dynamics of the states x_1 and x_2, while remaining small by comparison due to its fast decay rate. In fact, for $u_0 > 8/30$ we have $\dot{y}(0) > 0$ and the output experiences transient growth due to the nonlinear interaction of x_1 and x_2 with x_3. These nonlinear interactions become dominant for larger u_0, but are neglected completely by model reduction techniques like balanced truncation that consider only the linear part of Eq. 8.47. Figure 8.6.1a shows the result of such an approach, in which we obtain a nonlinear reduced-order model by Petrov-Galerkin projection of Eq. 8.47 onto a two-dimensional subspace determined by balanced truncation of the linearized system. As shown in the figure, the resulting model wildly over-predicts the transient growth when $u_0 = 1$.

On the other hand, a two-dimensional POD-based model retains the most energetic states, which align closely with x_1 and x_2, and essentially ignores the important low-energy state x_3. Consequently, the POD-based model of Eq. 8.47 does not predict any transient growth as shown in Figure 8.6.1a.

In order to find a two-dimensional reduced-order model of Eq. 8.47 using our new approach, we collected the two impulse-response trajectories shown in Figure 8.6.1a and used the $L = 21$ equally



(a) training trajectories

(b) error on testing trajectories

Figure 8.6.1: In panel (a), we show the outputs generated by the full-order model Eq. 8.47 and the two-dimensional reduced-order models found by POD Galerkin projection, balanced truncation, and our optimization approach in response to impulses with magnitudes $u_0 = 0.5$ and $u_0 = 1$ at $t = 0$. The sample points used to construct the objective function Eq. 8.48 used to optimize the projection operator are shown as black dots. In panel (b), we show the normalized square errors of the reduced-order model predictions in response to 50 impulses at $t = 0$ whose magnitudes u_0 were drawn uniformly at random from the interval $[0, 1]$.

spaced samples shown for each trajectory to define the cost function

$$J(V, W) = \sum_{u_0 \in \{0.5, 1.0\}} \frac{1}{\sum_{l=0}^{L} (y|_{u_0}(t_l))^2} \sum_{l=0}^{L} (\hat{y}|_{u_0}(t_l) - y|_{u_0}(t_l))^2 + \gamma \rho(V, W), \tag{8.48}$$

with $\gamma = 10^{-3}$ (although we note that the results were not sensitive to the choice of γ). The normalizing factor in the cost for each trajectory was used to penalize the error relative to the average energy content of the trajectory, rather than in an absolute sense which would be dominated by the trajectory with $u_0 = 1$. Starting from an initial model formed by balanced truncation, the conjugate gradient algorithm described above with Wolfe conditions defined by $c_1 = 0.4$ and $c_2 = 0.8$ achieved convergence with a gradient magnitude smaller than 10^{-4} after 88 steps.

In Figure 8.6.1a, we see that the resulting reduced-order model trajectories very closely match the trajectories used to find the oblique projection. Moreover, we tested the predictions of the three reduced-order models on 50 impulse-response trajectories with u_0 drawn uniformly at random from the interval $[0, 1]$. The square output prediction errors for each trajectory normalized by the average output energy of the full-order model are shown in Figure 8.6.1b. We observe that the POD-based model is poor regardless of the impulse magnitude u_0, whereas the balanced reduced-order model performs well when u_0 is very close to 0, but poorly when u_0 is closer to 1. On the other hand, our optimized reduced-order model yields very accurate predictions for all impulse response

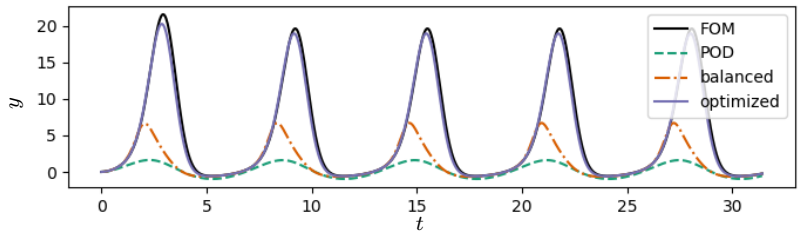


Figure 8.6.2: We show the responses of Eq. 8.47 and the reduced-order models to input $u(t) = \sin(t)$.

magnitudes in the desired range. Furthermore, the reduced-order model we trained to minimize error using two impulse responses has excellent predictive performance with different inputs. For instance, Figure 8.6.2 shows the predictions of the reduced-order models in response to sinusoidal input $u(t) = \sin(t)$ with zero initial condition.

8.7 Reduction of a High-Dimensional Nonlinear Fluid Flow

In this section we set out to develop a reduced-order model capable of predicting the response of an incompressible jet flow to impulsive disturbances in the proximity of the nozzle. We consider the evolution of an axisymmetric jet flow over the spatial domain $\Omega = \{(r, z) \mid r \in [0, L_r], z \in [0, L_z]\}$. Velocities are nondimensionalized by the centerline velocity U_0, lengths by the jet diameter D_0, and pressure by ρU_0^2, where ρ is the fluid density.

Letting $q = (u, v)$ denote the (dimensionless) velocity vector with axial component u and radial component v, and letting p be the (dimensionless) pressure field, we may write the governing equations in cylindrical coordinates as

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial z} - v\frac{\partial u}{\partial r} - \frac{\partial p}{\partial z} + \frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{\partial^2 u}{\partial z^2}\right) \tag{8.49}$$

$$\frac{\partial v}{\partial t} = -u\frac{\partial v}{\partial z} - v\frac{\partial v}{\partial r} - \frac{\partial p}{\partial r} + \frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) - \frac{v}{r^2} + \frac{\partial^2 v}{\partial z^2}\right) \tag{8.50}$$

$$\frac{\partial u}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}(rv) = 0, \tag{8.51}$$

where $Re = U_0 D_0/\nu$ is the Reynolds number (and ν denotes the kinematic viscosity of the fluid). Formulas (8.49) and (8.50) are conservation of momentum statements in the axial and radial directions, respectively, while formula (8.51) is a mass conservation statement. Conservation of mass may be used to eliminate pressure from formulas (8.49) and (8.50), as discussed in Appendix 8.G. We impose a zero-velocity boundary condition at $r = L_r$, a Neumann outflow boundary condition

at $z = L_z$, and we let the inflow velocity be

$$u(r,0) = \frac{1}{2}\left(1 - \tanh\left[\frac{1}{4\theta_0}\left(r - \frac{1}{r}\right)\right]\right), \tag{8.52}$$

where θ_0 is a dimensionless thickness, which we fix at $\theta_0 = 0.0125$.

The equations of motion are integrated in time using the fractional step method described in [206] in conjunction with the second-order Adams-Bashforth multistep scheme. The spatial discretization is performed on a fully-staggered grid of size $N_z \times N_r = 230 \times 150$ and with $L_z = 8$ and $L_r = 3$. If we let the state be composed of the axial and radial velocities at the cell faces, then the state dimension for this flow is $2(N_z \times N_r) = 69,000$. All the spatial derivatives are treated with second-order central differences, except for the advective term $q \cdot \nabla q$, which is treated with a third-order upwind scheme in order to avoid numerical instabilities. The solver has been validated against some of the results presented in [246], for which we observed very good qualitative agreement. While the inner product on the state space is given by

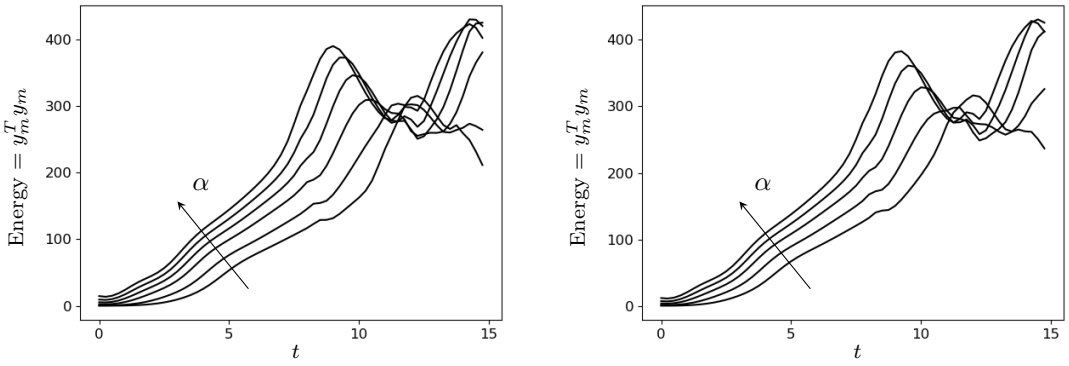
$$\langle f, g \rangle = \int_\Omega f(r,z) g(r,z)\, r \, \mathrm{d}r \, \mathrm{d}z, \tag{8.53}$$

our observations, y, will be the full velocity field, with the standard Euclidean inner product (i.e., without weighting by r). A detailed derivation of the adjoint of the Navier-Stokes operator required to compute the gradient is presented in Appendix 8.H.

8.7.1 Results

For the described flow configuration, there exists a convectively unstable steady-state solution, which we will denote Q. In particular, any perturbation q' about the steady-state solution will grow while advecting downstream and it will eventually leave the computational domain through the outflow located at $z = L_z$. During the growth process, nonlinear effects become dominant and lead to the formation of complicated vortical structures. In this section we seek to develop a reduced-order model of the initial growth of these disturbances in response to an impulse, and we consider impulses that enter the radial momentum equation (8.50) through a velocity perturbation localized near $r = 1/2$ and $z = 1$; in particular, the perturbation has the form $B(r,z)w(t)$, where

$$B(r,z) = \exp\left\{-\frac{(r-1/2)^2 + (z-1)^2}{\theta_0}\right\}. \tag{8.54}$$



(a) training data (b) testing data

Figure 8.7.1: In panel (a) we show the time history of the energy of the impulse responses in the training data set, with $\alpha = 0.1, 0.2, 0.4, 0.6, 0.8, 1.0$. In panel (b) we show the analog of panel (a) for the testing data set, with $\alpha = 0.15, 0.3, 0.5, 0.7, 0.9$. In each trajectory there are a total of 60 equally-spaced data points.

We simulate the response of the flow to a given impulse $w(t) = \alpha\delta(t)$, with $\alpha \in \mathbb{R}$, by integrating the governing equations (8.49)–(8.51) with initial condition

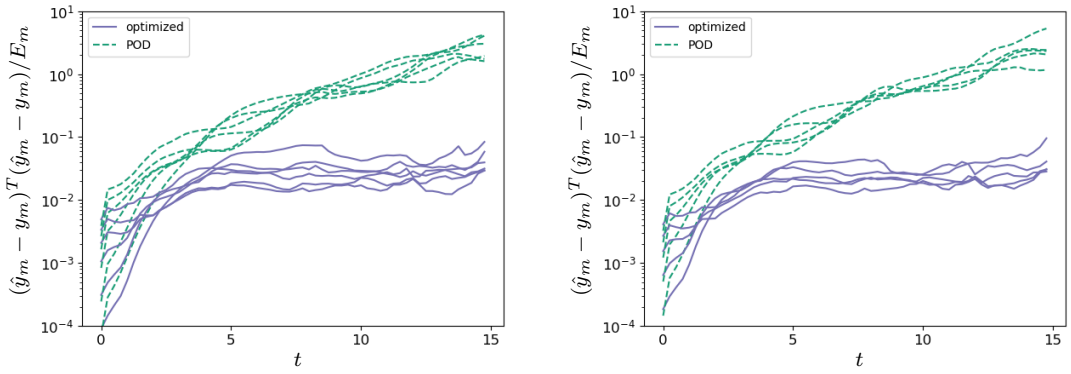
$$q(0) = Q + q'(0), \quad \text{where} \quad q'(0) = (0, B\alpha). \tag{8.55}$$

Here we construct a 50-dimensional reduced-order model to capture the initial response of the flow to impulses with $0.1 \leq \alpha \leq 1.0$ for times $t \in [0, 15]$.

We proceed as follows. We generate a training set of $M = 6$ trajectories corresponding to values $\alpha = 0.1, 0.2, 0.4, 0.6, 0.8, 1.0$, and from each trajectory we observe $L = 60$ equally-spaced snapshots of velocity perturbations $y = q'$ about the base flow Q. The energy content of the training set is shown in figure 8.7.1a. Observe the range of behavior for different values of α, reflecting the strong nonlinearity of this flow. Let $y_{m,l}$ denote the lth velocity snapshot in the mth trajectory and let $\hat{y}_{m,l}$ denote the corresponding prediction obtained by integrating the reduced-order model from the initial condition $\hat{q}'_{m,0} = P_{V,W} q'_{m,0}$. Letting E_m denote the average energy along the mth trajectory, we seek to minimize the cost function

$$J(V, W) = \frac{1}{ML} \sum_{m=0}^{M-1} \frac{1}{E_m} \sum_{l=0}^{L-1} \left(\hat{y}_{m,l} - y_{m,l}\right)^T \left(\hat{y}_{m,l} - y_{m,l}\right) + \gamma\rho(V, W), \tag{8.56}$$

where $\gamma = 10^{-3}$. Optimization was carried out using Algorithm 3 with a 50-dimensional model obtained by POD as the initial guess. The initial Ψ modes were smoothly truncated near the outflow to satisfy the adjoint boundary conditions described in Appendix 8.H.



(a) training error

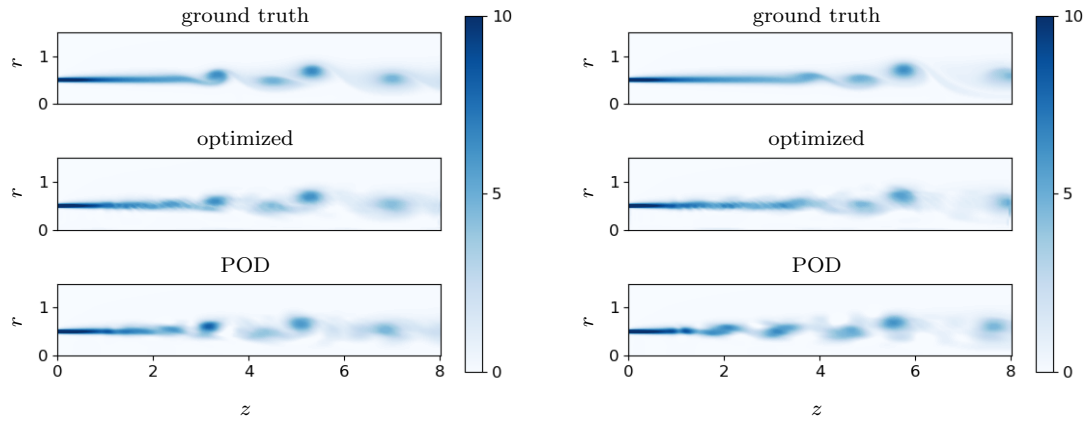
(b) testing error

Figure 8.7.2: In panel (a) we show the square error across all training trajectories for the optimal reduced-order model and for the POD-based reduced-order model. Panel (b) is the analog of panel (a), except that the error is computed against the testing trajectories.

Remark 8.7.1. *It was advantageous to begin training the model on shorter trajectories and work our way up to the full time horizon.*

After obtaining a minimizer, we test the performance of the optimal reduced-order model on a set of $M = 5$ unseen impulse responses with $\alpha = 0.15, 0.3, 0.5, 0.7, 0.9$. The energy content of the testing set is shown in figure 8.7.1b. The performance of our reduced-order model is shown in figure 8.7.2 and it is compared against a 50-dimensional POD/Galerkin reduced-order model. We do not show a comparison against a BPOD-based Petrov-Galerkin model because its predictions "blew up" after a few time units. Before proceeding in the analysis of the results, it is worth mentioning that the first 50 POD modes capture approximately 99.6% of the energy of the training data set, as well as approximately 99.6% of the energy of the testing data set. On the other hand, the subspace V we found by optimization captures 99.4% of the energy of both the training and testing sets. Figure 8.7.2a shows the error over time across all training trajectories for the optimal reduced-order model and for the POD-based reduced-order model. Figure 8.7.2b is the analog of figure 8.7.2a, except that the error is computed against the testing data. In both, we can observe that the average error of the optimal reduced-order model is one to two orders of magnitude lower than that of the POD-based model. Moreover, the error curves associated with the optimal model remain approximately constant for times $t \in [2, 15]$, which suggests that we capture the dynamically-relevant features of the full-order model. This is no surprise, since our framework is designed specifically to develop models that are dynamically accurate and that may therefore be used to predict the time-evolution of the full-order system over some time horizon.

Figures 8.7.3a and 8.7.3b show the predicted vorticity fields, $\nabla \times (\hat{y} + Q)$, at times $t = 10$ and



(a) vorticity at time $t = 10$ and $\alpha = 0.15$

(b) vorticity at time $t = 14.75$ and $\alpha = 0.15$

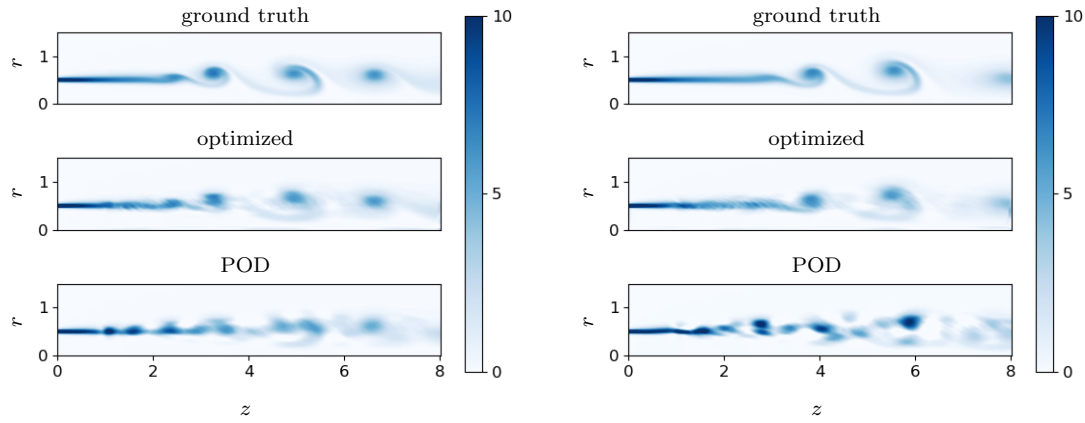
Figure 8.7.3: In panel (a) we show the predicted vorticity field from the testing trajectory with $\alpha = 0.15$ at time $t = 10.0$. From the top we have the ground truth from the full-order model, the prediction given by the optimized model and the prediction given by POD. Panel (b) is the analog of panel (a), at time $t = 14.75$. The colormap has been saturated to allow for better visualization of the downstream structures.

$t = 14.75$ from the unseen testing trajectory with initial impulse amplitude $\alpha = 0.15$. It can be seen from both panels in figure 8.7.3 that the prediction from the optimal model closely resembles the ground truth. While the POD-based prediction remains qualitatively close to the ground truth at time $t = 10$, the prediction at time $t = 14.75$ is inaccurate. In fact, the POD-based model mistakenly predicts the presence of two vortices at $z \approx 2$ and $z \approx 3$ and it also accumulates some phase error on the vortex located near the outflow.

Figures 8.7.4a and 8.7.4b show the predicted vorticity fields, at times ($t = 10$ and 14.75) from the unseen testing trajectory with initial impulse amplitude $\alpha = 0.9$. Here, nonlinear effects dominate the dynamics and this is the reason why the POD-based model struggles to even capture the qualitative behavior. The optimized model, on the other hand, captures all instances of vortex shedding and vortex pairing, and it is capable of accurately predicting the locations of the resulting vortical structures.

8.8 Conclusions

We have introduced a reduced-order modeling approach for large-scale nonlinear dynamical systems based on optimizing oblique projections of the governing equations to minimize prediction error over sampled trajectories. We implemented a provably convergent geometric conjugate gradient algorithm in order to optimize a regularized trajectory prediction error over the product of Grassmann manifolds defining the projection operators. The computational cost to evaluate the gradient



(a) vorticity at time $t = 10$ and $\alpha = 0.9$

(b) vorticity at time $t = 14.75$ and $\alpha = 0.9$

Figure 8.7.4: Analog of Figure 8.7.3 for a higher amplitude $\alpha = 0.9$.

is dominated by the cost to assemble the reduced-order model and to evaluate the time derivative of the full-order model at quadrature points along each trajectory. While time-stepping schemes for the full-order model require very fine temporal discretizations to resolve dynamics over a wide range of scales and to remain numerically stable, we may compute the gradient using a much coarser temporal sampling and high-order quadrature rules. Thus, computing the gradient may be orders of magnitude less costly than running a direct simulation when the cost to assemble the reduced-order model is small by comparison.

The method is compared with Proper Orhogonal Decomposition (POD)-based Galerkin projection as well as Petrov-Galerkin projection onto balancing and adjoint modes derived from linearized dynamics about equilibria. We considered a simple three-dimensional system with an important low-energy feature as well as a nonlinear axisymmetric jet flow with $69,000$ state variables. In both cases, the optimized Petrov-Galerkin reduced-order model vastly out-performs the predictions made using the projection-based models obtained by POD and balanced truncation on new trajectories. We argue that this is because POD, while optimal for reconstructing states, ignores important low-energy features that influence the dynamics in the future. On the other hand, model reduction approaches like balanced truncation that rely only on the linearized dynamics can fail to capture features that have important nonlinear interactions far away from equilibria. Our approach is capable of capturing the relevant features needed to predict the nonlinear dynamics on a representative collection of trajectories.

The primary limitation of our approach is that a sufficiently large collection of trajectories must be used to avoid over-fitting. Based on algebraic considerations, the total number of sample data should exceed the dimension $2nr - 2r^2$ of the product of Grassmann manifolds over which we

optimize, where n is the state dimension and r is the dimension of the reduced-order model. In order to avoid over-fitting we suggest exceeding this minimum data requirement by a factor of at least 3. For instance, in the case of our fluid flow example, r is small compared with n and we use 360 snapshots of the state as training data to optimize over subspaces with $2r = 100$. When a large number of trajectories are used, it may be advantageous to employ a stochastic gradient descent algorithm [28, 237] with randomized "minibatches" of trajectories. Finally, we performed all computations on a personal computer and did not take advantage of the natural parallel structure of the costly gradient computation step. By taking advantage of this step's parallel structure across trajectories, evaluations at quadrature points, and decomposed spatial domains of the full-order model, we believe that the method can be applied to systems whose state dimensions are much higher than in our jet flow example.

Appendix

8.A Proof of Theorem 8.3.3 (Topology of $\mathcal{P}$)

By Lemma 8.3.1 a function F on $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ is continuous if and only if $F \circ \pi$ is continuous on $\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}$. This allows us to establish the results by working with representatives in $\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}$ rather than abstract subspaces in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$.

To prove that $\mathcal{P}$ is open in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$, consider the function

$$F \circ \pi(\Phi, \Psi) = \frac{\det(\Psi^*\Phi)^2}{\det(\Phi^*\Phi)\det(\Psi^*\Psi)}, \qquad (\Phi, \Psi) \in \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}. \tag{8.57}$$

The function F is well-defined because the above expression does not depend on the representatives Φ, Ψ due to the product rule for determinants. Furthermore, $F \circ \pi$ is continuous on $\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}$ and so it follows that $\mathcal{P}$ is open because it is the pre-image $\mathcal{P} = F^{-1}((0, \infty))$ of the open set $(0, \infty)$ by Proposition 8.2.2.

To prove that $\mathcal{P}$ is dense in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$, consider a pair of subspaces $(V, W) \in (\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}) \setminus \mathcal{P}$ and representatives $(\Phi, \Psi) \in \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}$ such that $\pi(\Phi, \Psi) = (V, W)$. Consider the full-sized singular value decomposition

$$\Phi^*\Psi = U\Sigma Q^T \tag{8.58}$$

and define the continuously parameterized set of matrices

$$\Psi_t = \Psi + t\Phi(\Phi^*\Phi)^{-1}UQ^T, \qquad t \geq 0, \tag{8.59}$$

giving rise to a continuously parameterized set of subspaces $(V, W_t) = \pi(\Phi, \Psi_t) \in \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. We observe that for all $t > 0$, we have

$$\det(\Phi^*\Psi_t) = \det\left(U\Sigma Q^T + tUQ^T\right) = \det(U)\det(Q)\det\left(\Sigma + tI\right) \neq 0. \tag{8.60}$$

Therefore, $(V, W_0) = (V, W) \notin \mathcal{P}$, but $(V, W_t) \in \mathcal{P}$ for all $t > 0$, from which it follows that $\mathcal{P}$ is dense in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$.

Since we are working with manifolds, connectedness and path-connectedness are equivalent. In order to prove the connectedness part of Theorem 8.3.3, we will need the following result:

Lemma 8.A.1. *If $1 \leq r \leq n$ then $\mathcal{G}_{n,r}$ is connected. If $1 \leq r < n$ then $\mathbb{R}_*^{n\times r}$ is connected.*

Proof. Our proof relies on the well-known result that the general linear group GL_n has two connected components, corresponding to matrices with positive and negative determinants [109]. Choose any two subspaces $V_0, V_1 \in \mathcal{G}_{n,r}$ and let $T_0, T_1 \in GL_n$ be chosen such that the first r columns of T_0 span V_0 and the first r columns of T_1 span V_1. Recalling that changing the sign of a column changes the sign of the determinant, flip the sign on the first column of T_0 if necessary to satisfy $\operatorname{sgn}\det T_0 = \operatorname{sgn}\det T_1$. Now there exists a continuous path $t \mapsto A_t \in GL_n$ connecting T_0 at $t = 0$ to T_1 at $t = 1$. Since each $A_t \in GL_n$, its first r columns are linearly independent and span an r-dimensional subspace $V_t \in \mathcal{G}_{n,r}$ The path $t \mapsto V_t$ is a continuous path connecting V_0 at $t = 0$ to V_1 at $t = 1$.

Essentially the same approach is used to connect any two matrices $\Phi_0, \Phi_1 \in \mathbb{R}_*^{n\times r}$ by a continuous path, with the caveat that we must have $r < n$. Let $T_0, T_1 \in GL_n$ be matrices whose first r columns are given by Φ_0 and Φ_1 respectively. Since $r < n$ we may choose the sign of the last column of T_0 so that $\operatorname{sgn}\det T_0 = \operatorname{sgn}\det T_1$. Now the first r columns of the matrices $T_t \in GL_n$ along a continuous path connecting T_0 and T_1 form a continuous path connecting Φ_0 and Φ_1 in $\mathbb{R}_*^{n\times r}$. □

To prove that $\mathcal{P}$ is connected we need only consider the case when $r < n$ since when $r = n$ we obviously have $\mathcal{P} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$, which is connected by Lemma 8.A.1. Choose any (V_0, W_0), $(V_1, W_1) \in \mathcal{P}$ and let (Φ_0, Ψ_0) and (Φ_1, Ψ_1) be representatives of (V_0, W_0) and (V_1, W_1) respectively such that $\Psi_0^*\Phi_0 = I_r$ and $\Psi_1^*\Phi_1 = I_r$. Such representatives may always be found by first choosing any repre-

sentatives $\tilde{\Phi}_0, \tilde{\Psi}_0$ of the subspaces V_0, W_0 and then letting $\Phi_0 = \tilde{\Phi}_0$ and $\Psi_0 = \tilde{\Psi}_0(\Phi_0^*\tilde{\Psi}_0)^{-1}$. We may do the same for Φ_1 and Ψ_1. In order to construct our path in $\mathcal{P}$, we first consider any continuously parameterized matrices $(\tilde{\Phi}_t, \tilde{\Psi}_t) \in \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}$, $0 \leq t \leq 1$ furnished by Lemma 8.A.1 such that $(\tilde{\Phi}_0, \tilde{\Psi}_0) = (\Phi_0, \Psi_0)$ and $(\tilde{\Phi}_1, \tilde{\Psi}_1) = (\Phi_1, \Psi_1)$. Our approach will be to modify these matrices to avoid singularities.

Since $t \mapsto \det(\tilde{\Psi}_t^*\tilde{\Phi}_t)$ is a continuous function, there exists $\varepsilon > 0$ such that for every $t \in [0, \varepsilon) \cup (1-\varepsilon, 1]$, the matrix $\tilde{\Psi}_t^*\tilde{\Phi}_t$ is invertible. Moreover, $\varepsilon > 0$ may be chosen small enough so that

$$\hat{\Phi}_t = \tilde{\Phi}_t(\tilde{\Psi}_t^*\tilde{\Phi}_t)^{-1} \qquad \forall t \in [0, \varepsilon) \cup (1-\varepsilon, 1], \tag{8.61}$$

is sufficiently close to $\tilde{\Phi}_t$ that any convex combination of $\hat{\Phi}_t$ and $\tilde{\Phi}_t$ has linearly independent columns. We observe that on $[0, \varepsilon) \cup (1-\varepsilon, 1]$, $t \mapsto \hat{\Phi}_t$ is continuous and $\tilde{\Psi}_t^*\hat{\Phi}_t = I_r$. Furthermore, $\hat{\Phi}_t$ agrees with the original Φ_0 at $t = 0$ and with Φ_1 at $t = 1$.

Let $\varphi : \mathbb{R} \to [0, 1]$ be a smooth function such that $\varphi(t) = 1$ for every $t \in (-\infty, 8\varepsilon/10] \cup [1 - 8\varepsilon/10, \infty)$ and $\varphi(t) = 0$ for every $t \in [9\varepsilon/10, 1 - 9\varepsilon/10]$. We define the continuous set of matrices

$$\Phi_t = \begin{cases} \tilde{\Phi}_t & t \in [9\varepsilon/10, 1 - 9\varepsilon/10] \\ \varphi(t)\hat{\Phi}_t + (1 - \varphi(t))\tilde{\Phi}_t & \text{otherwise} \end{cases} \tag{8.62}$$

for $0 \leq t \leq 1$ and we observe that Φ_t agrees with the original matrix Φ_0 at $t = 0$ and with Φ_1 at $t = 1$. Now let $\psi : \mathbb{R} \to [0, 1]$ be a smooth function such that $\psi(t) = 1$ for every $t \in (-\infty, 1\varepsilon/10] \cup [1 - 1\varepsilon/10, \infty)$ and $\psi(t) = 0$ for every $t \in [2\varepsilon/10, 1 - 2\varepsilon/10]$. We now define the continuous set of matrices

$$\Psi_t = \begin{cases} \Phi_t & t \in [2\varepsilon/10, 1 - 2\varepsilon/10] \\ \psi(t)\tilde{\Psi}_t + (1 - \psi(t))\Phi_t & \text{otherwise} \end{cases} \tag{8.63}$$

for $0 \leq t \leq 1$ and we observe that Ψ_t agree with the original matrix Ψ_0 at $t = 0$ and with Ψ_1 at $t = 1$. Finally, we observe that

$$\Phi_t^*\Psi_t = \begin{cases} \Phi_t^*\Phi_t & t \in [2\varepsilon/10, 1 - 2\varepsilon/10] \\ \psi(t)I_r + (1 - \psi(t))\hat{\Phi}_t^*\hat{\Phi}_t & \text{otherwise} \end{cases} \tag{8.64}$$

for if $t \in [0, 2\varepsilon/10) \cup (1 - 2\varepsilon/10, 1]$ then $\Phi_t = \hat{\Phi}_t$ and $\hat{\Phi}_t^*\tilde{\Psi}_t = I_r$. Therefore, $\Phi_t^*\Psi_t$ is a positive-definite matrix for every $t \in [0, 1]$ and so $(V_t, W_t) = \pi(\Phi_t, \Psi_t) \in \mathcal{P}$ is a continuous path between

$(V_0, W_0) \in \mathcal{P}$ and $(V_1, W_1) \in \mathcal{P}$.

Finally, we conclude with

Lemma 8.A.2. *The submanifold $\mathcal{P} \subset \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ is diffeomorphic to*

$$\mathbb{P} = \left\{ P \in \mathbb{R}^{n \times n} \ : \ P^2 = P \quad \textit{and} \quad \operatorname{rank}(P) = r \right\}. \tag{8.65}$$

Proof. The map $\phi : \mathcal{P} \to \mathbb{P}$ defined by

$$\phi \circ \pi(\Phi, \Psi) = \Phi(\Psi^* \Phi)^{-1} \Psi^* \tag{8.66}$$

is smooth by Lemma 8.3.1. Moreover, ϕ is injective for if $(V_i, W_i) \in \mathcal{P}$, $i \in \{0, 1\}$ are subspace pairs with representatives $(\Phi_i, \Psi_i) \in \pi^{-1}(V_i, W_i)$ satisfying

$$\Phi_0(\Psi_0^* \Phi_0)^{-1} \Psi_0^* = \Phi_1(\Psi_1^* \Phi_1)^{-1} \Psi_1^*, \tag{8.67}$$

then $V_0 = \operatorname{Range}(\Phi_0) = \operatorname{Range}(\Phi_1) = V_1$ and $W_0 = \operatorname{Range}(\Psi_0) = \operatorname{Range}(\Psi_1) = W_1$. To show that ϕ is surjective, choose any $P \in \mathbb{P}$ and consider a slim singular value decomposition $P = U \Sigma Q^*$. It is clear that Σ is an invertible $r \times r$ diagonal matrix and the condition $P^2 = P$ implies that

$$\Sigma Q^* U \Sigma = \Sigma \quad \Rightarrow \quad Q^* U = \Sigma^{-1}. \tag{8.68}$$

Therefore, $P = U(Q^*U)^{-1}Q^*$ for some $Q, U \in \mathbb{R}_*^{n,r}$ such that $\det(Q^*U) \neq 0$. Taking $(V, W) = \pi(U, Q)$ we obtain $\phi(V, W) = P$.

It now remains to show that $\phi^{-1} : \mathbb{P} \to \mathcal{P}$ is differentiable. Let $\{e_i\}_{i=1}^n$ be an orthonormal basis for the state space $\mathcal{X}$. If $I = \{i_1, \ldots, i_r\} \subset \{1, \ldots, n\}$ is a subset of r indices, let $E_I : \mathbb{R}^r \to \mathcal{X}$ be defined by

$$E_I z = e_{i_1} z_1 + \cdots + e_{i_r} z_r. \tag{8.69}$$

Choose $P \in \mathbb{P}$ and let $I, J \subset \{1, \ldots, n\}$ be sets of indices with $|I| = |J| = r$ elements such that $\operatorname{Range}(P^* E_I) = \operatorname{Range}(P^*)$ and $\operatorname{Range}(P E_J) = \operatorname{Range}(P)$. Here P^* refers to the adjoint of P as an operator $P : \mathcal{X} \to \mathcal{X}$ on the state space $\mathcal{X}$. Since ϕ is bijective, we have

$$P = \phi(\operatorname{Range}(P), \ \operatorname{Range}(P^*)) = (PE_J) \left[(P^* E_I)^* (P E_J) \right]^{-1} (P^* E_I)^*. \tag{8.70}$$

Most importantly, the same sets of indices I and J satisfy the above properties for every $\tilde{P}$ in a sufficiently small neighborhood of P in $\mathbb{P}$. The maps

$$P \mapsto PE_J, \qquad P \mapsto P^* E_I, \tag{8.71}$$

are smooth and so

$$P \mapsto \pi(PE_J, P^* E_I) \tag{8.72}$$

is the smooth inverse of ϕ over a small neighborhood of P in $\mathbb{P}$. Since such a smooth inverse exists near every $P \in \mathcal{P}$ it follows that ϕ is a diffeomorphism. □

8.B Proof of Proposition 8.3.4 (Properties of ROM Solutions)

Any solution of Eq. 8.4 is unique since Eq. 8.4 is smooth. This is a trivial consequence of Grönwall's inequality (Corollary 8.62 in [134]). Suppose that $\hat{x}_0$ and $\hat{x}_1$ are two solutions of Eq. 8.4 over the interval $[t_0, t_{L-1}]$ at the same $(V, W) \in \mathcal{P}$. Since these solutions are are continuous in time, they are contained in some closed ball $\bar{B} \subset \mathcal{X}$. Since $(x, t) \mapsto f(x, u(t))$ is continuously differentiable by Assumption 8.2.1, it is L-Lipschitz in $\bar{B}$ for some finite L and we have

$$\begin{aligned} \|\hat{x}_0(t) - \hat{x}_1(t)\| &\leq \int_{t_0}^{t} \|P_{V,W} \left(f(\hat{x}_0(s), u(s)) - f(\hat{x}_1(s), u(s))\right)\| \, \mathrm{d}s \\ &\leq L \|P_{V,W}\|_{\mathrm{op}} \int_{t_0}^{t} \|\hat{x}_0(s) - \hat{x}_1(s)\| \mathrm{d}s. \end{aligned} \tag{8.73}$$

By Grönwall's inequality, it follows that

$$\|\hat{x}_0(t) - \hat{x}_1(t)\| \leq 0, \tag{8.74}$$

which implies that $\hat{x}_0(t) = \hat{x}_1(t)$ for all $t \in [t_0, t_{L-1}]$.

Suppose that a solution $\hat{x}_0(t) = \hat{x}(t; (V_0, W_0))$ exists for a given $(V_0, W_0) \in \mathcal{P}$. Since $\hat{x}_0$ is continuous over the finite interval $[t_0, t_{L-1}]$, it is bounded and contained in the open ball

$$B = \left\{ x \in \mathcal{X} \ : \ \|x\| < \sup_{t \in [t_0, t_{L-1}]} \|\hat{x}_0(t)\| + 1 \right\}. \tag{8.75}$$

Moreover, by Assumption 8.2.1 it follows that every $(t, x) \mapsto P_{V,W} f(x, u(t))$ with $(V, W) \in \mathcal{P}$ is

bounded and Lipschitz on $\bar{B}$. Therefore, for any $(V, W) \in \mathcal{P}$ such that $P_{V,W} x_0 \in B$, the Picard-Lindelof theorem (Theorem 8.13 in W. G. Kelly A. C. Peterson [134]), ensures the reduced-order model Eq. 8.4 has a unique solution $\hat{x}(t; (V, W))$ in B over an interval $[t_0, \alpha]$ for some $\alpha > t_0$. Moreover, the extension theorem for ODEs (Theorem 8.33 [134]) implies that the solution $\hat{x}(t; (V, W))$ of Eq. 8.4 exists in B for all time $t \geq t_0$, or there is a finite ω so that $\hat{x}(t; (V, W))$ remains in B for $t \in [t_0, \omega)$ and $\hat{x}(t; (V, W)) \to \partial B$ as $t \to \omega^-$. To be precise, the latter means that $\hat{x}(t; (V, W))$ leaves any compact subset of B as $t \to \omega^-$.

For the sake of producing a contradiction, suppose that there is a sequence $\{(V_k, W_k)\}_{k=1}^{\infty}$ such that $(V_k, W_k) \to (V_0, W_0)$ and for which the reduced-order model does not have a solution on $[t_0, t_{L-1}]$. Since the map $\phi : (V, W) \mapsto P_{V,W}$ is smooth by Theorem 8.3.3, we may assume that each (V_k, W_k) is already in a sufficiently small neighborhood of (V_0, W_0) such that $P_{V_k,W_k} x_0 \in B$. Consequently, each reduced-order model solution $\hat{x}_k(t) = \hat{x}(t; (V_k, W_k))$ exist and remains in B over some maximal interval $[t_0, \omega_k)$ with $t_0 < \omega_k < t_{L-1}$ and $\hat{x}_k(t) \to \partial B$ as $t \to \omega_k^-$.

To produce a contradiction, we show that $\hat{x}_k(t)$ remains close to $\hat{x}_0(t)$ over the interval $[t_0, \omega_k)$ for sufficiently large k, which will be at odds with $\hat{x}_k(t) \to \partial B$ as $t \to \omega_k^-$. For $t \in [t_0, \omega_k)$ we have the following bound on the difference between the trajectories

$$\begin{aligned} \|\hat{x}_k(t) - \hat{x}_0(t)\| \leq \|(P_{V_k,W_k} - P_{V_0,W_0}) x_0\| \\ + \int_{t_0}^{t} \|(P_{V_k,W_k} - P_{V_0,W_0}) f(\hat{x}_k(s), u(s))\| \, \mathrm{d}s \\ + \int_{t_0}^{t} \|P_{V_0,W_0} f(\hat{x}_k(s), u(s)) - P_{V_0,W_0} f(\hat{x}_0(s), u(s))\| \, \mathrm{d}s. \end{aligned} \tag{8.76}$$

Let $\|\cdot\|_{\mathrm{op}}$ denote the induced norm (operator norm) and observe that since $(t, x) \mapsto f(x, u(t))$ is continuously differentiable with respect to x by Assumption 8.2.1, there are finite constants M and L such that

$$\|f(x, u(t))\| \leq M \qquad \forall x \in \bar{B}, \ \forall t \in [t_0, t_{L-1}] \tag{8.77}$$

$$\|f(x, u(t)) - f(z, u(t))\| \leq L\|x - z\| \qquad \forall x, z \in \bar{B}, \ \forall t \in [t_0, t_{L-1}]. \tag{8.78}$$

Therefore, we have

$$\|\hat{x}_k(t) - \hat{x}_0(t)\| \leq \|P_{V_k,W_k} - P_{V_0,W_0}\|_{\text{op}} \left(\|x_0\| + M(t_{L-1} - t_0)\right) + L\|P_{V_0,W_0}\|_{\text{op}} \int_{t_0}^{t} \|\hat{x}_k(s) - \hat{x}_0(s)\| \, \mathrm{d}s. \quad (8.79)$$

Applying Grönwall's inequality (Corollary 8.62 in [134]), we see that

$$\|\hat{x}_k(t) - \hat{x}_0(t)\| \leq \|P_{V_k,W_k} - P_{V_0,W_0}\|_{\text{op}} \left(\|x_0\| + M(t_{L-1} - t_0)\right) e^{L\|P_{V_0,W_0}\|_{\text{op}} t}. \quad (8.80)$$

Since $\phi : (V, W) \mapsto P_{V,W}$ is continuous,

$$(V_k, W_k) \to (V_0, W_0) \quad \Rightarrow \quad \|P_{V_k,W_k} - P_{V_0,W_0}\|_{\text{op}} \to 0, \quad (8.81)$$

and so Eq. 8.80 implies that $\|\hat{x}_k(t) - \hat{x}_0(t)\| \to 0$ uniformly over $t \in [t_0, \omega_k)$ as $k \to \infty$. In particular, we may take K sufficiently large so that for any $k \geq K$ then

$$\|\hat{x}_k(t) - \hat{x}_0(t)\| \leq \frac{1}{2} \qquad \forall t \in [t_0, \omega_k), \quad (8.82)$$

contradicting the fact that $\hat{x}_k(t) \to \partial B$ as $t \to \omega_k^-$. Therefore, there is an open neighborhood of (V_0, W_0) in $\mathcal{P}$ in which the reduced order model Eq. 8.4 has a unique solution over the time interval $[t_0, t_{L-1}]$, which establishes the openness of $\mathcal{D}$ in $\mathcal{P}$.

Since $\mathcal{D}$ is open in $\mathcal{P}$ it follows that there is a set $\mathcal{D}' \in \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ such that $\mathcal{D} = \mathcal{D}' \cap \mathcal{P}$. Since $\mathcal{P}$ is open in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ by Theorem 8.3.3, it follows that $\mathcal{D}$ is open in $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ since it is a finite intersection of open sets.

Now, let us turn our attention to proving that $\mathcal{D} = \mathcal{P}$ when f has bounded x-derivatives. Since the partial derivatives of f with respect to x are bounded, it follows that for any $(V, W) \in \mathcal{P}$ there is a constant L such that

$$\|P_{V,W} f(x_1, u(t)) - P_{V,W} f(x_2, u(t))\| \leq L\|x_1 - x_2\| \qquad \forall x_1, x_2 \in \mathbb{R}^n, \quad \forall t \in \mathbb{R} \quad (8.83)$$

and so $(x, t) \mapsto P_{V,W} f(x, t)$ is Lipschitz in x uniformly over t. A trivial modification of Theorem 7.3 in H. Brezis [30] shows that a solution of the reduced-order model Eq. 8.4 exists on the interval $[t_0, \infty)$. As a consequence, $\mathcal{D} = \mathcal{P}$.

Now we shall establish the differentiability of $\hat{x}(t; (V, W))$ at each fixed $t \in [t_0, t_{L-1}]$ with respect

to $(V, W) \in \mathcal{D}$. Recall that by Theorem 8.3.3, the set of rank-r projection matrices $\mathbb{P}$ is smoothly diffeomorphic to the $2nr - 2r^2$ dimensional submanifold $\mathcal{P} \subset \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. Let $\psi : \mathbb{R}^{2nr-2r^2} \to \mathcal{U} \subset \mathcal{D}$ be a local parameterization of a an open subset $\mathcal{U} \subset \mathcal{D}$. Letting $\phi : (V, W) \mapsto P_{V,W}$ be the diffeomorphism established by Theorem 8.3.3, the map $P = \phi \circ \psi$ is a smooth parameterization of the open subset $\phi(\mathcal{U}) \subset \mathbb{P}$. It suffices to show that the solution $\hat{x}(t; \psi(p_0))$ is continuously differentiable with respect to $p_0 \in \mathbb{R}^{2nr-2r^2}$.

We define the augmented state variable $w = (x, p) \in \mathbb{R}^n \times \mathbb{R}^{2nr-2r^2}$ whose dynamics are described by

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, w = F(w, t) := \begin{bmatrix} P(p) f(x, u(t)) \\ 0_{2nr-r^2} \end{bmatrix} \qquad w(0) = w_0. \tag{8.84}$$

Clearly, we have $w(t; w_0) = \hat{x}(t; \psi(p_0))$ when $w_0 = (P(p_0)x_0, p_0)$. It is also clear from Assumption 8.2.1 that F is continuously differentiable with respect to w, and so by Theorem 8.43 in [134], it follows that $w(t; w_0)$ is continuously differentiable with respect to w_0. This proves that $\hat{x}(t; (V, W))$ is continuously differentiable with respect to (V, W) since $w_0 = (P(p_0)x_0, p_0)$ is continuously differentiable with respect to p_0.

Finally, suppose that $\{(V_k, W_k)\}_{k=1}^{\infty} \subset \mathcal{D}$ is a sequence approaching $(V_k, W_k) \to (V_0, W_0) \in \mathcal{P} \backslash \mathcal{D}$. Denote $\hat{x}_k(t) = \hat{x}(t; (V_k, W_k))$ and $\hat{x}_0(t) = \hat{x}(t; (V_0, W_0))$, and let $[t_0, \omega_R)$ be the maximum interval of existence for $\hat{x}_0$ in an open ball $B_R \subset \mathcal{X}$ of radius $R > \|P_{V_0,W_0} x_0\|$ centered about the origin. Here we have again made use of the extension theorem for solutions of ordinary differential equations (Theorem 8.33 [134]). It is clear that since $(x, t) \mapsto f(x, u(t))$ is continuously differentiable with respect to x by Assumption 8.2.1, it is L-Lipschitz and bounded by M on $\overline{B_R}$ for some finite L and M. Let us suppose that k is sufficiently large so that $P_{V_k,W_k} x_0 \in B_R$ and let $[t_0, \omega_k)$ denote the maximum interval of existence for $\hat{x}_k(t)$ in B_R. There are two possibilities, either $\omega_k < \omega_R$ which implies that $\sup_{t \in [t_0, t_{L-1}]} \|\hat{x}_k(t)\| \geq R$, or $\omega_k \geq \omega_R$ which implies that $\hat{x}_k(t) \in B_R$ for every $t \in [t_0, \omega_R)$. In the second case, the same Grönwall argument we used above shows that that

$$\|\hat{x}_k(t) - \hat{x}_0(t)\| \leq \|P_{V_k,W_k} - P_{V_0,W_0}\|_{\mathrm{op}} \left(\|x_0\| + M(t_{L-1} - t_0)\right) e^{L\|P_{V_0,W_0}\|_{\mathrm{op}} t} \tag{8.85}$$

for every $t \in [t_0, \omega_R)$. Since, $P_{V_k,W_k} \to P_{V_0,W_0}$, we may take k sufficiently large so that the above inequality implies that $\|\hat{x}_k(t) - \hat{x}_0(t)\| \leq R/2$ for every $t \in [t_0, \omega_R)$ when $\hat{x}_k(t) \in B_R$ for every $t \in [t_0, \omega_R)$. Since $\hat{x}_0(t) \to \partial B_R$ as $t \to \omega_R^-$, we must have $\sup_{t \in [t_0, t_{L-1}]} \|\hat{x}_k(t)\| \geq R/2$ in the case when $\hat{x}_k(t) \in B_R$ for every $t \in [t_0, \omega_R)$. It follows that for sufficiently large k, we always have

$\sup_{t\in[t_0,t_{L-1}]} \|\hat{x}_k(t)\| \geq R/2$. Since R was arbitrary, it follows that

$$\sup_{t\in[t_0,t_{L-1}]} \|\hat{x}_k(t)\| \to \infty \quad \text{as} \quad k \to \infty. \tag{8.86}$$

Furthermore, the sup is actually a max because the trajectories $\hat{x}_k$ are continuous. This completes the proof of Proposition 8.3.4.

8.C Regularization and Existence of a Minimizer

Proof of Theorem 8.3.5 (Regularization). We begin by showing that $\rho(V,W) \to +\infty$ as $(V,W) \to (V_0,W_0) \in \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. Let $\Phi_0, \Psi_0 \in \pi^{-1}(V_0,W_0)$. By the local submersion theorem [106], there is an open neighborhood $\mathcal{V} \subset \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ containing (V_0,W_0) and an open neighborhood $\mathcal{U} \subset \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}$ containing (Φ_0,Ψ_0) together with local parameterizations $\phi : \mathbb{R}^{2nr} \to \mathcal{U}$ and $\psi : \mathbb{R}^{2nr-2r^2} \to \mathcal{V}$ of these neighborhoods such that $(\Phi_0,\Psi_0) = \phi(0)$, $(V_0,W_0) = \psi(0)$, and

$$(\psi^{-1} \circ \pi \circ \phi)(x_1,\ldots,x_{2nr}) = (x_1,\ldots,x_{2nr-2r^2}). \tag{8.87}$$

Since $(V_n,W_n) \to (V_0,W_0)$ there exist N such that for every $n \geq N$, we have $(V_n,W_n) \in \mathcal{V}$. Let $z^{(n)} = \psi^{-1}(V_n,W_n)$ be the coordinates of these subspace pairs for $n \geq N$ and let us choose the representatives of these subspaces whose coordinates are $x^{(n)} = (z^{(n)},0,\ldots,0) \in \mathbb{R}^{2nr}$, i.e., let $(\Phi_n,\Psi_n) = \phi(z^{(n)},0,\ldots,0)$. It is clear that $z^{(n)} \to 0$ as $n \to \infty$ and so we have $(\Phi_n,\Psi_n) \to (\Phi_0,\Psi_0)$ as $n \to \infty$ by continuity of the local parameterizaions. Since the determinant is a continuous function, we have

$$\begin{aligned}
\lim_{n\to\infty} \det(\Psi_n^*\Phi_n) &= \det(\Psi_0^*\Phi_0) = 0,\\
\lim_{n\to\infty} \det(\Phi_n^*\Phi_n) &= \det(\Phi_0^*\Phi_0) > 0,\\
\lim_{n\to\infty} \det(\Psi_n^*\Psi_n) &= \det(\Psi_0^*\Psi_0) > 0
\end{aligned} \tag{8.88}$$

and so it follows that

$$\rho(V_n,W_n) = \rho \circ \pi(\Phi_n,\Psi_n) \to \infty \quad \text{as} \quad n \to \infty. \tag{8.89}$$

Now we seek a minimum of ρ by first considering the function $F : \mathcal{G}_{n,r} \times \mathcal{G}_{n,r} \to \mathbb{R}$ defined by

$$F \circ \pi(\Phi,\Psi) = \frac{\det(\Psi^*\Phi)^2}{\det(\Phi^*\Phi)\det(\Psi^*\Psi)} \tag{8.90}$$

and observing that it is continuous on $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. Since $\mathcal{G}_{n,r}$ is compact, it follows that $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ is also compact, and so F attains its maximum. Moreover, if $V = W$ then obviously $(V, W) \in \mathcal{P}$ and choosing the columns of $\Phi = \Psi$ to be an orthonormal basis for V, we find that

$$F(V,V) = \frac{\det(\Phi^*\Phi)^2}{\det(\Phi^*\Phi)\det(\Phi^*\Phi)} = 1 \quad \Rightarrow \quad \rho(V,V) = -\log F(V,V) = 0. \tag{8.91}$$

Consequently, the maximum value of F is at least 1 and so any subspace pair $(V_{\mathrm{m}}, W_{\mathrm{m}})$ that maximizes F must lie in $\mathcal{P} = F^{-1}((0,\infty))$ and also minimize $R = -\log F$. Since ρ is a smooth function on the open set $\mathcal{P}$ (see Theorem 8.3.3), a necessary condition for $(V_{\mathrm{m}}, W_{\mathrm{m}})$ to be a minimizer of ρ is $\mathrm{D}\,\rho(V_{\mathrm{m}}, W_{\mathrm{m}})(\xi, \eta) = 0$ for every $(\xi, \eta) \in T_{(V_{\mathrm{m}}, W_{\mathrm{m}})}\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. Or, in terms of representatives, if $(\Phi_{\mathrm{m}}, \Phi_{\mathrm{m}}) \in \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{n\times r}$ is a representative of subspaces $(V_{\mathrm{m}}, W_{\mathrm{m}}) = \pi(\Phi_{\mathrm{m}}, \Phi_{\mathrm{m}}) \in \mathcal{P}$ that minimize ρ, then for every pair of matrices $(X, Y) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r}$ we have

$$\begin{aligned} 0 = \mathrm{D}(\rho \circ \pi)(\Phi_{\mathrm{m}}, \Psi_{\mathrm{m}})(X, Y) &= \operatorname{Tr}\left\{(\Phi_{\mathrm{m}}^*\Phi_{\mathrm{m}})^{-1}(\Phi_{\mathrm{m}}^*X + X^*\Phi_{\mathrm{m}})\right\} \\ &\quad + \operatorname{Tr}\left\{(\Psi_{\mathrm{m}}^*\Psi_{\mathrm{m}})^{-1}(\Psi_{\mathrm{m}}^*Y + Y^*\Psi_{\mathrm{m}})\right\} - 2\operatorname{Tr}\left\{(\Psi_{\mathrm{m}}^*\Phi_{\mathrm{m}})^{-1}(\Psi_{\mathrm{m}}^*X + Y^*\Phi_{\mathrm{m}})\right\}. \end{aligned} \tag{8.92}$$

Applying permutation identities for the trace and collecting terms we have

$$\begin{aligned} 0 = \operatorname{Tr}\left\{\left[(\Phi_{\mathrm{m}}^*\Phi_{\mathrm{m}})^{-1}\Phi_{\mathrm{m}}^* - (\Psi_{\mathrm{m}}^*\Phi_{\mathrm{m}})^{-1}\Psi_{\mathrm{m}}^*\right]X\right\} \\ + \operatorname{Tr}\left\{Y^*\left[\Psi_{\mathrm{m}}(\Psi_{\mathrm{m}}^*\Psi_{\mathrm{m}})^{-1} - \Phi_{\mathrm{m}}(\Psi_{\mathrm{m}}^*\Phi_{\mathrm{m}})^{-1}\right]\right\} \end{aligned} \tag{8.93}$$

for every $(X, Y) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r}$, which implies that

$$(\Phi_{\mathrm{m}}^*\Phi_{\mathrm{m}})^{-1}\Phi_{\mathrm{m}}^* = (\Psi_{\mathrm{m}}^*\Phi_{\mathrm{m}})^{-1}\Psi_{\mathrm{m}}^* \quad \text{and} \quad \Psi_{\mathrm{m}}(\Psi_{\mathrm{m}}^*\Psi_{\mathrm{m}})^{-1} = \Phi_{\mathrm{m}}(\Psi_{\mathrm{m}}^*\Phi_{\mathrm{m}})^{-1}. \tag{8.94}$$

The above is true only if $\operatorname{Range}\Phi_{\mathrm{m}} = \operatorname{Range}\Psi_{\mathrm{m}}$; and so a necessary condition for $(V_{\mathrm{m}}, W_{\mathrm{m}})$ to minimize ρ over $\mathcal{P}$ is that $V_{\mathrm{m}} = W_{\mathrm{m}}$. But we have already seen that $\rho(V, W) = 0$ when $V = W$, proving that zero is the minimum value of ρ, and the minimum is attained if and only if the subspaces (V, W) satisfy $V = W$. □

Corollary 8.C.1 (Existence of a Minimizer)**.** *Let $\mathcal{D}$ be as in Proposition 8.3.4, and take $\gamma > 0$. We assume that $\mathcal{D}$ is nonempty. Then a minimizer of Eq. 8.5 exists in $\mathcal{D}$; that is, there exists a pair of*

subspaces $(V_{op}, W_{op}) \in \mathcal{D}$ *such that*

$$J(V_{op}, W_{op}) \leq J(V, W), \qquad \textit{for all } (V, W) \in \mathcal{D}. \tag{8.95}$$

Let the set of subspaces defining orthogonal projection operators be denoted

$$\mathcal{P}_0 = \{(V, W) \in \mathcal{P} \ : \ V = W\} \tag{8.96}$$

and assume that $\mathcal{D} \cap \mathcal{P}_0$ *is nonempty. Then, as* $\gamma \to \infty$*, any choice of minimizers, denoted* $(V_{op}(\gamma), W_{op}(\gamma))$*, approaches* $\mathcal{D} \cap \mathcal{P}_0$*. Furthermore, the corresponding cost, temporarily denoted* $J(V, W;\ \gamma)$ *to emphasize the dependence on* γ*, approaches the minimum over* $\mathcal{D} \cap \mathcal{P}_0$*, i.e.,*

$$\lim_{\gamma \to \infty} J(V_{op}(\gamma), W_{op}(\gamma);\ \gamma) \leq J(V, V), \qquad \textit{for all } V \in \mathcal{G}_{n,r}. \tag{8.97}$$

Note that $J(V, V)$ *does not depend on* γ*, since* $\rho(V, V) = 0$*.*

Proof. Choose a sequence
$\{(V_n, W_n)\}_{n=1}^{\infty}$ in $\mathcal{D}$ such that

$$\lim_{n \to \infty} J(V_n, W_n) = \inf_{(V,W) \in \mathcal{D}} J(V, W) < \infty. \tag{8.98}$$

Since the Grassmann manifold $\mathcal{G}_{n,r}$ is compact, it follows that $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ is compact, and so there exists a convergent subsequence $(V_{n_k}, W_{n_k}) \to (V_0, W_0)$ for some $(V_0, W_0) \in \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. We must have $(V_0, W_0) \in \mathcal{P}$; for if not, Theorem 8.3.5 tells us that $\rho(V_{n_k}, W_{n_k}) \to +\infty$ and so $J(V_{n_k}, W_{n_k}) \to +\infty$ as $k \to \infty$ because $L_y \geq 0$, contradicting Eq. 8.98. Furthermore, $(V_0, W_0) \in \mathcal{D}$, for if not then Proposition 8.3.4 and Assumption 8.3.6 imply that $J(V_{n_k}, W_{n_k}) \to +\infty$ as $k \to \infty$, contradicting Eq. 8.98. The cost function J defined by Eq. 8.5 is continuously differentiable on $\mathcal{D}$ because the ROM solution at each sample time $(V, W) \mapsto \hat{x}(t_i, (V, W))$ is continuously differentiable by Proposition 8.3.4 and the regularization function defined by Eq. 8.13 is smooth. Since J is continuous on $\mathcal{D}$, we have

$$\inf_{(V,W) \in \mathcal{D}} J(V, W) = \lim_{k \to \infty} J(V_{n_k}, W_{n_k}) = J(V_0, W_0), \tag{8.99}$$

proving that (V_0, W_0) achieves the minimum value of J over $\mathcal{D}$.

To prove the second claim about the behavior as $\gamma \to \infty$, we begin by observing that minimizing

J over $\mathcal{P}_0$ is equivalent to minimizing $V \mapsto J(V,V)$ over $\mathcal{G}_{n,r}$. By the convention in Assumption 8.3.6, we have $J(V,W) = \infty$ for any $(V,W) \in \mathcal{P}_0 \setminus \mathcal{D}$. In addition, this minimization does not depend on γ because $\rho(V,V) = 0$ by Theorem 8.3.5. Let us begin by showing that a minimizer of $V \mapsto J(V,V)$ over $\mathcal{G}_{n,r}$ exists. Let $\{(V_k,V_k)\}_{k=1}^{\infty} \subset \mathcal{D} \cap \mathcal{P}_0$ be a sequence such that

$$J(V_k,V_k) \to \inf_{V \in \mathcal{G}_{n,r}} J(V,V) < \infty \quad \text{as} \quad k \to \infty. \tag{8.100}$$

Since $\mathcal{G}_{n,r}$ is compact, we may pass to a convergent subsequence, still denoted by $\{(V_k,V_k)\}_{k=1}^{\infty}$, such that $V_k \to V_0 \in \mathcal{G}_{n,r}$. Clearly, we have $(V_0,V_0) \in \mathcal{P}_0$. If $(V_0,V_0) \notin \mathcal{D}$ then Proposition 8.3.4 and Assumption 8.3.6 imply that $J(V_k,V_k) \to \infty$. But this contradicts the fact that the sequence $\{J(V_k,V_k)\}_{k=1}^{\infty}$ approaches the infemum of J over $\mathcal{P}_0$, which is finite if $\mathcal{D} \cap \mathcal{P}_0 \neq \emptyset$. Therefore, $(V_0,V_0) \in \mathcal{D} \cap \mathcal{P}_0$ and since J is continuous over $\mathcal{D}$ it follows that

$$\inf_{V \in \mathcal{G}_{n,r}} J(V,V) = \lim_{k \to \infty} J(V_k,V_k) = J(V_0,V_0), \tag{8.101}$$

i.e., (V_0,V_0) achieves the minimum value of J over $\mathcal{P}_0$.

Suppose, for the sake of producing a contradiction, that there is an open neighborhood $\mathcal{U} \subset \mathcal{D}$ containing $\mathcal{D} \cap \mathcal{P}_0$ such that for every $\Gamma > 0$, there exists $\gamma \geq \Gamma$ such that $(V_{\mathrm{op}}(\gamma), W_{\mathrm{op}}(\gamma)) \notin \mathcal{U}$. Then $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r} \setminus \mathcal{U}$ is a closed subset of $\mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$ and hence is compact. By the same argument presented above, ρ attains its minimum over $\mathcal{P} \setminus \mathcal{U}$, and this value is strictly greater than zero by Theorem 8.3.5. Consequently, we would have

$$J(V_0,V_0) \geq J(V_{\mathrm{op}}(\gamma), W_{\mathrm{op}}(\gamma);\ \gamma) \geq \gamma \min_{(V,W) \in \mathcal{P} \setminus \mathcal{U}} \rho(V,W) \to \infty \quad \text{as} \quad \gamma \to \infty, \tag{8.102}$$

contradicting the fact that $J(V_0,V_0) < \infty$. Therefore, for every open neighborhood $\mathcal{U} \subset \mathcal{D}$ of $\mathcal{D} \cap \mathcal{P}_0$, there exists $\Gamma > 0$ such that for every $\gamma \geq \Gamma$, we have $(V_{\mathrm{op}}(\gamma), W_{\mathrm{op}}(\gamma)) \in \mathcal{U}$.

Finally, suppose for the sake of producing a contradiction that there exists $\varepsilon > 0$ such that for every $\Gamma > 0$ there exists $\gamma \geq \Gamma$ such that $J(V_{\mathrm{op}}(\gamma), W_{\mathrm{op}}(\gamma);\ \gamma) \leq J(V_0,V_0) - \varepsilon$. By continuity of the objective on $\mathcal{D}$, we know that the non-empty set

$$\mathcal{U} = \{(V,W) \in \mathcal{D} \ :\ J(V,W;\ 0) > J(V_0,V_0) - \varepsilon\} \tag{8.103}$$

is open in $\mathcal{D}$ and contains $\mathcal{D} \cap \mathcal{P}_0$. And so for every $\Gamma > 0$ we have a $\gamma \geq \Gamma$ such that

$$J(V_{\text{op}}(\gamma), W_{\text{op}}(\gamma);\ 0) \leq J(V_{\text{op}}(\gamma), W_{\text{op}}(\gamma);\ \gamma) \leq J(V_0, V_0) - \varepsilon, \tag{8.104}$$

which implies that $(V_{\text{op}}(\gamma), W_{\text{op}}(\gamma)) \notin \mathcal{U}$, contradicting the fact that

$$(V_{\text{op}}(\gamma), W_{\text{op}}(\gamma)) \to \mathcal{D} \cap \mathcal{P}_0 \quad \text{as} \quad \gamma \to \infty. \tag{8.105}$$

Therefore, we conclude that $J(V_{\text{op}}(\gamma), W_{\text{op}}(\gamma);\ \gamma) \to J(V_0, V_0)$ as $\gamma \to \infty$. □

8.D Adjoint-Based Gradient and Required Terms

Proof of Theorem 8.4.1 (Adjoint-Based Gradient). We shall compute the gradients of the component functions

$$J_i(C, z_0) := L_y(\tilde{g}(z(t_i);\ C) - y_i) \tag{8.106}$$

and use linear superposition to construct the gradient of Eq. 8.16. Consider a small perturbation $\delta z(t)$ about the trajectory $z(t)$ due to small perturbations of the independent variables $\delta C \in T_C \bar{\mathcal{M}}$ and $\delta z_0 \in \mathbb{R}^r$ governed by the linearized dynamics

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\,\delta z(t) - F(t)\delta z(t) = S(t)\delta C. \tag{8.107}$$

with perturbations to the observables described by

$$\delta y(t) = H(t)\delta z(t) + T(t)\delta C. \tag{8.108}$$

The resulting perturbation of each component of the objective is given by

$$\begin{aligned}\delta J_i = \langle H(t_i)^* \nabla L_y(\tilde{g}\,(z(t_i);\ C) - y_i),\ \delta z(t_i)\rangle \\ + \langle T(t_i)^* \nabla L_y(\tilde{g}\,(z(t_i);\ C) - y_i),\ \delta C\rangle_C\,,\end{aligned} \tag{8.109}$$

and we wish to express its dependence explicitly on the optimization variables. The second term's dependence on δC is trivial, so we will focus on revealing the implicit dependence of the first term on the optimization variables. To this end, we denote the first term of Eq. 8.109 by Δ_i and we

construct a signal $\lambda_i(t)$ so that

$$\Delta_i = \langle \lambda_i(t_0),\ \delta z(t_0) \rangle + \int_{t_0}^{t_i} \langle \lambda_i(t),\ S(t)\delta C \rangle\, dt. \tag{8.110}$$

This allows us to write the perturbation of the sub-objective in terms of inner products between the gradients and perturbations in each optimization variable:

$$\Delta_i = \langle \lambda_i(t_0),\ \delta z(t_0) \rangle + \left\langle \int_{t_0}^{t_i} S(t)^* \lambda_i(t) dt,\ \delta C \right\rangle. \tag{8.111}$$

To construct $\lambda_i(t)$, we substitute the linearized dynamics Eq. 8.107 into Eq. 8.110 and integrate by parts

$$\begin{aligned} \Delta_i &= \langle \lambda_i(t_0),\ \delta z(t_0) \rangle + \int_{t_0}^{t_i} \left\langle \lambda_i(t),\ \frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, \delta z(t) - F(t)\delta z(t) \right\rangle dt \\ &= \langle \lambda_i(t_i),\ \delta z(t_i) \rangle + \int_{t_0}^{t_i} \left\langle -\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, \lambda_i(t) - F(t)^* \lambda_i(t),\ \delta z(t) \right\rangle dt. \end{aligned} \tag{8.112}$$

Equating with the first term of Eq. 8.109 for all signals $\delta z(t)$, we find that the adjoint variable $\lambda_i(t)$ must satisfy the sub-objective adjoint dynamics

$$-\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, \lambda_i(t) = F(t)^* \lambda_i(t), \qquad \lambda_i(t_i) = H(t_i)^* \nabla L_y(\tilde{g}\,(z(t_i);\ C) - y_i). \tag{8.113}$$

Finally, by linear superposition we can write the variation of the entire objective in terms of the perturbations to the optimization variables as

$$\begin{aligned} \delta \bar{J}_0 = {} & \langle \lambda(t_0),\ \delta z(t_0) \rangle \\ & + \left\langle \int_{t_0}^{t_{L-1}} S(t)^* \lambda(t) dt + \sum_{i=0}^{L-1} T(t_i)^* \nabla L_y(\tilde{g}\,(z(t_i);\ C) - y_i),\ \delta C \right\rangle_C \end{aligned} \tag{8.114}$$

using an adjoint variable $\lambda(t) = \sum_{i=0}^{L-1} \chi_{[t_0,t_i]}(t)\lambda_i(t)$ satisfying

$$-\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\, \lambda(t) = F(t)^* \lambda(t), \qquad \lambda(t_i) = \lim_{t \to t_i^+} \lambda(t) + H(t_i)^* \nabla L_y(\tilde{g}\,(z(t_i);\ C) - y_i), \tag{8.115}$$

for $i = 0, \ldots, L-2$, and $\lambda(t_{L-1}) = H(t_{L-1})^* \nabla L_y(\tilde{g}\,(z(t_{L-1});\ C) - y_{L-1})$. □

Proof of Corollary 8.4.2 (Known Initial Condition). By the chain rule, we have

$$\frac{\partial}{\partial C}\bar{J}(C)\delta C = \frac{\partial}{\partial C}\bar{J}_0(C, z_0)\delta C + \frac{\partial}{\partial z_0}\bar{J}_0(C, z_0)\frac{\partial}{\partial C}z_0(C)\delta C \tag{8.116}$$

$$= \left\langle \nabla_C \bar{J}_0(C, z_0),\ \delta C \right\rangle_C + \left\langle \nabla_{z_0} \bar{J}_0(C, z_0),\ \frac{\partial}{\partial C}z_0(C)\delta C \right\rangle \tag{8.117}$$

$$= \left\langle \nabla_C \bar{J}_0(C, z_0) + \left(\frac{\partial}{\partial C}z_0(C)\right)^* \nabla_{z_0} \bar{J}_0(C, z_0),\ \delta C \right\rangle_C \tag{8.118}$$

for every $\delta C \in T_C\bar{\mathcal{M}}$. □

Proof of Proposition 8.4.4 (Required Terms for Gradient). Our proof of each expression follows directly from the definition of the adjoint of a linear operator between finite-dimensional real inner product spaces. Choosing a pair of vectors $v, w \in \mathbb{R}^r$, we have

$$\begin{aligned} \langle F(t)v,\ w\rangle &= \left(\frac{\partial}{\partial z}\tilde{f}(z(t), u(t);\ (\Phi, \Psi))v\right)^T w \\ &= \left\langle v,\ \left(\frac{\partial}{\partial z}\tilde{f}(z(t), u(t);\ (\Phi, \Psi))\right)^T w \right\rangle, \end{aligned} \tag{8.119}$$

which implies Eq. 8.26. In precisely the same way we obtain Eq. 8.28.

For the next parts, we will need the following:

Lemma 8.D.1. *Letting $A : \mathbb{R}^p \to \mathbb{R}^q$ and $L : \mathbb{R}^q \to \mathcal{X}$ be linear operators, we have*

$$(LA)^* = A^T L^*. \tag{8.120}$$

If $M : \mathcal{X} \to \mathbb{R}^p$ is another linear operator, then we have

$$(AM)^* = M^* A^T. \tag{8.121}$$

Proof. Let $v \in \mathbb{R}^p$ and $w \in \mathcal{X}$ and observe that

$$\begin{aligned} \langle LAv,\ w\rangle_{\mathcal{X}} &= \langle Av,\ L^* w\rangle_{\mathbb{R}^q} \\ &= \left\langle v,\ A^T L^* w\right\rangle_{\mathbb{R}^p}, \end{aligned} \tag{8.122}$$

from which we conclude $(LA)^* = A^T L^*$. Observing that $M^* : \mathbb{R}^p \to \mathcal{X}$, we use the above result to show

$$\left(M^* A^T\right)^* = AM, \tag{8.123}$$

from which we conclude $M^*A^T = \left(M^*A^T\right)^{**} = (AM)^*$. $\square$

Now we consider a vector $(X, Y) \in T_{(\Phi,\Psi)}\bar{\mathcal{M}}$ and a vector $w \in \mathbb{R}^{\dim y}$, yielding

$$\begin{aligned}\langle T(t)(X,Y),\ w\rangle &= \left\langle \frac{\partial}{\partial x} g(\Phi z(t)) X z(t),\ w \right\rangle \\ &= \operatorname{Tr}\left(z(t) w^T \frac{\partial}{\partial x} g(\Phi z(t)) X \right)\end{aligned} \tag{8.124}$$

Applying Lemma 8.D.1, we obtain

$$\begin{aligned}\langle T(t)(X,Y),\ w\rangle &= \operatorname{Tr}\left\{ \left[\left(\frac{\partial}{\partial x} g(\Phi z(t)) \right)^* w z(t)^T \right]^* X \right\} \\ &= \operatorname{Tr}\left\{ (\Phi^*\Phi)^{-1} \left[\left(\frac{\partial}{\partial x} g(\Phi z(t)) \right)^* w z(t)^T (\Phi^*\Phi) \right]^* X \right\} \\ &= \left\langle (X,\ Y),\ \left(\left(\frac{\partial}{\partial x} g(\Phi z(t)) \right)^* w z(t)^T (\Phi^*\Phi),\ 0 \right) \right\rangle_{(\Phi,\Psi)},\end{aligned} \tag{8.125}$$

from which we conclude that Eq. 8.29 holds for all $w \in \mathbb{R}^{\dim y}$.

Consider a vector $(X, Y) \in T_{(\Phi,\Psi)}\bar{\mathcal{M}}$ and a vector $v \in \mathbb{R}^r$, and observe that

$$\begin{aligned}&\langle S(t)(X,Y),\ v\rangle \\ &\quad= \left\langle \left(Y^* - Y^*\Phi\Psi^* - \Psi^* X \Psi^*\right) f(\Phi z(t), u(t)) + \Psi^* \frac{\partial}{\partial x} f(\Phi z(t), u(t)) X z(t),\ v \right\rangle \\ &\quad= \langle Y^* \left(I - \Phi\Psi^*\right) f(\Phi z(t), u(t)),\ v\rangle - \langle \Psi^* f(\Phi z(t), u(t)),\ X^*\Psi v\rangle \\ &\qquad + \left\langle z(t),\ X^* \left(\frac{\partial}{\partial x} f(\Phi z(t), u(t)) \right)^* \Psi v \right\rangle.\end{aligned} \tag{8.126}$$

Each term in Eq. 8.126 is given by a trace, in particular,

$$\langle Y^* \left(I - \Phi\Psi^*\right) f(\Phi z(t), u(t)),\ v\rangle = \operatorname{Tr}\left[Y^* \left(I - \Phi\Psi^*\right) f(\Phi z(t), u(t)) v^T \right]. \tag{8.127}$$

$$\langle \Psi^* f(\Phi z(t), u(t)),\ X^*\Psi v\rangle = \operatorname{Tr}\left[X^*\Psi v \left(\Psi^* f(\Phi z(t), u(t))\right)^T \right] \tag{8.128}$$

$$\left\langle z(t),\ X^* \left(\frac{\partial}{\partial x} f(\Phi z(t), u(t)) \right)^* \Psi v \right\rangle = \operatorname{Tr}\left[X^* \left(\frac{\partial}{\partial x} f(\Phi z(t), u(t)) \right)^* \Psi v z(t)^T \right]. \tag{8.129}$$

Substituting back into Eq. 8.126 and combining terms on X^* and Y^* we obtain

$$\begin{aligned}\langle S(t)(X,Y),\ v\rangle \\ &= \operatorname{Tr}\left\{X^*\left[\left(\frac{\partial}{\partial x}f(\Phi z(t),u(t))\right)^*\Psi v z(t)^T - \Psi v\left(\Psi^* f(\Phi z(t),u(t))\right)^T\right]\right\} \\ &\quad + \operatorname{Tr}\left\{Y^*\left(I-\Phi\Psi^*\right)f(\Phi z(t),u(t))v^T\right\}. \end{aligned} \tag{8.130}$$

Recalling that the time derivative of the reduced-order model is given by
$\tilde{f}(z(t),u(t)) = \Psi^* f(\Phi z(t),u(t))$ yields Eq. 8.27.

Now, consider a vector $(X,Y) \in T_{(\Phi,\Psi)}\bar{\mathcal{M}}$ and a vector $v \in \mathbb{R}^r$, and observe that

$$\begin{aligned}\left\langle \frac{\partial}{\partial(\Phi,\Psi)} z_0(\Phi,\Psi)(X,Y),\ v\right\rangle &= \langle Y^* x_0 - Y^*\Phi\Psi^* x_0 - \Psi^* X\Psi^* x_0,\ v\rangle \\ &= \operatorname{Tr}\left[Y^*\left(x_0-\Phi\Psi^* x_0\right)v^T\right] - \operatorname{Tr}\left[\Psi^* X\Psi^* x_0 v^T\right] \\ &= \operatorname{Tr}\left[Y^*\left(x_0-\Phi\Psi^* x_0\right)v^T\right] - \operatorname{Tr}\left[v(\Psi^* x_0)^T X^*\Psi\right] \\ &= \operatorname{Tr}\left[Y^*\left(x_0-\Phi\Psi^* x_0\right)v^T\right] - \operatorname{Tr}\left[X^*\Psi v(\Psi^* x_0)^T\right]. \end{aligned} \tag{8.131}$$

The above is written in terms of the Riemannian metric at (Φ,Ψ) as

$$\begin{aligned}\left\langle \frac{\partial}{\partial(\Phi,\Psi)} z_0(\Phi,\Psi)(X,Y),\ v\right\rangle &= \operatorname{Tr}\left[(\Psi^*\Psi)^{-1}Y^*\left(x_0-\Phi\Psi^* x_0\right)v^T\Psi^*\Psi\right] \\ &\quad - \operatorname{Tr}\left[(\Phi^*\Phi)^{-1}X^*\Psi v(\Psi^* x_0)^T\Phi^*\Phi\right] \\ &= \left\langle (X,Y),\ \left(-\Psi v(\Psi^* x_0)^T\Phi^*\Phi,\ \left(x_0-\Phi\Psi^* x_0\right)v^T\Psi^*\Psi\right)\right\rangle_{(\Phi,\Psi)}, \end{aligned} \tag{8.132}$$

which yields Eq. 8.30.

Finally, we compute the gradient of the regularization Eq. 8.13 by considering a perturbation $(X,Y) \in T_{(\Phi,\Psi)}\bar{\mathcal{M}}$ and writing the resulting perturbation of $\rho\circ\pi$ as

$$\begin{aligned}\mathrm{D}(\rho\circ\pi)(\Phi,\Psi)(X,Y) = \operatorname{Tr}\left\{(\Phi^*\Phi)^{-1}(\Phi^* X + X^*\Phi)\right\} + \operatorname{Tr}\left\{(\Psi^*\Psi)^{-1}(\Psi^* Y + Y^*\Psi)\right\} \\ - 2\operatorname{Tr}\left\{(\Psi^*\Phi)^{-1}(\Psi^* X + Y^*\Phi)\right\}. \end{aligned} \tag{8.133}$$

Applying permutation identities for the trace and collecting terms we have

$$\mathrm{D}(\rho\circ\pi)(\Phi,\Psi)(X,Y) = 2\,\mathrm{Tr}\left\{\left[(\Phi^*\Phi)^{-1}\Phi^* - (\Psi^*\Phi)^{-1}\Psi^*\right]X\right\} + 2\,\mathrm{Tr}\left\{Y^*\left[\Psi(\Psi^*\Psi)^{-1} - \Phi(\Psi^*\Phi)^{-1}\right]\right\}, \quad (8.134)$$

yielding

$$\begin{aligned}\mathrm{D}(\rho\circ\pi)(\Phi,\Psi)(X,Y) &= 2\,\mathrm{Tr}\left\{(\Phi^*\Phi)^{-1}\left[\Phi - \Psi(\Phi^*\Psi)^{-1}(\Phi^*\Phi)\right]^* X\right\} \\ &\quad + 2\,\mathrm{Tr}\left\{(\Psi^*\Psi)^{-1}\left[\Psi - \Phi(\Psi^*\Phi)^{-1}(\Psi^*\Psi)\right]^* Y\right\} \\ &= \left\langle\left(2\left[\Phi - \Psi(\Phi^*\Psi)^{-1}(\Phi^*\Phi)\right],\ 2\left[\Psi - \Phi(\Psi^*\Phi)^{-1}(\Psi^*\Psi)\right]\right),\ (X,Y)\right\rangle_{(\Phi,\Psi)}. \quad (8.135)\end{aligned}$$

Under the additional assumption that $\Psi^*\Phi = I_r$, we obtain Eq. 8.31. This completes the proof of Proposition 8.4.4. □

8.E Convergence Guarantees

Here we provide convergence guarantees for the algorithm presented in Section 8.5.3 applied to our optimal model reduction problem under modest conditions on the problem's setup. Leveraging the results proved by H. Sato in [235], we obtain general conditions for convergence that we state in Theorem 8.E.1 and specialize to useful classes of systems in Corollary 8.E.7, and Corollary 8.E.8. The algorithm converges in the sense of Eq. 8.138, which says that the gradient will eventually become arbitrarily small. Hence, an algorithm whose stopping condition is based on the gradient being sufficiently small will not run forever.

Theorem 8.E.1 (Convergence of Conjugate Gradient Algorithm)**.** *Suppose that the functions $x \mapsto g(x)$ and $(x,t) \mapsto f(x,u(t))$ describing the full-order model dynamics Eq. 8.1 along with their second-order partial derivatives with respect to x are continuous. Let the loss function L_y be twice continuously differentiable, take $\gamma > 0$, and let $\mathcal{D}$ be as in Proposition 8.3.4. We assume that $\mathcal{D}$ contains the initial point (V_0, W_0) and every $(V,W) \in \mathcal{P}$ such that $J(V,W) \le J(V_0,W_0)$. Then, at each $p_k = (V_k, W_k)$ starting from $p_0 = (V_0, W_0)$, there exists a step size $\alpha_k \ge 0$ so that $p_{k+1} = R_{p_k}(\alpha_k\eta_k)$ satisfies the Wolfe conditions along the search direction $\eta_k = (\xi_k, \omega_k) \in T_{p_k}\mathcal{P}$ computed by the scaled Riemannian Dai-Yuan conjugate gradient method. Consequently, the cost is non-increasing, i.e.,*

$$J(V_{k+1}, W_{k+1}) \le J(V_k, W_k) \qquad \forall k \ge 0. \quad (8.136)$$

Moreover, the step sizes α_k may always be chosen small enough such that

$$J(R_{p_k}(t\eta_k)) \leq J(V_0, W_0) \qquad \forall t \in [0, \alpha_k]. \tag{8.137}$$

Let us now assume that there is a subset $\mathcal{D}_c \subset \mathcal{D}$ such that $\mathcal{D}_c$ is closed in $\mathcal{M}$ and contains every $(V, W) \in \mathcal{P}$ for which $J(V, W) \leq J(V_0, W_0)$. If the step sizes are chosen so that $R_{p_k}(t\eta_k) \in \mathcal{D}_c$ for every $t \in [0, \alpha_k]$, which is always possible by Eq. 8.137, then the conjugate gradient algorithm converges in the sense that

$$\liminf_{k \to \infty} \| \nabla J(V_k, W_k) \|_{(V_k, W_k)} = 0. \tag{8.138}$$

Proof of Theorem 8.E.1. The openness of the set $\mathcal{D}$ in $\mathcal{P}$ on which the reduced-order model has a unique solution over the desired time interval was established in Proposition 8.3.4. Now we show that the cost function J given by Eq. 8.5 is twice continuously differentiable with respect to the subspaces (V, W) over the open subset $\mathcal{D} \subset \mathcal{P}$. The following Lemma 8.E.2 shows that the solution for the state of the reduced-order model Eq. 8.4 is twice continuously differentiable over $\mathcal{D}$. Combining this with the fact that g and L_y are twice continuously differentiable and that the regularization function ρ defined by Eq. 8.13 is infinitely many times continuously differentiable on $\mathcal{P}$, it follows that the cost function J is twice continuously differentiable over $\mathcal{D}$.

Lemma 8.E.2. *The solution $\hat{x}(t; (V, W))$ of the reduced-order model at any $t \in [t_0, t_{L-1}]$ is twice continuously differentiable with respect to (V, W) over the open subset $\mathcal{D} \subset \mathcal{P}$.*

Proof. Recall that by Theorem 8.3.3, the set of rank-r projection matrices $\mathbb{P}$ is smoothly diffeomorphic to the $2nr - 2r^2$ dimensional submanifold $\mathcal{P} \subset \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$. Let $\psi : \mathbb{R}^{2nr-2r^2} \to \mathcal{U} \subset \mathcal{D}$ be a local parameterization of a an open subset $\mathcal{U} \subset \mathcal{D}$. Letting $\phi : (V, W) \mapsto P_{V,W}$ be the diffeomorphism established by Theorem 8.3.3, the map $P = \phi \circ \psi$ is a smooth parameterization of the open subset $\phi(\mathcal{U}) \subset \mathbb{P}$.

Define the augmented state variable $w = (x, p) \in \mathbb{R}^n \times \mathbb{R}^{2nr-2r^2}$ whose dynamics are described by

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} w = F(w, t) := \begin{bmatrix} P(p) f(x, u(t)) \\ 0_{2nr-r^2} \end{bmatrix} \qquad w(0) = w_0. \tag{8.139}$$

Clearly, we have $w(t; w_0) = \hat{x}(t; \psi(p_0))$ when $w_0 = (P(p_0)x_0, p_0)$. It is also clear that F is twice continuously differentiable with respect to w, and so by Theorem 8.43 in [134], it follows that $w(t; w_0)$

is differentiable with respect to w_0. Furthermore, the Jacobian matrix $w^{(1)} = \frac{\partial w(t;w_0)}{\partial w_0}$ satisfies

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} \begin{bmatrix} w \\ w^{(1)} \end{bmatrix} = F^{(1)}(w, w^{(1)}, t) := \begin{bmatrix} F(w,t) \\ \frac{\partial}{\partial w} F(w,t) w^{(1)} \end{bmatrix}, \qquad w^{(1)}(0) = w_0^{(1)} = \begin{bmatrix} w_0 \\ I \end{bmatrix} \tag{8.140}$$

by the chain rule. Our smoothness assumptions on f now ensure that $F^{(1)}$ is continuously differentiable with respect to w and $w^{(1)}$. Applying Theorem 8.43 in [134] once more we find that $w^{(1)}(t; w_0^{(1)})$ is continuously differentiable with respect to $w_0^{(1)}$. Since w_0 is an element of $w_0^{(1)}$ it follows that $\frac{\partial w(t;w_0)}{\partial w_0}$ is continuosly differentiable with respect to w_0 and so $w(t; w_0)$ is twice continuously differentiable with respect to w_0.

Finally, since $w_0 = (P(p_0)x_0, p_0)$ is infinitely many times continuously differentiable with respect to p_0, the chain rule shows that $\hat{x}(t; \psi(p_0))$ is twice continuously differentiable with respect to p_0. Since ψ was an arbitrary smooth parameterization of an open subset of $\mathcal{D}$ it follows that $\hat{x}(t; (V, W))$ is twice continuously differentiable with respect to (V, W) over $\mathcal{D}$. □

Remark 8.E.3. *The argument used to prove Lemma 8.E.2 can be iterated as in [40] to prove d-times continuous differentiability of $\hat{x}(t; (V, W))$ with respect to (V, W) for any integer $d \geq 1$ as long as $(x, t) \mapsto f(x, u(t))$ has continuous partial derivatives with respect to x up to order d.*

We assumed that $p_0 = (V_0, W_0) \in \mathcal{D}$, so let us suppose that the current iterate $p_k = (V_k, W_k)$, $k \geq 1$, of the conjugate gradient algorithm also lies in $\mathcal{D}$ and satisfies $J(p_k) \leq J(p_{k-1}) \leq ... \leq J(p_0)$. When $\nabla J(p_k) = 0$ then we are already at a local extremum of the cost function and $\alpha_k = 0$ clearly satisfies the Wolfe conditions and yields $p_{k+1} = R_{p_k}(\alpha_k \eta_k) = p_k$ and $J(p_{k+1}) \leq J(p_k)$. On the other hand, Proposition 4.1 in [235] shows that when the gradient $\nabla J(p_k) \neq 0$ then the conjugate gradient-based search direction is a descent direction, that is

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}} J\left(R_{p_k}(t\eta_k)\right)\Big|_{t=0} = \langle \nabla J(p_k),\ \eta_k \rangle_{p_k} < 0. \tag{8.141}$$

By Lemma 8.3.1, the retraction $R_{p_k} : T_{p_k}\mathcal{M} \to \mathcal{M}$ defined by Eq. 8.37 is infinitely many times continuously differentiable. It follows that the line search function

$$J_k(t) = J\left(R_{p_k}(t\eta_k)\right) \tag{8.142}$$

is twice continuously differentiable over the open set $\mathcal{D}_k = \{t \in \mathbb{R} \ : \ R_{p_k}(t\eta_k) \in \mathcal{D}\}$. By our assumptions on $\mathcal{D}_c \subset \mathcal{D}$, it is clear that $\mathcal{D}_k$ contains the closed set $\{t \in \mathbb{R} \ : \ J_k(t) \leq J(V_0, W_0)\}$ and

so $\mathcal{D}_k$ contains $\{t \in \mathbb{R} \ : \ J_k(t) \leq J(V_k, W_k)\}$. To prove that there exists α_k satisfying the Wolfe conditions

$$J_k(\alpha_k) \leq J_k(0) + c_1 \alpha_k J_k'(0) \tag{8.143}$$

$$J_k'(\alpha_k) \geq c_2 J_k'(0), \tag{8.144}$$

with $0 < c_1 < c_2 < 1$, we follow the same argument as Lemma 3.1 in J. Nocedal and S. J. Wright [190]. First we observe that all $t \geq 0$ such that $J_k(t) \leq J_k(0) + c_1 t J_k'(0)$ automatically has $J_k(t) \leq J_k(0) \leq J(p_0)$ and so $t \in \mathcal{D}_k$. Since J, and hence J_k, is bounded below, but $\ell(t) := J_k(0) + c_1 t J_k'(0)$ is not, the graphs of $J_k(t)$ and $\ell(t)$ must intersect for some $t > 0$. Let $\alpha > 0$ be the smallest value such that $J_k(\alpha) = \ell(\alpha)$. Since $J_k'(0) < 0$ it follows that $J_k(t) < \ell(t)$ for every $t \in (0, \alpha) \subset \mathcal{D}_k$. By the mean value theorem, there exists $\alpha_k \in (0, \alpha)$ such that

$$J_k(\alpha) - J_k(0) = \alpha J_k'(\alpha_k). \tag{8.145}$$

It follows that

$$J_k'(\alpha_k) = \frac{J_k(\alpha) - J_k(0)}{\alpha} = \frac{\ell(\alpha) - J_k(0)}{\alpha} = c_1 J_k'(0) \geq c_2 J_k'(0), \tag{8.146}$$

and so α_k satisfies the Wolfe conditions. Moreover, we have

$$J(R_{p_k}(t\eta_k) \leq J(p_k) \qquad \forall t \in [0, \alpha_k], \tag{8.147}$$

and in particular $J(p_{k+1}) \leq J(p_k)$ where the next iterate is $p_{k+1} = R_{p_k}(\alpha_k \eta_k)$. Therefore we have proven that a Wolfe step satisfying Eq. 8.147, and hence Eq. 8.137 always exists and that the conjugate gradient algorithm produces a non-increasing sequence of costs $J(p_{k+1}) \leq J(p_k)$ for every $k \geq 0$.

We now turn our attention to showing that the conjugate gradient algorithm converges in the sense of Eq. 8.138. We shall do this by verifying the hypotheses of Theorem 4.2 in [235], which give sufficient conditions for the algorithm to converge in the sense of Eq. 8.138. In particular, if there is a Lipschitz constant L such that for every $p \in \mathcal{P}$ and $\xi \in T_p\mathcal{P}$ with unit magnitude $\|\xi\|_p = 1$ we have

$$|D(J \circ R_p)(t\xi)\xi - D(J \circ R_p)(0)\xi| \leq Lt, \quad \forall t > 0, \tag{8.148}$$

then Eq. 8.138 holds. In fact, following the argument in Appendix A of [236], the result of Theo-

rem 4.2 in [235], i.e., Eq. 8.138, still holds under the weaker condition that

$$|D(J \circ R_{p_k})(\alpha_k \eta_k)\eta_k - D(J \circ R_{p_k})(0)\eta_k| \leq L\alpha_k \|\eta_k\|_{p_k}^2, \tag{8.149}$$

for each of the iterates $k \geq 0$.

Here, we shall prove that there is a constant L such that

$$\boxed{\left|\frac{\mathrm{d}^2}{\mathrm{d}\,\mathrm{t}^2}(J \circ R_p)(t\xi)\right| \leq L, \qquad \forall t \geq 0 \quad \text{s.t.} \quad R_p(t\xi) \in \mathcal{D}_c} \tag{8.150}$$

for every $p \in \mathcal{P}$ and $\xi \in T_p\mathcal{P}$ with $\|\xi\|_p = 1$. Taking $p = p_k$ and $\xi = \eta_k / \|\eta_k\|_{p_k}$, the assumption $R_{p_k}(\tau\eta_k) \in \mathcal{D}_c$ for every $\tau \in [0, \alpha_k]$ ensures that $R_p(t\xi) \in \mathcal{D}_c$ for all t in the interval $0 \leq t \leq \alpha_k \|\eta_k\|_{p_k}$. Integrating Eq. 8.150 over this interval proves Eq. 8.149 and therefore Eq. 8.138. It now remains to verify Eq. 8.150.

As a closed subset of the compact manifold $\mathcal{M} = \mathcal{G}_{n,r} \times \mathcal{G}_{n,r}$, the set $\mathcal{D}_c$ is compact. Moreover, our assumptions guarantee that $\mathcal{D}_c$ contains the entire search path

$$\bigcup_{k=0}^{\infty} \{R_{p_k}(t\eta_k) \ : \ t \in [0, \alpha_k]\}. \tag{8.151}$$

Letting $\phi : \mathcal{P} \to \mathbb{P}$ be the C^∞ diffeomorphism established in Theorem 8.3.3, it is clear that $\phi(\mathcal{D}_c)$ is compact and contained in the relatively open set $\phi(\mathcal{D}) \subset \mathbb{P}$. In order to prove Eq. 8.150, it will be easier to lift the problem into $\mathbb{R}^{n \times n}$ by working with the equivalent cost function $J \circ \phi^{-1}$ on a neighborhood of $\phi(\mathcal{D})$ in $\mathbb{R}^{n \times n}$.

Since $\phi(\mathcal{D})$ is open in $\mathbb{P}$ and $\mathbb{P}$ is a smooth submanifold of $\mathbb{R}^{n \times n}$ by Theorem 8.3.3, it follows that $\phi(\mathcal{D})$ is a smooth submanifold of $\mathbb{R}^{n \times n}$. The following Lemma 8.E.4 shows that the restriction $J \circ \phi^{-1}\big|_{\phi(\mathcal{D})}$ of the lifted cost function to $\phi(\mathcal{D})$ may be extended to a twice continuously differentiable function $\tilde{J}$ defined on an open neighborhood $\mathcal{U}$ of $\phi(\mathcal{D})$ in $\mathbb{R}^{n \times n}$.

Lemma 8.E.4. *Let $\mathcal{N}$ be a smooth d-dimensional submanifold of $\mathbb{R}^k$ and let $h : \mathcal{N} \to \mathbb{R}$ be a C^r function. Then h may be extended to a C^r function $\tilde{h}$ defined on an open neighborhood of $\mathcal{N}$ in $\mathbb{R}^k$.*

Proof. Let $\imath : \mathcal{N} \to \mathbb{R}^k$ be the injection of $\mathcal{N}$ into $\mathbb{R}^k$. By the local immersion theorem [106] there is an open neighborhood $\mathcal{U}_p \subset \mathbb{R}^k$ of each point $p \in \mathcal{N}$ and a C^∞ diffeomorphism $\psi_p : \mathbb{R}^k \mapsto \mathcal{U}_p$ such that $(x_1, \ldots, x_d) \mapsto \psi(x_1, \ldots, x_d, 0, \ldots, 0)$ is a C^∞ parameterization of $\mathcal{N} \cap \mathcal{U}_p$. Let $\Pi : \mathbb{R}^k \to \mathbb{R}^k$ denote the projection onto the leading d-dimensional coordinate subspace $\Pi(x_1, \ldots, x_k) =$

$(x_1, \ldots, x_d, 0, \ldots, 0)$ and let $\tilde{h}_p : \mathcal{U}_p \to \mathbb{R}$ be given by

$$\tilde{h}_p = h \circ \psi_p \circ \Pi \circ \psi_p^{-1}. \tag{8.152}$$

Then $\tilde{h}_p$ is a C^r extension of h to the neighborhood $\mathcal{U}_p$ since $\psi_p \circ \Pi \circ \psi_p^{-1}$ is the identity on $\mathcal{N} \cap \mathcal{U}_p$.

Covering $\mathcal{N}$ by a union $\mathcal{U} = \bigcup_{p \in \mathcal{N}} \mathcal{U}_p$ of such neighborhoods, we may construct a smooth partition of unity $\{\theta_i\}_{i=1}^{\infty}$ with the following properties [106]:

1. $0 \leq \theta_i(p) \leq 1$ for each $p \in \mathcal{U}$,
2. each $p \in \mathcal{U}$ has a neighborhood on which all but finitely many θ_i are identically zero,
3. the support of each θ_i is contained in a closed subset of some $\mathcal{U}_{p_i}$,
4. and $\sum_{i=1}^{\infty} \theta_i(p) = 1$ for every $p \in \mathcal{U}$.

The final C^r extension of h to the neighborhood $\mathcal{U}$ is constructed by letting

$$\tilde{h} = \sum_{i=1}^{\infty} \theta_i \tilde{h}_{p_i}. \tag{8.153}$$

Since each θ_i is C^∞, each $\tilde{h}_{p_i}$ is C^r, and only finitely many θ_i are nonzero on a neighborhood of any $p \in \mathcal{U}$, it is easy to see that $\tilde{h}$ is C^r on $\mathcal{U}$. Finally, $\tilde{h}_{p_i}(p) = h(p)$ when $p \in \mathcal{U}_{p_i}$ implies that for any $p \in \mathcal{N}$ we have

$$\tilde{h}(p) = \sum_{i=1}^{\infty} \theta_i(p) \tilde{h}_{p_i}(p) = \sum_{i \,:\, p \in \mathcal{U}_{p_i}} \theta_i(p) \tilde{h}_{p_i}(p) = \left(\sum_{i=1}^{\infty} \theta_i(p) \right) h(p) = h(p), \tag{8.154}$$

and so $\tilde{h}$ agrees with h on $\mathcal{N}$. □

The function $J \circ R$ may now be written in terms of $\tilde{J}$ according to

$$\begin{aligned} J \circ R_{(V,W)}(\xi, \omega) &= J \circ \pi \circ \bar{R}_{V,W}(\bar{\xi}_\Phi, \bar{\omega}_\Psi) \\ &= J \circ \phi^{-1} \circ \phi \circ \pi(\Phi + \bar{\xi}_\Phi, \Psi + \bar{\omega}_\Psi) \\ &= \tilde{J}\left((\Phi + \bar{\xi}_\Phi) \left[(\Psi + \bar{\omega}_\Psi)^* (\Phi + \bar{\xi}_\Phi) \right]^{-1} (\Psi + \bar{\omega}_\Psi)^* \right), \end{aligned} \tag{8.155}$$

when $(\Phi, \Psi) \in \pi^{-1}(V, W)$ are representatives. Letting

$$P(t) := (\Phi + t\bar{\xi}_\Phi) \left[(\Psi + t\bar{\omega}_\Psi)^* (\Phi + t\bar{\xi}_\Phi) \right]^{-1} (\Psi + t\bar{\omega}_\Psi)^*, \tag{8.156}$$

for any projection subspaces $(V, W) \in \mathcal{P}$, tangent vector $(\xi, \omega) \in T_{(V,W)}\mathcal{P}$ with unit magnitude, and representatives $(\Phi, \Psi) \in \pi^{-1}(V, W)$, then Eq. 8.150 is equivalent to

$$\boxed{\left| \frac{\mathrm{d}^2}{\mathrm{d}\,t^2} \tilde{J}(P(t)) \right| \leq L \qquad \forall t \geq 0 \quad \text{s.t.} \quad P(t) \in \phi(\mathcal{D}_c).} \tag{8.157}$$

Without loss of generality, we shall assume that the chosen representatives are orthonormal, i.e., $\Phi^*\Phi = \Psi^*\Psi = I_r$. With this choice, the assumption that $(\xi, \omega) \in T_{(V,W)}\mathcal{P}$ has unit magnitude $\|(\xi, \omega)\|_{(V,W)} = 1$ implies that the Hilbert-Schmidt norms of the horizontal lifts satisfy $\|\bar{\xi}_\Phi\|_{HS}^2 + \|\bar{\omega}_\Psi\|_{HS}^2 = 1$.

Remark 8.E.5 (Hilbert-Schmidt Norm and Trace)**.** *Since we deal with operators between (finite-dimensional) Hilbert spaces with possibly different inner products, it will be convenient to use Hilbert-Schmidt norms. For instance $\bar{\xi}_\Phi$ is an operator from $\mathbb{R}^r$ into the state space $\mathcal{X}$, which has its own norm that may differ from the usual norm on $\mathbb{R}^n$. Since all the operators we deal with here are finite-dimensional, they are automatically Hilbert-Schmidt operators. If $A : \mathcal{H}_1 \to \mathcal{H}_2$ is a Hilbert-Schmidt operator between two Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, then the Hilbert-Schmidt norm is defined using a generalized notion of trace. In particular, if $\{e_i\}$ is an orthonormal basis for $\mathcal{H}_1$ then*

$$\|A\|_{HS}^2 = \sum_i \|Ae_i\|_{\mathcal{H}_2}^2 = \sum_i \langle e_i, A^*Ae_i \rangle =: \mathrm{Tr}\,(A^*A), \tag{8.158}$$

and this definition is independent of the choice of bases. If $B : \mathcal{H}_2 \to \mathcal{H}_3$ is another Hilbert-Schmidt operator then we have

$$\|AB\|_{HS} \leq \|A\|_{HS}\|B\|_{HS}. \tag{8.159}$$

For more details, see Problem 40 in H. Brezis [30] and Section 6.6 in M. Reed and B. Simon [218]. If $\mathcal{H}_1 = \mathcal{H}_3$ and $\mathcal{H}_2$ are finite-dimensional, then the trace satisfies the usual adjoint and permutation identities, namely

$$\mathrm{Tr}(BA) = \mathrm{Tr}(AB) = \mathrm{Tr}(B^*A^*) = \mathrm{Tr}(A^*B^*). \tag{8.160}$$

When $\mathcal{H}_1 = \mathbb{R}^p$ and $\mathcal{H}_2 = \mathbb{R}^q$ then the trace defined above is the usual trace and the Hilbert-Schmidt norm is the Frobenius norm.

Differentiating $\tilde{J}(P(t))$, and denoting derivates of $P(t)$ with respect to t by $P'(t)$, $P''(t)$, etc., we

shall bound each term of

$$\frac{\mathrm{d}^2}{\mathrm{d}\,t^2}\,\tilde{J}(P(t)) = \mathrm{D}^2\,\tilde{J}(P(t))[P'(t), P'(t)] + \mathrm{D}\,\tilde{J}(P(t))P''(t), \tag{8.161}$$

in order to prove Eq. 8.157. Since $P(t)$ is assumed to remain in the compact set $\phi(\mathcal{D}_c)$ and the derivative $\mathrm{D}\,\tilde{J}(P(t))$ and Hessian $\mathrm{D}^2\,\tilde{J}(P(t))$ are continuous on $\mathcal{U} \supset \phi(\mathcal{D}_c)$, it follows immediately that $\mathrm{D}\,\tilde{J}(P(t))$ and $\mathrm{D}^2\,\tilde{J}(P(t))$ are bounded. Obviously $P(t)$ is also bounded since it lies in the compact set $\phi(\mathcal{D}_c)$.

It remains to show that $P'(t)$ and $P''(t)$ are bounded by brute-force differentiation. To simplify our notation, let us define

$$\begin{aligned} G_\Phi(t) &= (\Phi + t\bar{\xi}_\Phi)^*(\Phi + t\bar{\xi}_\Phi), \\ G_\Psi(t) &= (\Psi + t\bar{\omega}_\Psi)^*(\Psi + t\bar{\omega}_\Psi), \\ G_{\Psi,\Phi}(t) &= (\Psi + t\bar{\omega}_\Psi)^*(\Phi + t\bar{\xi}_\Phi), \end{aligned} \tag{8.162}$$

and observe that since we chose Φ and Ψ to be orthonormal representatives, we have $G_\Phi(0) = G_\Psi(0) = I_r$. The following Lemma 8.E.6 shows that several of the terms that will appear in the expressions for $P'(t)$ and $P''(t)$ are uniformly bounded.

Lemma 8.E.6. *The following bounds hold uniformly over all $(V, W) \in \mathcal{P}$, all orthonormal representatives $(\Phi, \Psi) \in \pi^{-1}(V, W)$, all unit magnitude $(\xi, \omega) \in T_{(V,W)}\mathcal{P}$, and all $t \geq 0$ such that $P(t) \in \phi(\mathcal{D}_c)$:*

$$\|G_\Phi(t)^{-1}\|_{HS} \leq \sqrt{r} \quad \textit{and} \quad \|G_\Psi(t)^{-1}\|_{HS} \leq \sqrt{r} \tag{8.163}$$

$$\|(\Phi + t\bar{\xi}_\Phi)G_\Phi(t)^{-1}\|_{HS} \leq \sqrt{r} \quad \textit{and} \quad \|(\Psi + t\bar{\omega}_\Psi)G_\Psi(t)^{-1}\|_{HS} \leq \sqrt{r} \tag{8.164}$$

$$\|(\Phi + t\bar{\xi}_\Phi)G_\Phi(t)^{-1}(\Phi + t\bar{\xi}_\Phi)^*\|_{HS} = \|(\Psi + t\bar{\omega}_\Psi)G_\Psi(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*\|_{HS} = \sqrt{r} \tag{8.165}$$

$$\|G_{\Psi,\Phi}(t)^{-1}\|_{HS} \leq r\|P(t)\|_{HS} \tag{8.166}$$

$$\begin{aligned} \|(\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}\|_{HS} &\leq \sqrt{r}\|P(t)\|_{HS} \quad \textit{and} \\ \|G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*\|_{HS} &\leq \sqrt{r}\|P(t)\|_{HS}. \end{aligned} \tag{8.167}$$

Proof. Since $\bar{\xi}_\Phi$ is in the horizontal subspace at Φ, we have $\Phi^*\bar{\xi}_\Phi = 0$ and so $G_\Phi(t) = \Phi^*\Phi + t^2\bar{\xi}_\Phi^*\bar{\xi}_\Phi$ is positive-definite. The same holds for $G_\Psi(t) = \Psi^*\Psi + t^2\bar{\omega}_\Psi^*\bar{\omega}_\Psi$. Consequently, the eigenvalues $\lambda_i(t)$

of $G_\Phi(t)$ are positive and non-decreasing with t, so

$$\|G_\Psi(t)^{-1}\|_{HS}^2 = \sum_{i=1}^{r} \frac{1}{\lambda_i(t)^2} \leq \sum_{i=1}^{r} \frac{1}{\lambda_i(0)^2} = \|G_\Psi(0)^{-1}\|_{HS}^2 = r. \tag{8.168}$$

Similarly, we have $\|G_\Psi(t)^{-1}\|_{HS}^2 \leq r$.

By direct calculation, we have

$$\begin{aligned} \|(\Phi + t\bar{\xi}_\Phi)G_\Phi(t)^{-1}\|_{HS}^2 &= \mathrm{Tr}\left[G_\Phi(t)^{-1}(\Phi + t\bar{\xi}_\Phi)^*(\Phi + t\bar{\xi}_\Phi)G_\Phi(t)^{-1}\right] \\ &= \mathrm{Tr}\left[G_\Phi(t)^{-1}\right] = \sum_{i=1}^{r} \frac{1}{\lambda_i(t)} \leq \sum_{i=1}^{r} \frac{1}{\lambda_i(0)} = r. \end{aligned} \tag{8.169}$$

Similarly, we have $\|(\Psi + t\bar{\omega}_\Psi)G_\Psi(t)^{-1}\|_{HS}^2 \leq r$.

By the permutation identity for the trace, we have

$$\begin{aligned} \|(\Phi + t\bar{\xi}_\Phi)G_\Phi(t)^{-1}(\Phi + t\bar{\xi}_\Phi)^*\|_{HS}^2 &= \mathrm{Tr}\left[(\Phi + t\bar{\xi}_\Phi)G_\Phi(t)^{-1}(\Phi + t\bar{\xi}_\Phi)^*\right] \\ &= \mathrm{Tr}\left[G_\Phi(t)G_\Phi(t)^{-1}\right] = \mathrm{Tr}(I_r) = r. \end{aligned} \tag{8.170}$$

Similarly, we have $\|(\Psi + t\bar{\omega}_\Psi)G_\Psi(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*\|_{HS}^2 = r$.

To bound $G_{\Psi,\Phi}(t)^{-1}$, we observe that

$$\begin{aligned} G_{\Psi,\Phi}(t)^{-1} &= G_\Phi(t)^{-1}G_\Phi(t)G_{\Psi,\Phi}(t)^{-1}G_\Psi(t)G_\Psi(t)^{-1} \\ &= G_\Phi(t)^{-1}(\Phi + t\bar{\xi}_\Phi)^*P(t)(\Psi + t\bar{\omega}_\Psi)G_\Psi(t)^{-1}. \end{aligned} \tag{8.171}$$

By the above arguments, we have

$$\begin{aligned} \|G_{\Psi,\Phi}(t)^{-1}\|_{HS} &\leq \|G_\Phi(t)^{-1}(\Phi + t\bar{\xi}_\Phi)^*\|_{HS}\|P(t)\|_{HS}\|(\Psi + t\bar{\omega}_\Psi)G_\Psi(t)^{-1}\|_{HS} \\ &\leq r\|P(t)\|_{HS}. \end{aligned} \tag{8.172}$$

Since $P(t) \in \phi(\mathcal{D}_c)$ and $\phi(\mathcal{D}_c)$ is a compact subset of $\mathbb{R}^{n\times n}$, it follows that $\|P(t)\|_{HS}$ is bounded and so $\|G_{\Psi,\Phi}(t)^{-1}\|_{HS}$ is bounded as a consequence.

Using a similar argument, we observe that

$$G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^* = G_\Phi(t)^{-1}(\Phi + t\bar{\xi}_\Phi)^*P(t). \tag{8.173}$$

By the bound for $G_\Phi(t)^{-1}(\Phi + t\bar{\xi}_\Phi)^*$ proved previously, we conclude that

$$\|G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*\|_{HS} \leq \|G_\Phi(t)^{-1}(\Phi + t\bar{\xi}_\Phi)^*\|_{HS}\|P(t)\|_{HS} \leq \sqrt{r}\|P(t)\|_{HS} \tag{8.174}$$

is bounded. Similarly, we have $\|(\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}\|_{HS} \leq \sqrt{r}\|P(t)\|_{HS}$. □

We take another swig of coffee and differentiate:

$$\begin{aligned} P'(t) = \bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^* + (\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^* \\ - (\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^* \underbrace{(\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*}_{P(t)} \\ - \underbrace{(\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*}_{P(t)}\bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*. \end{aligned} \tag{8.175}$$

By Lemma 8.E.6 and the fact that $\|\bar{\xi}_\Phi\|_{HS}^2 + \|\bar{\omega}_\Psi\|_{HS}^2 = 1$, every term in the above expression for $P'(t)$ is uniformly bounded and so $P'(t)$ is uniformly bounded.

Let us now write $P'(t)$ as a sum of four terms

$$\begin{aligned} P'(t) = \underbrace{\bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*}_{A(t)} + \underbrace{(\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^*}_{B(t)} \\ - \underbrace{(\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^* P(t)}_{C(t)} - \underbrace{P(t)\bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*}_{D(t)}, \end{aligned} \tag{8.176}$$

and show that the derivative of each term is bounded. Fortunately, $B(t)$ and $D(t)$ are what we get when we swap Φ and Ψ in $A(t)^*$ and $C(t)^*$ respectively. We shall only show that $A'(t)$ and $C'(t)$ are uniformly bounded since the corresponding arguments for $B'(t)$ and $D'(t)$ are simply obtained by swapping Φ and Ψ line for line.

$$\begin{aligned} A'(t) = -\bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^* \underbrace{(\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*}_{P(t)} \\ - \bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*\bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^* + \bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^* \end{aligned} \tag{8.177}$$

Again by Lemma 8.E.6 and the fact that $\|\bar{\xi}_\Phi\|_{HS}^2 + \|\bar{\omega}_\Psi\|_{HS}^2 = 1$, every term in the above expression for $A'(t)$ is uniformly bounded and so $A'(t)$ is uniformly bounded.

We now differentiate $C(t)$ and obtain

$$\begin{aligned} C'(t) = \bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^* P(t) + (\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^* P'(t) \\ - (\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^*(\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^* P(t) \\ - \underbrace{(\Phi + t\bar{\xi}_\Phi)G_{\Psi,\Phi}(t)^{-1}(\Psi + t\bar{\omega}_\Psi)^*}_{P(t)}\bar{\xi}_\Phi G_{\Psi,\Phi}(t)^{-1}\bar{\omega}_\Psi^* P(t). \end{aligned} \quad (8.178)$$

Finally, by Lemma 8.E.6, the boundedness of $P'(t)$ proved above, and the fact that $\|\bar{\xi}_\Phi\|_{HS}^2 + \|\bar{\omega}_\Psi\|_{HS}^2 = 1$, every term in the above expression for $C'(t)$ is uniformly bounded and so $C'(t)$ is uniformly bounded. Repeating the symmetric arguments for $B'(t)$ and $D'(t)$ we finally conclude that $P''(t)$ is uniformly bounded. We have now established that every term in Eq. 8.161 is uniformly bounded and so Eq. 8.157 holds for some fixed L, completing the proof of Theorem 8.E.1. □

While the conditions in Theorem 8.E.1 appear abstract, they actually encompass two very important special cases. In Corollary 8.E.7, below, we show that when the full-order model has bounded first derivatives with respect to the state variables, then the conjugate gradient algorithm always converges. For instance, Corollary 8.E.7 implies that the algorithm always converges when the governing equations are linear.

Corollary 8.E.7. *Suppose that f, g, and L_y are as in Theorem 8.E.1 and $\gamma > 0$ and suppose that $\frac{\partial}{\partial x} f(x, u(t))$ is bounded. If there is a finite constant $C \geq \rho(V_0, W_0)/\gamma$ so that the step sizes α_k satisfy the Wolfe conditions and $\rho(R_{p_k}(t\eta_k)) \leq C$ for every $t \in [0, \alpha_k]$, then the conjugate gradient algorithm converges in the sense of Eq. 8.138.*

Proof. We need only verify the assumptions in Theorem 8.E.1: namely, that there is a subset $\mathcal{D}_c \subset \mathcal{D}$ that is closed in $\mathcal{M}$ and contains every $(V, W) \in \mathcal{P}$ for which $J(V, W) \leq J(V_0, W_0)$. By Proposition 8.3.4, our assumption that the Jocobian of the full-order model is bounded implies that $\mathcal{D} = \mathcal{P}$. It is also clear that $\mathcal{D}_c = \{(V, W) \in \mathcal{P} \,:\, \rho(V, W) \leq C\}$ is closed in $\mathcal{M}$. To see this, suppose that $\{p_k\}_{k=1}^\infty \subset \mathcal{D}_c$ is a sequence such that $p_k \to p \in \mathcal{M}$. Then Theorem 8.3.5 implies that $p \in \mathcal{P}$, for if not then $\rho(p_k) \to \infty$, which contradicts $\rho(p_k) \leq C$. Since ρ is continuous on $\mathcal{P}$, it follows that $\rho(p) = \lim_{k\to\infty} \rho(p_k) \leq C$ and so $p \in \mathcal{D}_c$.

Finally, if $C \geq \rho(V_0, W_0)/\gamma$, then any $(V, W) \in \mathcal{P}$ with $J(V, W) \leq J(V_0, W_0)$ must have

$$\rho(V, W) \leq \frac{1}{\gamma} J(V, W) \leq \frac{1}{\gamma} J(V_0, W_0) \leq C \quad (8.179)$$

and so $(V, W) \in \mathcal{D}_c$.

Furthermore, the property that $R_{p_k}(t\eta_k) \in \mathcal{D}_c$ for every $t \in [0, \alpha_k]$ is satisfied automatically by our assumption, so Theorem 8.E.1 implies that (8.138) holds, and the algorithm converges. □

On the other hand, many important systems, such as the discretized Navier-Stokes equations have quadratic nonlinearities resulting in unbounded first derivatives with respect to the state variables. In such cases, a poorly chosen projection-based reduced-order model Eq. 8.4 may have states that blow up in finite time. As long as Assumption 8.3.6 holds (i.e., any finite-time blow-up also results in an unbounded cost J), then the following Corollary 8.E.8 shows that the conjugate gradient algorithm will converge.

Corollary 8.E.8. *Suppose that f, g, and L_y are as in Theorem 8.E.1 and $\gamma > 0$. If there is a finite constant $C \geq J(V_0, W_0)$ so that the step sizes satisfy the Wolfe conditions and $J(R_{p_k}(t\eta_k)) \leq C$ for every $t \in [0, \alpha_k]$, then the conjugate gradient algorithm converges in the sense of Eq. 8.138.*

Proof. As we have seen in the proof of Proposition 8.3.4, the smoothness of f ensures that if a solution of the reduced-order model Eq. 8.4 exists, it must be unique. Moreover, if a solution of the reduced-order model does not exist over $[t_0, t_{L-1}]$, then the solution $\hat{x}(t; (V, W))$ can be defined over some maximal interval $[t_0, \omega)$ with $t_0 < \omega < t_{L-1}$ and $\|\hat{x}(t; (V, W))\| \to \infty$ as $t \to \omega^-$. Consequently, if the reduced-order model does not have a solution for $(V, W) \in \mathcal{P}$ over $[t_0, t_{L-1}]$ then $J(V, W) = \infty$. Therefore, the set $\mathcal{D}$ can be identified with the set over which the cost function is finite, i.e.,

$$\mathcal{D} = \{(V, W) \in \mathcal{P} \ : \ J(V, W) < \infty\}. \tag{8.180}$$

We shall show that the set

$$\mathcal{D}_c = \{(V, W) \in \mathcal{P} \ : \ J(V, W) \leq C\} \tag{8.181}$$

is closed in $\mathcal{M}$, which will complete the proof since Eq. 8.180 implies that $\mathcal{D}_c \subset \mathcal{D}$ and $C > J(V_0, W_0)$. Suppose that $\{(\tilde{V}_k, \tilde{W}_k)\}_{k=1}^{\infty} \subset \mathcal{D}_c$ is a sequence such that $(\tilde{V}_k, \tilde{W}_k) \to (\tilde{V}, \tilde{W}) \in \mathcal{M}$. Then it is clear that $(\tilde{V}, \tilde{W}) \in \mathcal{P}$, for if it were not then Theorem 8.3.5 would give $\rho(\tilde{V}, \tilde{W}) \to \infty$ and so $J(\tilde{V}, \tilde{W}) \to \infty$. Moreover, if $(\tilde{V}, \tilde{W}) \in \mathcal{P} \setminus \mathcal{D}$ then by Proposition 8.3.4 we would have

$$\max_{t \in [t_0, t_{L-1}]} \|\hat{x}(t; (\tilde{V}_k, \tilde{W}_k))\| \to \infty \quad \text{as} \quad k \to \infty, \tag{8.182}$$

which implies that $J(\tilde{V}_k, \tilde{W}_k) \to \infty$ by Assumption 8.3.6. This contradicts the assumption that $J(\tilde{V}_k, \tilde{W}_k) \leq C$ for every k. Therefore, the limit point $(\tilde{V}, \tilde{W}) \in \mathcal{D}$. In Proposition 8.3.4 we showed

that J is differentiable, and hence continuous on $\mathcal{D}$ and so

$$J(\tilde{V}, \tilde{W}) = \lim_{k\to\infty} J(\tilde{V}_k, \tilde{W}_k) \leq C. \tag{8.183}$$

Therefore, $(\tilde{V}, \tilde{W}) \in \mathcal{D}_c$ and we conclude that $\mathcal{D}_c$ is closed in $\mathcal{M}$. Applying Theorem 8.E.1 completes the proof. □

8.F Auxiliary Proofs and Results

Proof of Proposition 8.2.2 (Subspaces Defining Oblique Projections). Suppose that $w \in W$ is nonzero and $w \perp V$. Since $\operatorname{Range}\Psi = W$ and $\operatorname{Range}\Phi = V$, there exists $z \in \mathbb{R}^r$ such that $w = \Psi z$ and

$$0 = \langle \Psi z,\ \Phi x\rangle = z^T \Psi^* \Phi x \qquad \forall x \in \mathbb{R}^r. \tag{8.184}$$

It follows that $z \perp \operatorname{Range}(\Psi^*\Phi)$ and so $\det(\Psi^*\Phi) = 0$. Similarly, suppose that $v \in V$ and $v \perp W$. Then there exists $z \in \mathbb{R}^r$ such that $v = \Phi z$ and $x^T\Psi^*\Phi z = 0$ for every $x \in \mathbb{R}^r$. Hence $z \in \operatorname{Null}(\Psi^*\Phi)$ and so $\det(\Psi^*\Phi) = 0$.

On the other hand, if $\det(\Psi^*\Phi) = 0$ then there exists nonzero $n \in \operatorname{Null}(\Psi^*\Phi)$ and nonzero $z \in \operatorname{Range}(\Psi^*\Phi)^\perp$. It follows that $\Phi n \in V$ and $\Psi z \in W$ are nonzero vectors satisfying $\Phi n \perp W$ and $\Psi z \perp V$ because for every $x \in \mathbb{R}^r$ we have $0 = x^T\Psi^*\Phi n = \langle \Psi x,\ \Phi n\rangle$ and $0 = z^T\Psi^*\Phi x = \langle \Psi z,\ \Phi x\rangle$. This proves that the first three statements are equivalent.

Suppose that the first three statement hold, then $\hat{x} = \Phi(\Psi^*\Phi)^{-1}\Psi^* x$ satisfies

$$\langle w,\ \hat{x}\rangle = \left\langle \Psi(\Phi^*\Psi)^{-1}\Phi^* w,\ x\right\rangle = \langle w,\ x\rangle \qquad \forall w \in W \tag{8.185}$$

because any $w \in W$ can be written as $w = \Psi z$ for some $z \in \mathbb{R}^r$ and $\Psi(\Phi^*\Psi)^{-1}\Phi^* w = \Psi(\Phi^*\Psi)^{-1}\Phi^*\Psi z = \Psi z = w$. Moreover, if there is another $\hat{x}'$ satisfying the desired condition then $\hat{x} - \hat{x}' \in V$ is orthogonal to W; hence $\hat{x} = \hat{x}'$ by the first statement.

Finally suppose that there exists a nonzero element $w \in W$ that is orthogonal to V. For this w, we have $\langle w,\ w\rangle \neq 0$ and $\langle w,\ \hat{x}\rangle = 0$ for every $\hat{x} \in V$, contradicting the fourth statement when $x = w$. Even if we have an $x \in \mathbb{R}^n$ for which there exists $\hat{x} \in V$ satisfying $\langle w,\ x\rangle = \langle w,\ \hat{x}\rangle$ for every $w \in W$, there is a nonzero $v \in V$ that is orthogonal to W, any multiple of which can be added to $\hat{x}$, contradicting uniqueness of $\hat{x}$. □

Proposition 8.F.1 (Bound on Projection Operators)**.** *The operator norm of $P_{V,W}$ for $(V,W) \in \mathcal{P}$ is bounded by the regularization function $\rho(V,W)$ according to*

$$\|P_{V,W}\|_{op} = \sup_{v \in \mathcal{X} \,:\, \|v\|_{\mathcal{X}} \leq 1} \|P_{V,W} v\|_{\mathcal{X}} \leq e^{\rho(V,W)/2}. \tag{8.186}$$

Proof. Let $(\Phi, \Psi) \in \pi^{-1}(V,W)$ be orthonormal representatives of $(V,W) \in \mathcal{P}$, i.e., $\Phi^*\Phi = \Psi^*\Psi = I_r$. By the interlacing properties of singular values [214, 261], it follows that the lth singular values of $\Psi^*\Phi$ and Φ satisfy $\sigma_l(\Psi^*\Phi) \leq \sigma_l(\Phi) = 1$, for each $1 \leq l \leq r$. Therefore, by Eq. 8.8 and Eq. 8.13 we have

$$\begin{aligned} \|P_{V,W}\|_{\mathrm{op}} \leq \|\Phi\|_{\mathrm{op}} \|(\Psi^*\Phi)^{-1}\|_{\mathrm{op}} \|\Psi^*\|_{\mathrm{op}} = \|(\Psi^*\Phi)^{-1}\|_{\mathrm{op}} &= \frac{1}{\sigma_r(\Psi^*\Phi)} \\ &\leq \frac{1}{\prod_{i=1}^{r} \sigma_i(\Psi^*\Phi)} = \frac{1}{|\det(\Psi^*\Phi)|} = e^{\rho(V,W)/2}. \end{aligned} \tag{8.187}$$

□

8.G The role of pressure in incompressible flows

As mentioned in the body of the paper in section 8.7, the pressure may be removed entirely from the Navier-Stokes formulation (8.49)–(8.51).

Let us write equations (8.49)–(8.51) in compact form as

$$\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} F & -G \\ D & 0 \end{bmatrix} \begin{bmatrix} q \\ p \end{bmatrix} + \begin{bmatrix} g(q) \\ 0 \end{bmatrix}, \tag{8.188}$$

where $q = (u, v)$, D is the divergence operator, G is the gradient operator, F contains the vector Laplacian and $g(q)$ contains the nonlinear terms in the momentum equations (8.49) and (8.50). Taking the divergence of the first row of (8.188), and using $Dq = 0$ (by the second row of (8.188)) along with $DFq = FDq = 0$, we obtain a Poisson equation

$$\underbrace{DG}_{\widetilde{L}} p = Dg(q), \tag{8.189}$$

where $\widetilde{L}$ is the scalar Laplacian operator. Often, instead of prescribing pressure boundary conditions at the physical boundaries of the spatial domain, a unique solution to (8.189) is instead computed by fixing the pressure and the pressure gradient at some location (r_0, z_0) in physical space [206].

That is, in cylindrical coordinates

$$p = \frac{\partial p}{\partial z} = \frac{\partial p}{\partial r} = 0 \quad \text{at } (r_0, z_0) \in \Omega, \tag{8.190}$$

where (r_0, z_0) may be chosen arbitrarily. This approach is particularly convenient in numerical methods based on the finite volume or finite difference discretization of the spatial dimensions. The pressure may thus be written as

$$p = \widetilde{L}^{-1} D g(q), \tag{8.191}$$

and consequently (8.188) may be reduced to

$$\frac{\partial}{\partial t} q = Fq + g(q) - G\widetilde{L}^{-1} D g(q) = f(q), \tag{8.192}$$

which is in the form of (8.1). In practice, in order to compute the action of f on some vector field q, we proceed as follows:

1. compute $\varphi(q) = Fq + g(q)$,
2. compute the pressure by solving $\widetilde{L}p = D\varphi(q)$, and finally
3. compute $f(q) = \varphi(q) - Gp$.

8.H Derivation of the adjoint of the Navier-Stokes equation

In this appendix we derive the adjoint of the Navier-Stokes equation linearized about a steady solution $Q = (U, V)$, which satisfies the boundary conditions described in section 8.7. We will work in cylindrical coordinates, and the adjoint equation will be derived with respect to the inner product

$$\langle f, g \rangle = \int_\Omega f(r, z) g(r, z)\, r \, \mathrm{d}r \, \mathrm{d}z, \tag{8.193}$$

where $\Omega = \{(r, z) |\ r \in [0, L_r],\ z \in [0, L_z]\}$ is the spatial domain.

We let $q = (u, v)$ be the two-dimensional velocity with axial component u and radial component v, and we let p be the pressure, as in section 8.7. For a given Reynolds number Re, the linearized

Navier-Stokes equation and the continuity equation then read,

$$\frac{\partial u}{\partial t} = -u\frac{\partial U}{\partial z} - v\frac{\partial U}{\partial r} - U\frac{\partial u}{\partial z} - V\frac{\partial u}{\partial r} - \frac{\partial p}{\partial z} + \frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{\partial^2 u}{\partial z^2}\right) \tag{8.194}$$

$$\frac{\partial v}{\partial t} = -u\frac{\partial V}{\partial z} - v\frac{\partial V}{\partial r} - U\frac{\partial v}{\partial z} - V\frac{\partial v}{\partial r} - \frac{\partial p}{\partial r} + \frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) - \frac{v}{r^2} + \frac{\partial^2 v}{\partial z^2}\right) \tag{8.195}$$

$$\frac{\partial u}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}(rv) = 0, \tag{8.196}$$

with velocity boundary conditions

$$\frac{\partial u}{\partial r} = v = 0 \quad \text{at } r = 0 \tag{8.197}$$

$$u = v = 0 \quad \text{at } r = L_r,\ z = 0 \tag{8.198}$$

$$\frac{\partial u}{\partial z} = \frac{\partial v}{\partial z} = 0 \quad \text{at } z = L_z. \tag{8.199}$$

As discussed in appendix Appendix 8.G, in order to uniquely determine the pressure field, it suffices to fix the pressure and the pressure gradients at some location (r_0, z_0) in physical space, rather than specifying pressure boundary conditions at the boundaries of the physical domain. We therefore let

$$p = \frac{\partial p}{\partial z} = \frac{\partial p}{\partial r} = 0 \quad \text{at } (r_0, z_0) \in \Omega. \tag{8.200}$$

Compactly, the equations of motion (8.194)-(8.196) may be written as

$$\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} q \\ p \end{bmatrix} = \underbrace{\begin{bmatrix} L & -G \\ D & 0 \end{bmatrix}}_{N} \begin{bmatrix} q \\ p \end{bmatrix}. \tag{8.201}$$

Letting $q^\dagger = (u^\dagger, v^\dagger)$ and $p^\dagger$ denote the adjoint velocity field and adjoint pressure field, we seek an operator N^* such that

$$\langle (q^\dagger, p^\dagger), N(q,p) \rangle = \langle N^*(q^\dagger, p^\dagger), (q,p) \rangle. \tag{8.202}$$

Using the inner product defined in (8.193), we have

$$\begin{aligned}
&\langle (q^\dagger, p^\dagger), N(q,p)\rangle = \\
&\int_\Omega \left\{ u^\dagger \left(-u\frac{\partial U}{\partial z} - v\frac{\partial U}{\partial r} - U\frac{\partial u}{\partial z} - V\frac{\partial u}{\partial r} - \frac{\partial p}{\partial z} + \frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{\partial^2 u}{\partial z^2}\right)\right) + \right. \\
&v^\dagger \left(-u\frac{\partial V}{\partial z} - v\frac{\partial V}{\partial r} - U\frac{\partial v}{\partial z} - V\frac{\partial v}{\partial r} - \frac{\partial p}{\partial r} + \frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) - \frac{v}{r^2} + \frac{\partial^2 v}{\partial z^2}\right)\right) + \\
&\left. p^\dagger \left(\frac{\partial u}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}(rv)\right)\right\} r\,\mathrm{d}r\,\mathrm{d}z.
\end{aligned}$$

Integrating by parts twice with respect to r and with respect to z, and using the fact that the steady-state solution $Q = (U, V)$ satisfies the continuity equation (8.196), we obtain

$$\begin{aligned}
&\int_\Omega \left\{ u \left(-u^\dagger\frac{\partial U}{\partial z} - v^\dagger\frac{\partial V}{\partial z} + U\frac{\partial u^\dagger}{\partial z} + V\frac{\partial u^\dagger}{\partial r} - \frac{\partial p^\dagger}{\partial z} + \frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u^\dagger}{\partial r}\right) + \frac{\partial^2 u^\dagger}{\partial z^2}\right)\right) + \right. \\
&v \left(-u^\dagger\frac{\partial U}{\partial r} - v^\dagger\frac{\partial V}{\partial r} + U\frac{\partial v^\dagger}{\partial z} + V\frac{\partial v^\dagger}{\partial r} - \frac{\partial p^\dagger}{\partial r} + \frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v^\dagger}{\partial r}\right) - \frac{v^\dagger}{r^2} + \frac{\partial^2 v^\dagger}{\partial z^2}\right)\right) + \\
&\left. p \left(\frac{\partial u^\dagger}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}(rv^\dagger)\right)\right\} r\,\mathrm{d}r\,\mathrm{d}z + I_1 + I_2 = \langle N^*(q^\dagger, p^\dagger), (q,p)\rangle,
\end{aligned}$$

where I_1 and I_2 are boundary integrals arising from the integration by parts and they are given by

$$\begin{aligned}
I_1 = \int \Big\{ &- ru^\dagger Vu + \frac{1}{Re}\left(u^\dagger r\frac{\partial u}{\partial r} - ur\frac{\partial u^\dagger}{\partial r}\right) - rv^\dagger Vv - rv^\dagger p + \\
&\frac{1}{Re}\left(v^\dagger r\frac{\partial v}{\partial r} - vr\frac{\partial v^\dagger}{\partial r}\right) + rp^\dagger v\Big\}\Big|_{r=0}^{r=L_r} \mathrm{d}z,
\end{aligned}$$

and

$$\begin{aligned}
I_2 = \int \Big\{ &- u^\dagger Uu - u^\dagger p + \frac{1}{Re}\left(u^\dagger\frac{\partial u}{\partial z} - u\frac{\partial u^\dagger}{\partial z}\right) - v^\dagger Uv + \\
&\frac{1}{Re}\left(v^\dagger\frac{\partial v}{\partial z} - v\frac{\partial v^\dagger}{\partial z}\right) + p^\dagger u\Big\}\Big|_{z=0}^{z=L_z} r\,\mathrm{d}r.
\end{aligned}$$

Using the fact that $V(r=0) = V(r=L_r) = 0$, the boundary integrals I_1 and I_2 vanish if the adjoint fields satisfy the following boundary conditions:

$$\frac{\partial u^\dagger}{\partial r} = v^\dagger = 0 \quad \text{at } r = 0 \tag{8.203}$$

$$u^\dagger = v^\dagger = 0 \quad \text{at } r = L_r,\ z = 0 \tag{8.204}$$

$$u^\dagger = v^\dagger U + \frac{1}{Re}\frac{\partial v^\dagger}{\partial z} = p^\dagger - \frac{1}{Re}\frac{\partial u^\dagger}{\partial z} = 0 \quad \text{at } z = L_z. \tag{8.205}$$

A noteworthy difference between the forward and adjoint formulations, is the fact that a pressure boundary condition at the outflow $z = L_z$ now arises in the adjoint formulation, while in the

forward formulation we had prescribed the constraints (8.200) at $(r_0, z_0) \in \Omega$. As a concluding remark, the pressure may be removed from both the forward and adjoint formulations in a similar fashion as described in appendix Appendix 8.G. The pressure Poisson equation arising in the forward formulation may be solved with the constraints in (8.200), while the Poisson equation arising in the adjoint formulation may be solved using the pressure boundary condition in (8.205).

Chapter 9

Linearly-Recurrent Autoencoder Networks for Learning Dynamics

Samuel E. Otto and Clarence W. Rowley

This paper describes a method for learning low-dimensional approximations of nonlinear dynamical systems, based on neural-network approximations of the underlying Koopman operator. Extended Dynamic Mode Decomposition (EDMD) provides a useful data-driven approximation of the Koopman operator for analyzing dynamical systems. This paper addresses a fundamental problem associated with EDMD: a trade-off between representational capacity of the dictionary and over-fitting due to insufficient data. A new neural network architecture combining an autoencoder with linear recurrent dynamics in the encoded state is used to learn a low-dimensional and highly informative Koopman-invariant subspace of observables. A method is also presented for balanced model reduction of over-specified EDMD systems in feature space. Nonlinear reconstruction using partially linear multi-kernel regression aims to improve reconstruction accuracy from the low-dimensional state when the data has complex but intrinsically low-dimensional structure. The techniques demonstrate the ability to identify Koopman eigenfunctions of the unforced Duffing equation, create accurate low-dimensional models of an unstable cylinder wake flow, and make short-time predictions of the chaotic Kuramoto-Sivashinsky equation.

9.1 Introduction

The Koopman operator first introduced in [136] describes how Hilbert space functions on the state of a dynamical system evolve in time. These functions, referred to as observables, may correspond to measurements taken during an experiment or the output of a simulation. This makes the Koopman operator a natural object to consider for data-driven analysis of dynamical systems. Such an approach is also appealing because the Koopman operator is linear, though infinite dimensional, enabling the concepts of modal analysis for linear systems to be extended to dynamics of observables in nonlinear systems. Hence, the invariant subspaces and eigenfunctions of the Koopman operator are of particular interest and provide useful features for describing the system if they can be found. For example, level sets of Koopman eigenfunctions may be used to form partitions of the phase space into ergodic sets along with periodic and wandering chains of sets [38]. They allow us to parameterize limit cycles and tori as well as their basins of attraction. The eigenvalues allow us to determine the stability of these structures and the frequencies of periodic and quasiperiodic attractors [178]. Furthermore, by projecting the full state as an observable onto the eigenfunctions of the Koopman operator, it is decomposed into a linear superposition of components called Koopman modes which each have a fixed frequency and rate of decay. Koopman modes therefore provide useful coherent structures for studying the system's evolution and dominant pattern-forming behaviors. This has made the Koopman operator a particularly useful object of study for high-dimensional spatiotemporal systems like unsteady fluid dynamics beginning with the work of Mezić on spectral properties of dynamical systems [177] then Rowley [231] and Schmid [241] on the Dynamic Mode Decomposition (DMD) algorithm. Rowley, recognizing that DMD furnishes an approximation of the Koopman operator and its modes, applied the technique to data collected by simulating a jet in a crossflow. The Koopman modes identified salient patterns of spatially coherent structure in the flow which evolved at fixed frequencies.

The Extended Dynamic Mode Decomposition (EDMD) [280] is an algorithm for approximating the Koopman operator on a dictionary of observable functions using data. If a Koopman-invariant subspace is contained in the span of the observables included in the dictionary, then as long as enough data is used, the representation on this subspace will be exact. EDMD is a Galerkin method with a particular data-driven inner product as long as enough data is used. Specifically, this will be true as long as the rank of the data matrix is the same as the dimension of the subspace spanned by the (nonlinear) observables [226]. However, the choice of dictionary is ad hoc, and it is often not clear how to choose a dictionary that is sufficiently rich to span a useful Koopman-invariant

subspace. One might then be tempted to consider a very large dictionary, with enough capacity to represent any complex-valued function on the state space to within an ϵ tolerance. However, such a dictionary has combinatorial growth with the dimension of the state space and would be enormous for even modestly high-dimensional problems.

One approach to mitigate the cost of large or even infinite dictionaries is to formulate EDMD as a kernel method referred to as KDMD [281]. However, we are still essentially left with the same problem of deciding which kernel function to use. Furthermore, if the kernel or EDMD feature space is endowed with too much representational capacity (a large dictionary), the algorithm will over-fit the data (as we shall demonstrate with a toy problem in Example 9.2.1). EDMD and KDMD also identify a number of eigenvalues, eigenfunctions, and modes which grows with the size of the dictionary. If we want to build reduced order models of the dynamics, a small collection of salient modes or a low-dimensional Koopman invariant subspace must be identified. It is worth mentioning two related algorithms for identifying low-rank approximations of the Koopman operator. Optimal Mode Decomposition (OMD) [286] finds the optimal orthogonal projection subspace of user-specified rank for approximating the Koopman operator. Sparsity-promoting DMD [127] is a post-processing method which identifies the optimal amplitudes of Koopman modes for reconstructing the snapshot sequence with an ℓ^1 penalty. The sparsity-promoting penalty picks only the most salient Koopman modes to have nonzero amplitudes. Another related scheme is Sparse Identification of Nonlinear Dynamics (SINDy) [36] which employs a sparse regression penalty on the number of observables used to approximate nonlinear evolution equations. By forcing the dictionary to be sparse, the over-fitting problem is reduced.

In this paper, we present a new technique for learning a very small collection of informative observable functions spanning a Koopman invariant subspace from data. Two neural networks in an architecture similar to an under-complete autoencoder [99] represent the collection of observables together with a nonlinear reconstruction of the full state from these features. A learned linear transformation evolves the function values in time as in a recurrent neural network, furnishing our approximation of the Koopman operator on the subspace of observables. This approach differs from recent efforts that use neural networks to learn dictionaries for EDMD [293, 152] in that we employ a second neural network to reconstruct the full state. Ours and concurrent approaches utilizing nonlinear decoder neural networks [257, 164] enable learning of very small sets of features that carry rich information about the state and evolve linearly in time. Previous methods for data-driven analysis based on the Koopman operator utilize linear state reconstruction via the Koopman modes. Therefore they rely on an assumption that the full state observable is in the learned Koopman

invariant subspace. Nonlinear reconstruction is advantageous since it relaxes this strong assumption, allowing recent techniques to recover more information about the state from fewer observables. By minimizing the state reconstruction error over several time steps into the future, our architecture aims to detect highly observable features even if they have small amplitudes. This is the case in non-normal linear systems, for instance as arise in many fluid flows (in particular, shear flows [242]), in which small disturbances can siphon energy from mean flow gradients and excite larger-amplitude modes. The underlying philosophy of our approach is similar to Visual Interaction Networks (VINs) [276] that learn physics-based dynamics models for encoded latent variables.

Deep neural networks have gained attention over the last decade due to their ability to efficiently represent complicated functions learned from data. Since each layer of the network performs simple operations on the output of the previous layer, a deep network can learn and represent functions corresponding to high-level or abstract features. For example, your visual cortex assembles progressively more complex information sequentially from retinal intensity values to edges, to shapes, all the way up to the facial features that let you recognize your friend. By contrast, shallow networks — though still universal approximators — require exponentially more parameters to represent classes of natural functions like polynomials [223, 153] or the presence of eyes in a photograph. Function approximation using linear combinations of preselected dictionary elements is somewhat analogous to a shallow neural network where capacity is built by adding more functions. We therefore expect deep neural networks to represent certain complex nonlinear observables more efficiently than a large, shallow dictionary. Even with the high representational capacity of our deep neural networks, the proposed technique is regularized by the small number of observables we learn and is therefore unlikely to over-fit the data.

We also present a technique for constructing reduced order models in nonlinear feature space from over-specified KDMD models. Recognizing that the systems identified from data by EDMD/KDMD can be viewed as state-space systems where the output is a reconstruction of the full state using Koopman modes, we use Balanced Proper Orthogonal Decomposition (BPOD) [225] to construct a balanced reduced-order model. The resulting model consists of only those nonlinear features that are most excited and observable over a finite time horizon. Nonlinear reconstruction of the full state is introduced in order to account for complicated, but intrinsically low-dimensional data. In this way, the method is analogous to an autoencoder where the nonlinear decoder is learned separately from the encoder and dynamics.

Finally, the two techniques we introduce are tested on a range of example problems. We first investigate the eigenfunctions learned by the autoencoder and the KDMD reduced-order model by

identifying and parameterizing basins of attraction for the unforced Duffing equation. The prediction accuracy of the models is then tested on a high-dimensional cylinder wake flow problem. Finally, we see if the methods can be used to construct reduced order models for the short-time dynamics of the chaotic Kuramoto-Sivashinsky equation. Several avenues for future work and extensions of our proposed methods are discussed in the conclusion.

9.2 Extended Dynamic Mode Decomposition

Before discussing the new method for approximating the Koopman operator, it will be beneficial to review the formulation of Extended Dynamic Mode Decomposition (EDMD) [280] and its kernel variant KDMD [281]. Besides providing the context for developing the new technique, it will be useful to compare our results to those obtained using reduced order KDMD models.

9.2.1 The Koopman operator and its modes

Consider a discrete-time autonomous dynamical system on the state space $\mathcal{M} \subset \mathbb{R}^n$ given by the function $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t)$. Let $\mathcal{F}$ be a Hilbert space of complex-valued functions on $\mathcal{M}$. We refer to elements of $\mathcal{F}$ as observables. The Koopman operator acts on an observable $\psi \in \mathcal{F}$ by composition with the dynamics:

$$\mathcal{K}\psi = \psi \circ \mathbf{f}. \tag{9.1}$$

It is easy to see that the Koopman operator is linear; however, the Hilbert space $\mathcal{F}$ on which it acts is often infinite dimensional.[1] Since the operator $\mathcal{K}$ is linear, it may have eigenvalues and eigenfunctions. If a given observable lies within the span of these eigenfunctions, then we can predict the time evolution of the observable's values, as the state evolves according to the dynamics. Let $\mathbf{g} : \mathcal{M} \to \mathbb{C}^{N_0}$ be a vector-valued observable whose components are in the span of the Koopman eigenfunctions. The vector-valued coefficients needed to reconstruct $\mathbf{g}$ in a Koopman eigenfunction basis are called the Koopman modes associated with $\mathbf{g}$.

In particular, the dynamics of the original system can be recovered by taking the observable $\mathbf{g}$ to be the full-state observable defined by $\mathbf{g}(\mathbf{x}) = \mathbf{x}$. Assume $\mathcal{K}$ has eigenfunctions $\{\varphi_1, \ldots, \varphi_K\}$ with corresponding eigenvalues $\{\mu_1, \ldots, \mu_K\}$, and suppose the components of the vector-valued function $\mathbf{g}$

[1] One must also be careful about the choice of the space $\mathcal{F}$, since $\psi \circ \mathbf{f}$ must also lie in $\mathcal{F}$ for any $\psi \in \mathcal{F}$. It is common, especially in the ergodic theory literature, to assume that $\mathcal{M}$ is a measure space and $\mathbf{f}$ is measure preserving. In this case, this difficulty goes away: one lets $\mathcal{F} = L^2(\mathcal{M})$, and since $\mathbf{f}$ is measure preserving, it follows that $\mathcal{K}$ is an isometry.

lie within the span of $\{\varphi_k\}$. The Koopman modes $\boldsymbol{\xi}_k$ are then defined by

$$\mathbf{x} = \sum_{k=1}^{K} \boldsymbol{\xi}_k \varphi_k(\mathbf{x}), \tag{9.2}$$

from which we can recover the evolution of the state, according to

$$\mathbf{f}^t(\mathbf{x}) = \sum_{k=1}^{K} \boldsymbol{\xi}_k \mu_k^t \varphi_k(\mathbf{x}). \tag{9.3}$$

The entire orbit of an initial point $\mathbf{x}_0$ may thus be determined by evaluating the eigenfunctions at $\mathbf{x}_0$ and evolving the coefficients $\boldsymbol{\xi}_k$ in time by multiplying by the eigenvalues. The eigenfunctions φ_k are intrinsic features of the dynamical system which decompose the state dynamics into a linear superposition of autonomous first-order systems. The Koopman modes $\boldsymbol{\xi}_k$ depend on the coordinates we use to represent the dynamics, and allow us to reconstruct the dynamics in those coordinates.

9.2.2 Approximating Koopman on an explicit dictionary with EDMD

The aim of EDMD is to approximate the Koopman operator using data snapshot pairs taken from the system $\{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{M}$ where $\mathbf{y}_j = \mathbf{f}(\mathbf{x}_j)$. For convenience, we organize these data into matrices

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_M \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_M \end{bmatrix}. \tag{9.4}$$

Consider a finite dictionary of observable functions $\mathcal{D} = \{\psi_i : \mathcal{M} \to \mathbb{C}\}_{i=1}^{N}$ that span a subspace $\mathcal{F}_\mathcal{D} \subset \mathcal{F}$. EDMD approximates the Koopman operator on $\mathcal{F}_\mathcal{D}$ by minimizing an empirical error when the Koopman operator acts on elements $\psi \in \mathcal{F}_\mathcal{D}$. Introducing the feature map

$$\boldsymbol{\Psi}(\mathbf{x}) = \begin{bmatrix} \psi_1(\mathbf{x}) & \psi_2(\mathbf{x}) & \cdots & \psi_N(\mathbf{x}) \end{bmatrix}^*, \tag{9.5}$$

where $(\cdot)^*$ is the complex conjugate transpose, we may succinctly express elements in the dictionary's span as a linear combination with coefficients $\mathbf{a}$:

$$\psi_\mathbf{a} = \boldsymbol{\Psi}^* \mathbf{a}. \tag{9.6}$$

EDMD represents an approximation of the Koopman operator as a matrix $\mathbf{K} : \mathbb{C}^N \to \mathbb{C}^N$ that updates the coefficients in the linear combination Eq. 9.6 to approximate the new observable $\mathcal{K}\psi_\mathbf{a}$ in the span of the dictionary. Of course we cannot expect the span of our dictionary to be an invariant

subspace, so the approximation satisfies

$$\mathcal{K}\psi_{\mathbf{a}} = \mathbf{\Psi}^*\mathbf{K}\mathbf{a} + r, \tag{9.7}$$

where $r \in \mathcal{F}$ is a residual that we wish to minimize in some sense, by appropriate choice of the matrix $\mathbf{K}$. The values of the Koopman-updated observables are known at each of the data points $\mathcal{K}\psi_{\mathbf{a}}(\mathbf{x}_j) = \psi_{\mathbf{a}} \circ \mathbf{f}(\mathbf{x}_j) = \psi_{\mathbf{a}}(\mathbf{y}_j)$, allowing us to define an empirical error of the approximation in terms of the residuals at these data points. Minimizing this error yields the EDMD matrix $\mathbf{K}$. The empirical error on a single observable in $\mathcal{F}_{\mathcal{D}}$ is given by

$$\begin{aligned} J(\psi_{\mathbf{a}}) &= \sum_{i=1}^{M} \left|\psi_{\mathbf{a}}(\mathbf{y}_i) - \mathbf{\Psi}(\mathbf{x}_i)^*\mathbf{K}\mathbf{a}\right|^2 \\ &= \sum_{i=1}^{M} \left|\left(\mathbf{\Psi}(\mathbf{y}_i)^* - \mathbf{\Psi}(\mathbf{x}_i)^*\mathbf{K}\right)\mathbf{a}\right|^2 \end{aligned} \tag{9.8}$$

and the total empirical error on a set of observables $\left\{\psi_{\mathbf{a}_j}\right\}_{j=1}^{N'}$, $N' \geq N$ spanning $\mathcal{F}_{\mathcal{D}}$ is given by

$$J = \sum_{j=1}^{N'} \sum_{i=1}^{M} \left|\left(\mathbf{\Psi}(\mathbf{y}_i)^* - \mathbf{\Psi}(\mathbf{x}_i)^*\mathbf{K}\right)\mathbf{a}_j\right|^2. \tag{9.9}$$

Regardless of how the above observables are chosen, the matrix $\mathbf{K}$ that minimizes Eq. 9.9 is given by

$$\mathbf{K} = \mathbf{G}^+\mathbf{A}, \qquad \mathbf{G} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{\Psi}(\mathbf{x}_i)\mathbf{\Psi}(\mathbf{x}_i)^*, \qquad \mathbf{A} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{\Psi}(\mathbf{x}_i)\mathbf{\Psi}(\mathbf{y}_i)^*, \tag{9.10}$$

where $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse of a matrix.

The EDMD solution Eq. 9.10 requires us to evaluate the entries of $\mathbf{G}$ and $\mathbf{A}$ and compute the pseudoinverse of $\mathbf{G}$. Both matrices have size $N \times N$ where N is the number of observables in our dictionary. In problems where the state dimension is large, as it is in many fluids datasets coming from experimental or simulated flow fields, a very large number of observables is needed to achieve even modest resolution on the phase space. The problem of evaluating and storing the matrices needed for EDMD becomes intractable as N grows large. However, the rank of these matrices does not exceed $\min\{M, N\}$. The kernel DMD method provides a way to compute an EDMD-like approximation of the Koopman operator using kernel matrices whose size scales with the number of snapshot pairs M^2 instead of the number of features N^2. This makes it advantageous for problems where the state dimension is greater than the number of snapshots or where very high resolution of

the Koopman operator on a large dictionary is needed.

9.2.3 Approximating Koopman on an implicit dictionary with KDMD

KDMD can be derived by considering the data matrices

$$\mathbf{\Psi_X} = \begin{bmatrix} \mathbf{\Psi}(\mathbf{x}_1) & \mathbf{\Psi}(\mathbf{x}_2) & \cdots & \mathbf{\Psi}(\mathbf{x}_M) \end{bmatrix}, \qquad \mathbf{\Psi_Y} = \begin{bmatrix} \mathbf{\Psi}(\mathbf{y}_1) & \mathbf{\Psi}(\mathbf{y}_2) & \cdots & \mathbf{\Psi}(\mathbf{y}_M) \end{bmatrix}, \tag{9.11}$$

in feature space i.e., after applying the now only hypothetical feature map $\mathbf{\Psi}$ to the snapshots. We will see that the final results of this approach make reference only to inner products $\mathbf{\Psi}(\mathbf{x})^*\mathbf{\Psi}(\mathbf{z})$ which will be defined using a suitable non-negative definite kernel function $k(\mathbf{x}, \mathbf{z})$. Choice of such a kernel function implicitly defines the corresponding dictionary via Mercer's theorem. By employing simply-defined kernel functions, the inner products are evaluated at a lower computational cost than would be required to evaluate a high or infinite dimensional feature map and compute inner products in the feature space explicitly.

The total empirical error for EDMD Eq. 9.9 can be written as the Frobenius norm

$$J = \left\| \left(\mathbf{\Psi_Y^*} - \mathbf{\Psi_X^*} \mathbf{K} \right) \mathbf{A} \right\|_F^2, \quad \text{where} \quad \mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_{N'} \end{bmatrix}. \tag{9.12}$$

Let us consider an economy sized SVD $\mathbf{\Psi_X} = \mathbf{U\Sigma V}^*$, the existence of which is guaranteed by the finite rank r of our feature data matrix. In Eq. 9.12 we see that any components of the range $\mathcal{R}(\mathbf{K})$ orthogonal to $\mathcal{R}(\mathbf{\Psi_X})$ are annihilated by $\mathbf{\Psi_X^*}$ and cannot be inferred from the data. We therefore restrict the dictionary to those features which can be represented in the range of the feature space data $\mathcal{F}_\mathcal{D} = \{\psi_\mathbf{a} = \mathbf{\Psi}^*\mathbf{a} : \mathbf{a} \in \mathcal{R}(\mathbf{U})\}$ and represent $\mathbf{K} = \mathbf{U}\hat{\mathbf{K}}\mathbf{U}^*$ for some matrix $\hat{\mathbf{K}} \in \mathbb{C}^{r \times r}$. After some manipulation, it can be shown that minimizing the empirical error Eq. 9.12 with respect to $\hat{\mathbf{K}}$ is equivalent to minimizing

$$J' = \left\| \left(\mathbf{V}^* \mathbf{\Psi_Y^*} - \mathbf{\Sigma} \hat{\mathbf{K}} \mathbf{U}^* \right) \mathbf{A} \right\|_F^2. \tag{9.13}$$

Regardless of how the columns of $\mathbf{A}$ are chosen, as long as $\mathcal{R}(\mathbf{A}) = \mathcal{R}(\mathbf{U})$ the minimum norm solution for the KDMD matrix is

$$\hat{\mathbf{K}} = \mathbf{\Sigma}^+ \mathbf{V}^* \mathbf{\Psi_Y^*} \mathbf{U} = \mathbf{\Sigma}^+ \mathbf{V}^* \mathbf{\Psi_Y^*} \mathbf{\Psi_X} \mathbf{V} \mathbf{\Sigma}^+. \tag{9.14}$$

Each component in the above KDMD approximation can be found entirely in terms of inner products in the feature space, enabling the use of a kernel function to implicitly define the feature space. The

two matrices whose entries are $[\mathbf{K_{XX}}]_{ij} = \mathbf{\Psi}(\mathbf{x}_i)^*\mathbf{\Psi}(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$ and $[\mathbf{K_{YX}}]_{ij} = [\mathbf{\Psi}_\mathbf{Y}^*\mathbf{\Psi}_\mathbf{X}]_{ij} = \mathbf{\Psi}(\mathbf{y}_i)^*\mathbf{\Psi}(\mathbf{x}_j) = k(\mathbf{y}_i, \mathbf{x}_j)$ are computed using the kernel. The Hermitian eigenvalue decomposition $\mathbf{K_{XX}} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^*$ provides the matrices $\mathbf{V}$ and $\mathbf{\Sigma}$.

It is worth pointing out that the EDMD and KDMD solutions Eq. 9.10 and Eq. 9.14 can be regularized by truncating the rank r of the SVD $\mathbf{\Psi_X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$. In EDMD, we recognize that $\mathbf{G} = \frac{1}{M}\mathbf{\Psi_X}\mathbf{\Psi_X}^* = \frac{1}{M}\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^*$ is a Hermitian eigendecomposition. Before finding the pseudoinverse, the rank is truncated to remove the dyadic components having small singular values.

9.2.4 Computing Koopman eigenvalues, eigenfunctions, and modes

Suppose that $\varphi = \mathbf{\Psi}^*\mathbf{w}$ is an eigenvector of the Koopman operator in the span of the dictionary with eigenvalue μ. Suppose also that $\mathbf{w} = \mathbf{U}\hat{\mathbf{w}}$ is in the span of the data in feature space. From $\mathcal{K}\varphi = \mu\varphi$ it follows that $\mathbf{\Psi}_\mathbf{Y}^*\mathbf{w} = \mu\mathbf{\Psi}_\mathbf{X}^*\mathbf{w}$ by substituting all of the snapshot pairs. Left-multiplying by $\frac{1}{M}\mathbf{\Psi_X}$ and taking the pseudoinverse, we obtain $(\mathbf{G}^+\mathbf{A})\mathbf{w} = \mu(\mathbf{G}^+\mathbf{G})\mathbf{w} = \mu\mathbf{w}$ where the second equality holds because $\mathbf{w} \in \mathcal{R}(\mathbf{\Psi_X})$. Therefore, $\mathbf{w}$ is an eigenvector with eigenvalue μ of the EDMD matrix Eq. 9.10. In terms of the coefficients $\hat{\mathbf{w}}$, we have $\mathbf{\Psi}_\mathbf{Y}^*\mathbf{U}\hat{\mathbf{w}} = \mu\mathbf{\Psi}_\mathbf{X}^*\mathbf{U}\hat{\mathbf{w}}$, which upon substituting the definition $\mathbf{\Psi_X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ gives $\mathbf{\Psi}_\mathbf{Y}^*\mathbf{\Psi_X}\mathbf{V}\mathbf{\Sigma}^+\hat{\mathbf{w}} = \mu\mathbf{V}\mathbf{\Sigma}\hat{\mathbf{w}}$. From the previous statement we it is evident that $\mathbf{\Sigma}^+\mathbf{V}^*\mathbf{\Psi}_\mathbf{Y}^*\mathbf{\Psi_X}\mathbf{V}\mathbf{\Sigma}^+\hat{\mathbf{w}} = \hat{\mathbf{K}}\hat{\mathbf{w}} = \mu\hat{\mathbf{w}}$. Hence, $\hat{\mathbf{w}}$ is an eigenvector of the KDMD matrix Eq. 9.14 with eigenvalue μ. Unfortunately, the converses of these statements do not hold. Nonetheless, approximations of Koopman eigenfunctions,

$$\varphi(\mathbf{x}) = \mathbf{\Psi}(\mathbf{x})^*\mathbf{w} = \mathbf{\Psi}(\mathbf{x})^*\mathbf{\Psi_X}\mathbf{V}\mathbf{\Sigma}^+\hat{\mathbf{w}}, \tag{9.15}$$

are formed using the right eigenvectors $\mathbf{w}$ and $\hat{\mathbf{w}}$ of $\mathbf{K}$ and $\hat{\mathbf{K}}$ respectively. In Eq. 9.15 the inner products $\mathbf{\Psi}(\mathbf{x})^*\mathbf{\Psi_X}$ can be found by evaluating the kernel function between $\mathbf{x}$ and each point in the training data $\{\mathbf{x}_j\}_{j=1}^M$ yielding a row-vector.

The Koopman modes $\{\boldsymbol{\xi}_k\}_{k=1}^r$ associated with the full state observable reconstruct the state as a linear combination of Koopman eigenfunctions. They can be found from the provided training data using a linear regression process. Let us define the matrices

$$\mathbf{\Xi} = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \cdots & \boldsymbol{\xi}_r \end{bmatrix}, \qquad \mathbf{\Phi_X} = \begin{bmatrix} \varphi_1(\mathbf{x}_1) & \cdots & \varphi_1(\mathbf{x}_M) \\ \vdots & \ddots & \vdots \\ \varphi_r(\mathbf{x}_1) & \cdots & \varphi_r(\mathbf{x}_M) \end{bmatrix} = \mathbf{W}_R^T\overline{\mathbf{\Psi_X}} = \hat{\mathbf{W}}_R^T\mathbf{\Sigma}\mathbf{V}^T, \tag{9.16}$$

containing the Koopman modes and eigenfunction values at the training points. In the above, $\mathbf{W}_R$ and $\hat{\mathbf{W}}_R$ are the matrices whose columns are the right eigenvectors of $\mathbf{K}$ and $\hat{\mathbf{K}}$ respectively. Seeking to linearly reconstruct the state from the eigenfunction values at each training point, the regression problem,

$$\underset{\boldsymbol{\Xi}\in\mathbb{C}^{n\times r}}{\text{minimize}} \quad \|\mathbf{X}-\boldsymbol{\Xi}\boldsymbol{\Phi}_{\mathbf{X}}\|_F^2\,, \tag{9.17}$$

is formulated. The solution to this standard least squares problem is

$$\boldsymbol{\Xi} = \mathbf{X}\overline{\boldsymbol{\Psi}_{\mathbf{X}}^{+}\mathbf{W}_L} = \mathbf{X}\overline{\mathbf{V}\boldsymbol{\Sigma}^{+}\hat{\mathbf{W}}_L}, \tag{9.18}$$

where $\mathbf{W}_L$ and $\hat{\mathbf{W}}_L$ are the left eigenvector matrices of $\mathbf{K}$ and $\hat{\mathbf{K}}$ respectively. These matrices must be suitably normalized so that the left and right eigenvectors form bi-orthonormal sets $\mathbf{W}_L^*\mathbf{W}_R = \mathbf{I}_r$ and $\hat{\mathbf{W}}_L^*\hat{\mathbf{W}}_R = \mathbf{I}_r$.

9.2.5 Drawbacks of EDMD

One of the drawbacks of EDMD and KDMD is that the accuracy depends on the chosen dictionary. For high-dimensional data sets, constructing and evaluating an explicit dictionary becomes prohibitively expensive. Though the kernel method allows us to use high-dimensional dictionaries implicitly, the choice of kernel function significantly impacts the results. In both techniques, higher resolution is achieved directly by adding more dictionary elements. Therefore, enormous dictionaries are needed in order to represent complex features. The shallow representation of features in terms of linear combinations of dictionary elements means that the effective size of the dictionary must be limited by the rank of the training data in feature space. As one increases the resolution of the dictionary, the rank r of the feature space data $\boldsymbol{\Psi}_{\mathbf{X}}$ grows and eventually reaches the number of points M assuming the points are distinct. The number of data points therefore is an upper bound on the effective number of features we can retain for EDMD or KDMD. This effective dictionary selection is implicit when we truncate the SVD of $\mathbf{G}$ or $\boldsymbol{\Psi}_{\mathbf{X}}$. It is when $r = M$ that we have retained enough features to memorize the data set up to projection of $\boldsymbol{\Psi}_{\mathbf{Y}}$ onto $\mathcal{R}(\boldsymbol{\Psi}_{\mathbf{X}})$. Consequently, overfitting becomes problematic as we seek dictionaries with high enough resolution to capture complex features. We illustrate this problem with the following simple example.

Example 9.2.1. *Let us consider the linear dynamical system*

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_{t+1}) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}_{t+1} \tag{9.19}$$

with $\mathbf{x} = [x_1,\ x_2]^T \in \mathbb{R}^2$. *We construct this example to reflect the behavior of EDMD with rich dictionaries containing more elements than snapshots. Suppose that we have only two snapshot pairs,*

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 0.5 \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} 1 & 1 \\ 0.5 & 0.25 \end{bmatrix}, \tag{9.20}$$

taken by evolving the trajectory two steps from the initial condition $x_0 = [1,\ 1]^T$. *Let us define the following dictionary. Its first two elements are Koopman eigenfunctions whose values are sufficient to describe the full state. In fact, EDMD recovers the original dynamics perfectly from the given snapshots when we take only these first two observables. In this example, we show that by including an extra, unnecessary observable we get a much worse approximation of the dynamics. A third dictionary element which is not an eigenfunction is included in order to demonstrate the effects of an overcomplete dictionary. With these dictionary elements, the data matrices are*

$$\mathbf{\Psi}(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 \\ (x_1)^2 + (x_2)^2 \end{bmatrix} \implies \mathbf{\Psi}_\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 0.5 \\ 2 & 1.25 \end{bmatrix}, \quad \mathbf{\Psi}_\mathbf{Y} = \begin{bmatrix} 1 & 1 \\ 0.5 & 0.25 \\ 1.25 & 1.0625 \end{bmatrix}. \tag{9.21}$$

Applying Eq. 9.10 we compute the EDMD matrix and its eigendecomposition,

$$\mathbf{K} = \begin{bmatrix} 0.9286 & -0.1071 & 0.7321 \\ -0.2143 & 0.1786 & -0.0536 \\ 0.1429 & 0.2143 & 0.2857 \end{bmatrix} \implies \begin{cases} \mu_1 = 1.0413 \\ \mu_2 = 0 \\ \mu_3 = 0.3515 \end{cases}, \tag{9.22}$$

as well as the eigenfunction approximations,

$$\begin{bmatrix} \varphi_1(\mathbf{x}) \\ \varphi_2(\mathbf{x}) \\ \varphi_3(\mathbf{x}) \end{bmatrix} = \mathbf{W}_R^T \overline{\mathbf{\Psi}(\mathbf{x})} = \begin{bmatrix} -0.9627 & 0.2461 & -0.1122 \\ 0.5735 & 0.4915 & 0.5722 \\ -0.6013 & 0.5722 & 0.5577 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ (x_1)^2 + (x_2)^2 \end{bmatrix}. \tag{9.23}$$

It is easy to see that none of the eigenfunctions or eigenvalues are correct for the given system even

though the learned matrix satisfies $\|\mathbf{\Psi}_{\mathbf{Y}}^* - \mathbf{\Psi}_{\mathbf{X}}^*\mathbf{K}\|_F < 6 * 10^{-15}$ *with 16 digit precision computed with standard Matlab tools. This shows that even with a single additional function in the dictionary, we have severely over-fit the data. This is surprising since our original dictionary included two eigenfunctions by definition. The nuance comes since EDMD is only guaranteed to capture eigenfunctions* $\varphi(\mathbf{x}) = \mathbf{w}^T\overline{\mathbf{\Psi}(\mathbf{x})}$ *where* $\mathbf{w}$ *is in the span of the feature space data* $\mathcal{R}(\mathbf{\Psi}_{\mathbf{X}})$*. In this example, the true eigenfunctions do not satisfy this condition; one can check that neither* $\mathbf{w} = [1,\ 0,\ 0]^T$ *nor* $\mathbf{w} = [0,\ 1,\ 0]^T$ *is in* $\mathcal{R}(\mathbf{\Psi}_{\mathbf{X}})$*.*

9.3 Recent approach for dictionary learning

Example 9.2.1 makes clear the importance of choosing an appropriate dictionary prior to performing EDMD. In two recent papers [293, 152], the universal function approximation property of neural networks was used to learn dictionaries for approximating the Koopman operator. A fixed number of observables making up the dictionary are given by a neural network $\mathbf{\Psi}(\mathbf{x};\boldsymbol{\theta}) \in \mathbb{R}^d$ parameterized by $\boldsymbol{\theta}$. The linear operator $\mathbf{K}^T \in \mathbb{R}^{d\times d}$ evolving the dictionary function values one time step into the future is learned simultaneously through minimization of

$$J(\mathbf{K},\boldsymbol{\theta}) = \sum_{i=1}^{M} \left\|\mathbf{\Psi}(\mathbf{y}_i;\boldsymbol{\theta}) - \mathbf{K}^T\mathbf{\Psi}(\mathbf{x}_i;\boldsymbol{\theta})\right\|^2 + \Omega(\mathbf{K},\boldsymbol{\theta}). \tag{9.24}$$

A schematic of this architecture is depicted in Figure 9.3.1 where $\mathbf{z} = \mathbf{\Psi}(\mathbf{x};\boldsymbol{\theta})$ and $\mathbf{z}^{\#} = \mathbf{\Psi}(\mathbf{y};\boldsymbol{\theta})$ are the dictionary function values before and after the time increment. The term Ω is used for regularization and the Tikhonov regularizer $\Omega(\mathbf{K},\boldsymbol{\theta}) = \lambda\|\mathbf{K}\|_F^2$ was used. One notices that as the problem is formulated, the trivial solution $\mathbf{\Psi}(\mathbf{x};\boldsymbol{\theta}) \equiv \mathbf{0}_d$ and $\mathbf{K} = \mathbf{0}_{d\times d}$ is a global minimizer. [152] solves this problem by fixing some of the dictionary elements to not be trainable. Since the full state observable is to be linearly reconstructed in the span of the dictionary elements via the Koopman modes, it is natural to fix the first N dictionary elements to be $\mathbf{x}$ while learning the remaining $d-N$ elements through parameterization as a neural network. The learned dictionary then approximately spans a Koopman invariant subspace containing the full state observable. Training proceeds by iterating two steps: (1) Fix $\boldsymbol{\theta}$ and optimize $\mathbf{K}$ by explicit solution of the least squares problem; Then (2) fix $\mathbf{K}$ and optimize $\boldsymbol{\theta}$ by gradient descent. The algorithm implemented in [152] is summarized in Algorithm 4.

When learning an adaptive dictionary of a fixed size using a neural network (or other function approximation method), let us consider two objects: the dictionary space $\mathcal{S} = \{\psi_j(\bullet;\boldsymbol{\theta}) : \mathbb{R}^n \to \mathbb{R} \ : \ \forall\boldsymbol{\theta} \in \Theta,\ j = 1,\ldots$

Algorithm 4 EDMD with dictionary learning [152]

Initialize $\mathbf{K}$, $\boldsymbol{\theta}$
while $J(\mathbf{K}, \boldsymbol{\theta}) > \epsilon$ **do**
  Tikhonov regularized EDMD: $\mathbf{K} \leftarrow (\mathbf{G}(\boldsymbol{\theta}) + \lambda \mathbf{I}_d)^{-1} \mathbf{A}(\boldsymbol{\theta})$
  Gradient descent: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \delta \nabla_{\boldsymbol{\theta}} J(\mathbf{K}, \boldsymbol{\theta})$
end while

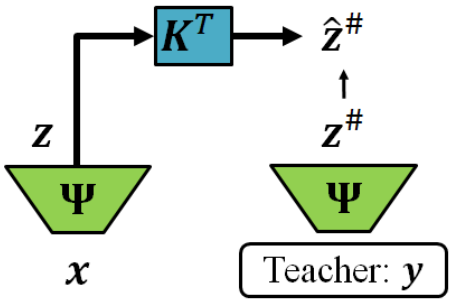


Figure 9.3.1: EDMD with dictionary learning architecture

is the set of all functions which can be parameterized by the neural network and the dictionary $\mathcal{D}(\boldsymbol{\theta}) = \{\psi_j(\bullet; \boldsymbol{\theta}) : \mathbb{R}^n \to \mathbb{R} \ : \ j = 1, \ldots, d\}$ is the d elements of $\mathcal{S}$ fixed by choosing $\boldsymbol{\theta}$. In EDMD and the dictionary learning approach just described, the Koopman operator is always approximated on a subspace $\mathcal{F}_{\mathbf{X}} = \{\boldsymbol{\Psi}^* \mathbf{w} \ : \ \mathbf{w} \in \mathcal{R}(\boldsymbol{\Psi}_{\mathbf{X}})\} \subset \mathcal{F}_{\mathcal{D}} = \operatorname{span} \mathcal{D}$. As discussed earlier, the EDMD method always uses $\mathcal{S} = \mathcal{D}$ and the only way to increase resolution and feature complexity is to grow the dictionary — leading to the over-fitting problems illustrated in Example 9.2.1. By contrast, the dictionary learning approach enables us to keep the size of the dictionary relatively small while exploring a much larger space $\mathcal{S}$. In particular, the dictionary size is presumed to be much smaller than the total number of training data points and probably small enough so that $\mathcal{F}_{\mathbf{X}} = \mathcal{F}_{\mathcal{D}}$. Otherwise, the number of functions d learned by the network could be reduced so that this becomes true. The small dictionary size therefore prevents the method from memorizing the snapshot pairs without learning true invariant subspaces. This is not done at the expense of resolution since the allowable complexity of functions in $\mathcal{S}$ is extremely high.

Deep neural networks are advantageous since they enable highly efficient representations of certain natural classes of complex features [223, 153]. In particular, deep neural networks are capable of learning functions whose values are built by applying many simple operations in succession. It is shown empirically that this is indeed an important and natural class of functions since deep neural networks have recently enabled near human level performance on tasks like image and handwritten digit recognition [99]. This proved to be a useful property for dictionary learning for EDMD since [293, 152] achieve state of the art results on examples including the Duffing equation, Kuramoto-

Sivashinsky PDE, a system representing the glycolysis pathway, and power systems.

9.4 New approach: deep feature learning using the LRAN

By removing the constraint that the full state observable is in the learned Koopman invariant subspace, one can do even better. This is especially important for high-dimensional systems where it would be prohibitive to train such a large neural network-based dictionary with limited training data. Furthermore, it may simply not be the case that the full state observable lies in a finite-dimensional Koopman invariant subspace. The method described here is capable of learning extremely low-dimensional invariant subspaces limited only by the intrinsic dimensionality of linearly-evolving patterns in the data. A schematic of our general architecture is presented in Figure 9.4.1. The dictionary function values are given by the output of an encoder neural network $\mathbf{z} = \boldsymbol{\Psi}(\mathbf{x}; \boldsymbol{\theta}_{enc})$ parameterized by $\boldsymbol{\theta}_{enc}$. We avoid the trivial solution by nonlinearly reconstructing an approximation of the full state using a decoder neural network $\hat{\mathbf{x}} = \tilde{\boldsymbol{\Psi}}(\mathbf{z}; \boldsymbol{\theta}_{dec})$ parameterized by $\boldsymbol{\theta}_{dec}$. The decoder network takes the place of Koopman modes for reconstructing the full state from eigenfunction values. However, if Koopman modes are desired it is still possible to compute them using two methods. The first is to employ the same regression procedure whose solution is given by Eq. 9.18 to compute the Koopman modes from the EDMD dictionary provided by the encoder. Reconstruction using the Koopman modes will certainly achieve lower accuracy than the nonlinear decoder, but may still provide a useful tool for feature extraction and visualization. The other option is to employ a linear decoder network $\tilde{\boldsymbol{\Psi}}(\mathbf{z}; \boldsymbol{\theta}_{dec}) = \mathbf{B}(\boldsymbol{\theta}_{dec})\mathbf{z}$ where $\mathbf{B}(\boldsymbol{\theta}_{dec}) \in \mathbb{R}^{n \times d}$ is a matrix whose entries are parameterized by $\boldsymbol{\theta}_{dec}$. The advantage of using a nonlinear decoder network is that the full state observable need not be in the span of the learned encoder dictionary functions. A nonlinear decoder can reconstruct more information about the full state from fewer features provided by the encoder. This enables the dictionary size d to be extremely small — yet informative enough to enable nonlinear reconstruction. This is exactly the principle underlying the success of undercomplete autoencoders for feature extraction, manifold learning, and dimensionality reduction. Simultaneous training of the encoder and decoder networks extract rich dictionary elements which the decoder can use for reconstruction.

The technique includes a linear time evolution process given by the matrix $\mathbf{K}(\boldsymbol{\theta}_{\mathbf{K}})$ parameterized by $\boldsymbol{\theta}_{\mathbf{K}}$. This matrix furnishes our approximation of the Koopman operator on the learned dictionary. Taking the eigendecomposition $\mathbf{K} = \mathbf{W}_R \boldsymbol{\Lambda} \mathbf{W}_L^*$ allows us to compute the Koopman eigenvalues, eigenfunctions, and modes exactly as we would for EDMD using Eq. 9.15 and Eq. 9.18.

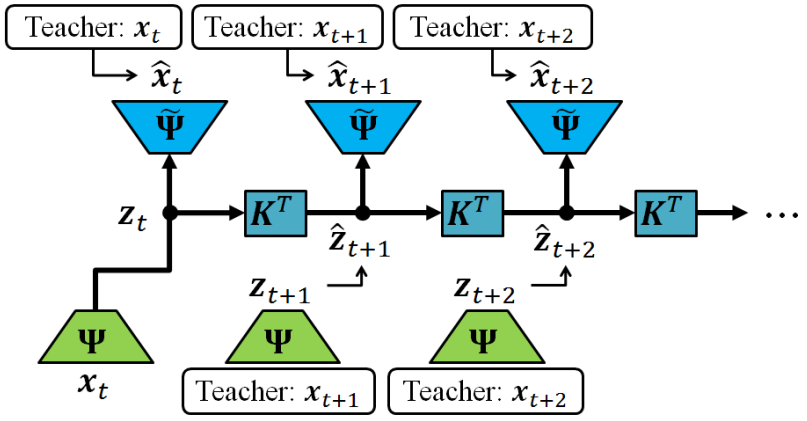


Figure 9.4.1: Linearly-Recurrent Autoencoder Network (LRAN) architecture

By training the operator $\mathbf{K}$ simultaneously with the encoder and decoder networks, the dictionary of observables learned by the encoder is forced to span a low-dimensional Koopman invariant subspace which is sufficiently informative to approximately reconstruct the full state. In many real-world applications, the scientist has access to data sets consisting of several sequential snapshots. The LRAN architecture shown in Figure 9.4.1 takes advantage of longer sequences of snapshots during training. This is especially important when the system dynamics are highly non-normal. In such systems, low-amplitude features which could otherwise be ignored for reconstruction purposes are highly observable and influence the larger amplitude dynamics several time-steps into the future. One may be able to achieve reasonable accuracy on snapshot pairs by neglecting some of these low-energy modes, but accuracy will suffer as more time steps are predicted. Inclusion of multiple time steps where possible forces the LRAN to incorporate these dynamically important non-normal features in the dictionary. As we will discuss later, it is possible to generalize the LRAN architecture to continuous time systems with irregular sampling intervals and sequence lengths. It is also possible to restrict the LRAN to the case when only snapshot pairs are available. Here we consider the case when our data contains equally-spaced snapshot sequences $\{\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+\mathcal{T}-1}\}$ of length $\mathcal{T}$. The loss function

$$J(\boldsymbol{\theta}_{enc}, \boldsymbol{\theta}_{dec}, \boldsymbol{\theta}_{\mathbf{K}}) = \mathop{\mathbb{E}}_{\mathbf{x}_t, \ldots, \mathbf{x}_{t+\mathcal{T}-1} \sim P_{data}} \frac{1}{1+\beta} \left[\sum_{\tau=0}^{\mathcal{T}-1} \frac{\delta^{\tau}}{N_1(\delta)} \frac{\|\hat{\mathbf{x}}_{t+\tau} - \mathbf{x}_{t+\tau}\|^2}{\|\mathbf{x}_{t+\tau}\|^2 + \epsilon_1} + \beta \sum_{\tau=1}^{\mathcal{T}-1} \frac{\delta^{\tau-1}}{N_2(\delta)} \frac{\|\hat{\mathbf{z}}_{t+\tau} - \mathbf{z}_{t+\tau}\|^2}{\|\mathbf{z}_{t+\tau}\|^2 + \epsilon_2} \right] + \Omega(\boldsymbol{\theta}_{enc}, \boldsymbol{\theta}_{dec}, \boldsymbol{\theta}_{\mathbf{K}}) \quad (9.25)$$

is minimized during training, where $\mathbb{E}$ denotes the expectation over the data distribution. The

encoding, latent state dynamics, and decoding processes are given by

$$\mathbf{z}_{t+\tau} = \boldsymbol{\Psi}(\mathbf{x}_{t+\tau}; \boldsymbol{\theta}_{enc}), \quad \hat{\mathbf{z}}_{t+\tau} = \left[\mathbf{K}(\boldsymbol{\theta}_{\mathbf{K}})^{\tau}\right]^{T} \mathbf{z}_t, \quad \hat{\mathbf{x}}_{t+\tau} = \tilde{\boldsymbol{\Psi}}(\hat{\mathbf{z}}_{t+\tau}; \boldsymbol{\theta}_{dec}),$$

respectively. The regularization term Ω has been included for generality, though our numerical experiments show that it was not necessary. Choosing a small dictionary size d provides sufficient regularization for the network. The loss function Eq. 9.25 consists of a weighted average of the reconstruction error and the hidden state time evolution error. The parameter β determines the relative importance of these two terms. The errors themselves are relative square errors between the predictions and the ground truth summed over time with a decaying weight $0 < \delta \leq 1$. This decaying weight is used to facilitate training by prioritizing short term predictions. The corresponding normalizing constants,

$$N_1(\delta) = \sum_{\tau=0}^{\mathcal{T}-1} \delta^{\tau}, \qquad N_2(\delta) = \sum_{\tau=1}^{\mathcal{T}-1} \delta^{\tau-1}$$

ensure that the decay-weighted average is being taken over time. The small constants ϵ_1 and ϵ_2 are used to avoid division by 0 in the case that the ground truth values vanish. The expectation value was estimated empirically using minibatches consisting of sequences of length $\mathcal{T}$ drawn randomly from the training data. Stochastic gradient descent with the Adaptive Moment Estimation (ADAM) method and slowly decaying learning rate was used to simultaneously optimize the parameters $\boldsymbol{\theta}_{enc}$, $\boldsymbol{\theta}_{dec}$, and $\boldsymbol{\theta}_{\mathbf{K}}$ in the open-source software package TensorFlow.

9.4.1 Neural network architecture and initialization

The encoder and decoder consist of deep neural networks whose schematic is sketched in Figure 9.4.2. The figure is only a sketch since many more hidden layers were actually used in the architectures applied to example problems in this paper. In order to achieve sufficient depth in the encoder and decoder networks, hidden layers employed exponential linear units or "elu's" as the nonlinearity [67]. These units mitigate the problem of vanishing and exploding gradients in deep networks by employing the identity function for all non-negative arguments. A shifted exponential function for negative arguments is smoothly matched to the linear section at the origin, giving the activation function

$$\mathrm{elu}(x) = \begin{cases} x & x \geq 0 \\ \exp(x) - 1 & x < 0 \end{cases}. \tag{9.26}$$

This prevents the units from "dying" as standard rectified linear units or "ReLU's" do when the arguments are always negative on the data. Furthermore, "elu's" have the advantage of being continuously differentiable. This will be a nice property if we want to approximate a C^1 data manifold whose chart map and its inverse are given by the encoder and decoder. If the maps are differentiable, then the tangent spaces can be defined as well as push-forward, pull-back, and connection forms. Hidden layers map the activations $\mathbf{x}^{(l)}$ at layer l to activations at the next layer $l+1$ given by a linear transformation and subsequent element-wise application of the activation function,

$$\mathbf{x}^{(l+1)} = \text{elu}\left[\mathbf{W}^{(l)}(\boldsymbol{\theta})\mathbf{x}^{(l)} + \mathbf{b}^{(l)}(\boldsymbol{\theta})\right], \quad \mathbf{W}^{(l)}(\boldsymbol{\theta}) \in \mathbb{R}^{n_{l+1}\times n_l}, \quad \mathbf{b}^{(l)}(\boldsymbol{\theta}) \in \mathbb{R}^{n_{l+1}}. \tag{9.27}$$

The weight matrices $\mathbf{W}$ and vector biases $\mathbf{b}$ parameterized by $\boldsymbol{\theta}$ are learned by the network during training. The output layers L for both the encoder and decoder networks utilize linear transformations without a nonlinear activation function:

$$\mathbf{x}^{(L)} = \mathbf{W}^{(L-1)}(\boldsymbol{\theta})\mathbf{x}^{(L-1)} + \mathbf{b}^{(L-1)}(\boldsymbol{\theta}), \quad \mathbf{W}^{(L-1)}(\boldsymbol{\theta}) \in \mathbb{R}^{n_L\times n_{L-1}}, \quad \mathbf{b}^{(L-1)}(\boldsymbol{\theta}) \in \mathbb{R}^{n_L}, \tag{9.28}$$

where $L = L_{enc}$ or $L = L_{dec}$ is the last layer of the encoder or decoder with $n_{L_{enc}} = d$ or $n_{L_{dec}} = n$ respectively. This allows for smooth and consistent treatment of positive and negative output values without limiting the flow of gradients back through the network.

The weight matrices were initialized using the Xavier initializer in Tensorflow. This initialization distributes the entries in $\mathbf{W}^{(l)}$ uniformly over the interval $[-\alpha, \alpha]$ where $\alpha = \sqrt{6/(n_l + n_{l+1})}$ in order to keep the scale of gradients approximately the same in all layers. This initialization method together with the use of exponential linear units allowed deep networks to be used for the encoder and decoder. The bias vectors $\mathbf{b}^{(l)}$ were initialized to be zero. The transition matrix $\mathbf{K}$ was initialized to have diagonal blocks of the form $\begin{bmatrix} \sigma & \omega \\ -\omega & \sigma \end{bmatrix}$ with eigenvalues $\lambda = \sigma \pm \omega\imath$ equally spaced around the circle of radius $r = \sqrt{\sigma^2 + \omega^2} = 0.8$. This was done heuristically to give the initial eigenvalues good coverage of the unit disc. One could also initialize this matrix using a low-rank DMD matrix.

9.4.2 Simple modifications of LRANs

Several extensions and modifications of the LRAN architecture are possible. Some simple modifications are discussed here, with several more involved extensions suggested in the conclusion. In

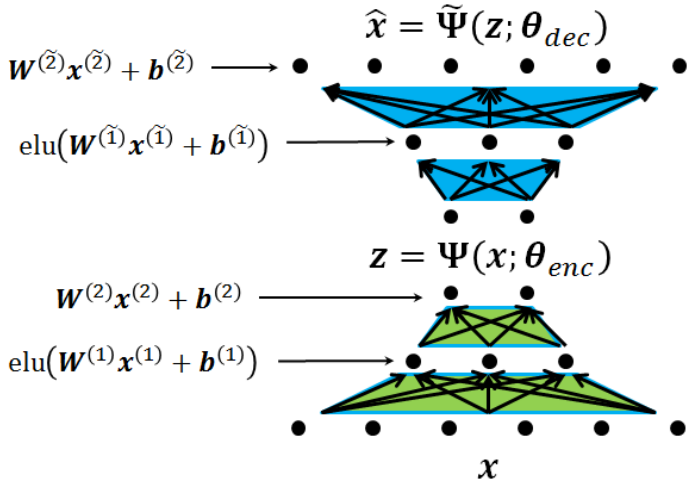


Figure 9.4.2: Architecture of the encoder and decoder networks

the first extension, we observe that it is easy learn Koopman eigenfunctions associated with known eigenvalues simply by fixing the appropriate entries in the matrix $\mathbf{K}$. In particular, if we know that our system has Koopman eigenvalue $\mu = \sigma + \omega\imath$ then we may formulate the state transition matrix

$$\mathbf{K}(\boldsymbol{\theta}) = \begin{bmatrix} \begin{bmatrix} \sigma & \omega \\ -\omega & \sigma \end{bmatrix} & \mathbf{0}_{2\times(d-2)} \\ \mathbf{0}_{(d-2)\times 2} & \tilde{\mathbf{K}}(\boldsymbol{\theta}) \end{bmatrix}. \tag{9.29}$$

In the above, the known eigenvalue is fixed and only the entries of $\tilde{\mathbf{K}}$ are allowed to be trained. If more eigenvalues are known, we simply fix the additional entries of $\mathbf{K}$ in the same way. The case where some eigenvalues are known is particularly interesting because in certain cases, eigenvalues of the linearized system are Koopman eigenvalues whose eigenfunctions have useful properties. Supposing the autonomous system under consideration has a fixed point with all eigenvalues μ_i, $i = 1, \ldots, n$ inside the unit circle, the Hartman-Grobman theorem establishes a topological conjugacy to a linear system with the same eigenvalues in a small neighborhood $\mathcal{U}$ of the fixed point. One easily checks that the coordinate transformations $h_i : \mathcal{M} \cap \mathcal{U} \to \mathbb{R}$, $i = 1, \ldots, n$ establishing this conjugacy are Koopman eigenfunctions restricted to the neighborhood. Composing them with the flow map allows us to extend the eigenfunctions to the entire basin of attraction by defining $\varphi_i(\mathbf{x}) = \mu_i^{-\tau(\mathbf{x})} h_i\left(\mathbf{f}^{\tau(\mathbf{x})}(\mathbf{x})\right)$ where $\tau(\mathbf{x})$ is the smallest integer τ such that $\mathbf{f}^{\tau}(\mathbf{x}) \in \mathcal{U}$. These eigenfunctions extend the topological conjugacy by parameterizing the basin. Similar results hold for stable limit cycles and tori [178]. This is nice because we can often find eigenvalues at fixed points explicitly by linearization. Choosing to fix these eigenvalues in the $\mathbf{K}$ matrix forces the LRAN to learn corresponding eigenfunctions

parameterizing the basin of attraction. It is also easy to include a set of observables explicitly by appending them to the encoder function $\boldsymbol{\Psi}(\mathbf{x};\boldsymbol{\theta}_{enc}) = \left[\boldsymbol{\Psi}_{fixed}(\mathbf{x})^T,\ \tilde{\boldsymbol{\Psi}}(\mathbf{x};\boldsymbol{\theta}_{enc})^T\right]^T$ so that only the functions $\tilde{\boldsymbol{\Psi}}$ are learned by the network. This may be useful if we want to accurately reconstruct some observables $\boldsymbol{\Psi}_{fixed}$ linearly using Koopman modes.

The LRAN architecture and loss function Eq. 9.25 may be further generalized to non-uniform sampling of continuous-time systems. In this case, we consider T sequential snapshots $\{\mathbf{x}(t_0), \mathbf{x}(t_1), \ldots, \mathbf{x}(t_{T-1})\}$ where the times $t_0, t_1, \ldots, t_{T-1}$ are not necessarily evenly spaced. In the continuous time case, we have a Koopman operator semigroup $\mathcal{K}_{t+s} = \mathcal{K}_t\mathcal{K}_s$ defined as $\mathcal{K}_t\psi(\mathbf{x}) = \psi(\Phi_t(\mathbf{x}))$ and generated by the operator $\mathcal{K}\psi(\mathbf{x}) = \dot{\psi}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \cdot \nabla_{\mathbf{x}}\psi(\mathbf{x})$ where the dynamics are given by $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ and Φ_t is the time t flow map. The generator $\mathcal{K}$ is clearly a linear operator which we can approximate on our dictionary of observables with a matrix $\mathbf{K}$. By integrating, we can approximate elements from the semigroup $\mathcal{K}_t$ using the matrices $\mathbf{K}_t = \exp(\mathbf{K}t)$ on the dictionary. Finally, in order to formulate the analogous loss function, we might utilize continuously decaying weights

$$\rho_1(t) = \frac{\delta^t}{\sum_{k=0}^{\mathcal{T}-1}\delta^{t_k}}, \qquad \rho_2(t) = \frac{\delta^t}{\sum_{k=1}^{\mathcal{T}-1}\delta^{t_k}}, \tag{9.30}$$

normalized so that they sum to 1 for the given sampling times. Neural networks can be used for the encoder and decoder together with the loss function

$$J(\boldsymbol{\theta}_{enc}, \boldsymbol{\theta}_{dec}, \boldsymbol{\theta}_{\mathbf{K}}) = \mathop{\mathbb{E}}_{\mathbf{x}(t_0),\ldots,\mathbf{x}(t_{\mathcal{T}-1})\sim P_{data}} \frac{1}{1+\beta}\left[\sum_{k=0}^{\mathcal{T}-1}\rho_1(t_k)\frac{\|\hat{\mathbf{x}}(t_k)-\mathbf{x}(t_k)\|^2}{\|\mathbf{x}(t_k)\|^2+\epsilon_1} + \beta\sum_{k=1}^{\mathcal{T}-1}\rho_2(t_k)\frac{\|\hat{\mathbf{z}}(t_k)-\mathbf{z}(t_k)\|^2}{\|\mathbf{z}(t_k)\|^2+\epsilon_2}\right] + \Omega(\boldsymbol{\theta}_{enc}, \boldsymbol{\theta}_{dec}, \boldsymbol{\theta}_{\mathbf{K}}) \tag{9.31}$$

to be minimized during training. In this case, the dynamics evolve the observables linearly in continuous time, so we let

$$\mathbf{z}(t_k) = \boldsymbol{\Psi}(\mathbf{x}(t_k);\boldsymbol{\theta}_{enc}), \quad \hat{\mathbf{z}}(t_k) = \exp\left[\mathbf{K}(\boldsymbol{\theta}_{\mathbf{K}})(t_k - t_0)\right]^T\mathbf{z}(t_0), \quad \hat{\mathbf{x}}(t_k) = \tilde{\boldsymbol{\Psi}}(\hat{\mathbf{z}}(t_k);\boldsymbol{\theta}_{dec}).$$

This loss function can be evaluated on the training data and minimized in essentially the same way as Eq. 9.25. The only difference is that we are discovering a matrix approximation to the generator of the Koopman semigroup. We will not explore irregularly sampled continuous time systems further in this paper, leaving it as a subject for future work.

We briefly remark that the general LRAN architecture can be restricted to the case of snapshot pairs as shown in Figure 9.4.3. In this special case, training might be accelerated using a technique

similar to Algorithm 4. During the initial stage of training, it may be beneficial to periodically re-initialize the $\mathbf{K}$ matrix with its EDMD approximation using the current dictionary functions and a subset of the training data. This might provide a more suitable initial condition for the matrix as well as accelerate the training process. However, this update for $\mathbf{K}$ is not consistent with all the terms in the loss function J since it does not account for reconstruction errors. Therefore, the final stages of training must always proceed by gradient descent on the complete loss function.

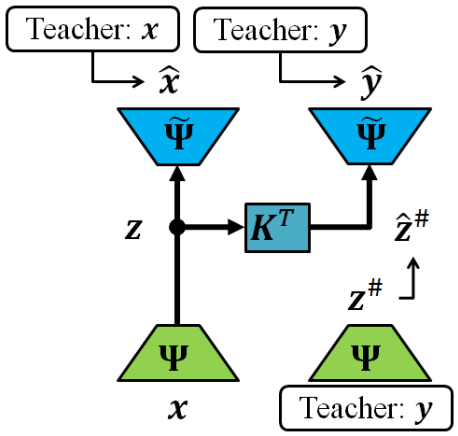


Figure 9.4.3: LRAN architecture restricted to snapshot pairs

Finally, we remark that the LRAN architecture sacrifices linear reconstruction using Koopman modes for nonlinear reconstruction using a decoder neural network in order to learn ultra-low dimensional Koopman invariant subspaces. Interestingly, this formulation allows the LRAN to parameterize the Nonlinear Normal Modes (NNMs) frequently encountered in structural dynamics. These modes are two-dimensional, periodic invariant manifolds containing a fixed point of a Hamiltonian system lacking internal resonances. Therefore, if $\mu = \omega\imath$ and $\overline{\mu} = -\omega\imath$ are a complex conjugate pair of pure imaginary eigenvalues of $\mathbf{K}$ with corresponding left eigenvectors $\mathbf{w}_L$ and $\overline{\mathbf{w}_L}$ then a NNM is parameterized as follows:

$$\mathbf{x}(\alpha) = \tilde{\mathbf{\Psi}}(\alpha\mathbf{w}_L + \overline{\alpha\mathbf{w}_L}; \boldsymbol{\theta}_{dec}) = \tilde{\mathbf{\Psi}}\left(2\Re(\alpha)\Re(\mathbf{w}_L) - 2\Im(\alpha)\Im(\mathbf{w}_L); \boldsymbol{\theta}_{dec}\right). \tag{9.32}$$

The global coordinates on the manifold are $(\Re(\alpha), \Im(\alpha_2))$. Coordinate projection of the full state onto the NNM,

$$\begin{bmatrix} \Re(\alpha) \\ \Im(\alpha) \end{bmatrix} = \begin{bmatrix} \Re(\mathbf{w}_R)^T \\ \Im(\mathbf{w}_R)^T \end{bmatrix} \mathbf{\Psi}(\mathbf{x}; \boldsymbol{\theta}_{enc}), \tag{9.33}$$

is accomplished by employing the encoder network and the right eigenvector $\mathbf{w}_R$ corresponding

to eigenvalue μ. These coordinates are the real and imaginary parts of the associated Koopman eigenfunction $\alpha = \varphi(\mathbf{x})$. The NNM has angular frequency $\angle(\omega\imath)/\Delta t$ where Δt is the sampling interval between the snapshots in the case of the discrete time LRAN.

We may further generalize the notion of NNMs by considering the Koopman mode expansion of the real-valued observable vector $\mathbf{\Psi}$ making up our dictionary. In this particular case, the associated Koopman modes are the complex conjugate left eigenvectors of $\mathbf{K}$. They allow exact reconstruction and prediction using the decomposition

$$\mathbf{z}_t = \sum_{j=1}^{r} \overline{\mathbf{w}_{L,j}} \mu_j^t \left(\mathbf{w}_{R,j}^T \mathbf{z}_0\right) = \sum_{j=1}^{r} \overline{\mathbf{w}_{L,j}} \mu_j^t \varphi_j(\mathbf{x}_0), \tag{9.34}$$

assuming a Koopman invariant subspace has been learned that contains the full state observable. Reconstructing and predicting with the decoder instead, we have

$$\mathbf{x}_t = \tilde{\mathbf{\Psi}} \left[\sum_{j=1}^{r} \overline{\mathbf{w}_{L,j}} \mu_j^t \varphi_j(\mathbf{x}_0);\ \ \boldsymbol{\theta}_{dec} \right]. \tag{9.35}$$

Therefore, each invariant subspace of $\mathbf{K}$ given by its left eigenvectors corresponds to an invariant manifold in the n-dimensional phase space. These manifolds have global charts whose coordinate projections are given by the Koopman eigenfunctions $\varphi_j(\mathbf{x}) = \mathbf{w}_{R,j}^T \mathbf{\Psi}(\mathbf{x}; \boldsymbol{\theta}_{enc})$. The dynamics on these manifolds is incredibly simple and entails repeated multiplication of the coordinates by the eigenvalues. Generalized eigenspaces may also be considered in the natural way by using the Jordan normal form of $\mathbf{K}$ instead of its eigendecomposition in the above arguments. The only necessary change is in the evolution equations, where instead of taking powers of $\mathbf{\Lambda} = \operatorname{diag}\{\mu_1, \ldots, \mu_r\}$, we take powers of $\mathbf{J}$, the matrix consisting of Jordan blocks [178]. Future work might use a variational autoencoder (VAE) formulation [99, 176, 165, 142] where a given distribution is imposed on the latent state in order to facilitate sampling.

9.5 EDMD-based model reduction as a shallow autoencoder

In this section we examine how the EDMD method might be used to construct low-dimensional Koopman invariant subspaces while still allowing for accurate reconstructions and predictions of the full state. The idea is to find a reduced order model of the large linear system identified by EDMD in the space of observables. This method is sometimes called overspecification [228] and essentially determines an encoder function into an appropriate small set of features. From this

reduced set of features, we then employ nonlinear reconstruction of the full state through a learned decoder function. Introduction of the nonlinear decoder should allow for lower-dimensional models to be identified which are still able to make accurate predictions. The proposed framework therefore constructs a kind of autoencoder where encoded features evolve with linear time invariant dynamics. The encoder functions are found explicitly as linear combinations of EDMD observables and are therefore analogous to a shallow neural network with a single hidden layer. The nonlinear decoder function is also found explicitly through a regression process involving linear combinations of basis functions.

We remark that this approach differs from training an LRAN by minimization of Eq. 9.25 in two important ways. First, the EDMD-based model reduction and reconstruction processes are performed sequentially; thus, the parts are not simultaneously optimized as in the LRAN. The LRAN is advantageous since we only learn to encode observables which the decoder can successfully use for reconstruction. There are no such guarantees here. Second, the EDMD dictionary remains fixed albeit overspecified whereas the LRAN explicitly learns an appropriate dictionary. Therefore, the EDMD shallow autoencoder framework will still suffer from the overfitting problem illustrated in Example 9.2.1. If the EDMD-identified matrix $\mathbf{K}$ does not correctly represent the dynamics on a Koopman invariant subspace, then any reduced order models derived from it cannot be expected to be accurately relect the dynamics either. Nonetheless, in many cases, this method could provide a less computationally expensive alternative to training a LRAN which retains some of the benefits owing to nonlinear reconstruction.

Dimensionality reduction is achieved by first performing EDMD with a large dictionary, then projecting the linear dynamics onto a low-dimensional subspace. A naive approach would be to simply project the large feature space system onto the most energetic POD modes — equivalent to low-rank truncation of the SVD $\mathbf{\Psi_X} = \mathbf{U\Sigma V}^*$. While effective for normal systems with a few dominant modes, this approach yields very poor predictions in non-normal systems since low amplitude modes with large impact on the dynamics would be excluded from the model. One method which resolves this issue is balanced truncation of the identified feature space system. Such an idea is suggested in [228] for reducing the system identified by linear DMD. Drawing from the model reduction procedure for snapshot-based realizations developed in [162], we will construct a balanced reduced order model for the system identified using EDMD or KDMD. In the formulation of EDMD that led to the kernel method, an approximation of the Koopman operator,

$$\mathcal{K}\mathbf{\Psi}(\mathbf{x})^*\mathbf{a} = \mathbf{\Psi}(\mathbf{x})^*\mathbf{U}\hat{\mathbf{K}}\mathbf{U}^*\mathbf{a} + \mathbf{r}(\mathbf{x}), \quad \forall \mathbf{a} \in \mathcal{R}(\mathbf{U}), \tag{9.36}$$

was obtained. The approximation allows us to model the dynamics of a vector of observables,

$$\boldsymbol{\Psi}_{\mathbf{U}}(\mathbf{x}) = \mathbf{U}^*\boldsymbol{\Psi}(\mathbf{x}) = \boldsymbol{\Sigma}^+\mathbf{V}^*\boldsymbol{\Psi}_{\mathbf{X}}^*\boldsymbol{\Psi}(\mathbf{x}), \tag{9.37}$$

with the linear input-output system

$$\begin{aligned} \boldsymbol{\Psi}_{\mathbf{U}}(\mathbf{x}_{t+1}) &= \hat{\mathbf{K}}^*\boldsymbol{\Psi}_{\mathbf{U}}(\mathbf{x}_t) + \frac{1}{\sqrt{M}}\boldsymbol{\Sigma}\mathbf{u}_t \\ \mathbf{x}_t &= \mathbf{C}\boldsymbol{\Psi}_{\mathbf{U}}(\mathbf{x}_t) \end{aligned}, \tag{9.38}$$

where $\hat{\mathbf{K}}$ is the matrix Eq. 9.14 identified by EDMD or KDMD. The input $\mathbf{u}_t$ is provided in order to equate varying initial conditions $\boldsymbol{\Psi}_{\mathbf{U}}(\mathbf{x}_0)$ with impulse responses of Eq. 9.38. Since the input is used to set the initial condition, we choose to scale each component by its singular value to reflect the covariance

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim P_{data}}[\boldsymbol{\Psi}_{\mathbf{U}}(\mathbf{x})\boldsymbol{\Psi}_{\mathbf{U}}(\mathbf{x})^*] \approx \frac{1}{M}\mathbf{U}^*\boldsymbol{\Psi}_{\mathbf{X}}\boldsymbol{\Psi}_{\mathbf{X}}^*\mathbf{U} = \frac{1}{M}\boldsymbol{\Sigma}^2 = \mathop{\mathbb{E}}_{\mathbf{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_r)}\left[\frac{1}{M}\boldsymbol{\Sigma}\mathbf{u}\mathbf{u}^*\boldsymbol{\Sigma}^*\right], \tag{9.39}$$

in the observed data. Therefore, initializing the system using impulse responses $\mathbf{u}_0 = \mathbf{e}_j$, $j = 1, \ldots, r$ from the σ-points of the distribution $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r)$ ensures that the correct empirical covariances are obtained. The output matrix,

$$\mathbf{C} = \mathbf{X}\mathbf{V}\boldsymbol{\Sigma}^+, \tag{9.40}$$

is used to linearly reconstruct the full state observable from the complete set of features. It is found using linear regression similar to the Koopman modes Eq. 9.18. The low-dimensional set of observables making up the encoder will be found using a balanced reduced order model of Eq. 9.38.

9.5.1 Balanced Model Reduction

Balanced truncation [182] is a projection-based model reduction techniqe that attempts to retain a subspace in which Eq. 9.38 is both maximally observable and controllable. While these notions generally do not coincide in the original space, remarkably it is possible to find a left-invertible linear transformation $\boldsymbol{\Psi}_{\mathbf{U}}(\mathbf{x}) = \mathbf{T}\mathbf{z}$ under which these properties are balanced. This so called balancing transformation of the learning subspace simultaneously diagonalizes the observability and controllability Gramians. Therefore, the most observable states are also the most controllable and vice versa. The reduced order model is formed by truncating the least observable/controllable states of the transformed system. If the discrete time observability Gramian $\mathbf{W}_o$ and controllability

Gramian $\mathbf{W}_c$ are given by

$$\mathbf{W}_o = \sum_{t=0}^{\infty} (\hat{\mathbf{K}})^t \mathbf{C}^* \mathbf{C} (\hat{\mathbf{K}}^*)^t, \qquad \mathbf{W}_c = \frac{1}{M} \sum_{t=0}^{\infty} (\hat{\mathbf{K}}^*)^t \mathbf{\Sigma}^2 (\hat{\mathbf{K}})^t, \tag{9.41}$$

then the Gramians transform according to

$$\mathbf{W}_o \mapsto \mathbf{T}^* \mathbf{W}_o \mathbf{T}, \qquad \mathbf{W}_c \mapsto \mathbf{T}_L^+ \mathbf{W}_c (\mathbf{T}_L^+)^* \tag{9.42}$$

under the change of coordinates. In the above, $\mathbf{S}^* = \mathbf{T}_L^+$ is the left pseudoinverse satisfying $\mathbf{S}^* \mathbf{T} = \mathbf{I}_d$ where d is the rank of $\mathbf{T}$ and $\mathbf{T}\mathbf{S}^* = \mathbf{P}_{\mathbf{T}}$ is a (not necessarily orthogonal) projection operator onto $\mathcal{R}(\mathbf{T})$.

Since the Gramians are Hermitian positive semidefinite, they can be written as $\mathbf{W}_o = \mathbf{A}\mathbf{A}^*$, $\mathbf{W}_c = \mathbf{B}\mathbf{B}^*$ for some not necessarily unique matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{r \times r}$. Forming an economy sized singular value decomposition $\mathbf{H} = \mathbf{A}^* \mathbf{B} = \mathbf{U}_H \mathbf{\Sigma}_H \mathbf{V}_H$ allows us to construct the transformations

$$\mathbf{T} = \mathbf{B} \mathbf{V}_H (\mathbf{\Sigma}_H^+)^{1/2}, \qquad \mathbf{S} = \mathbf{A} \mathbf{U}_H (\mathbf{\Sigma}_H^+)^{1/2}. \tag{9.43}$$

Using this construction, it is easy to check that the resulting transformation simultaneously diagonalizes the Gramians:

$$\mathbf{T}^* \mathbf{W}_o \mathbf{T} = \mathbf{S}^* \mathbf{W}_c \mathbf{S} = \mathbf{\Sigma}_H. \tag{9.44}$$

Entries of the diagonal matrix $\mathbf{\Sigma}_H$ are called the Hankel singular values. The columns of $\mathbf{T}$ and $\mathbf{S}$ are called the "balancing modes" and "adjoint modes" respectively. The balancing modes span the subspace where Eq. 9.38 is both observable and controllable while the adjoint modes furnish the projected coefficients of the state $\mathbf{S}^* \mathbf{\Psi}_{\mathbf{U}}(\mathbf{x}) = \mathbf{z}$ onto the space where these properties are balanced. The corresponding Hankel singular values quantify the observability/controllability of the states making up $\mathbf{z}$. Therefore, a reduced order model which is provably close to optimal truncation in the $\mathcal{H}_\infty$ norm is formed by rank-d truncation of the SVD, retaining only the first d balancing and adjoint modes $\mathbf{T}_d$ and $\mathbf{S}_d$ [84]. The reduced state space system modeling the dynamics of Eq. 9.38 is given by

$$\begin{aligned} \mathbf{z}_{t+1} &= \mathbf{S}_d^* \hat{\mathbf{K}}^* \mathbf{T}_d \mathbf{z}_t + \frac{1}{\sqrt{M}} \mathbf{S}_d^* \mathbf{\Sigma} \mathbf{u}_t \\ \mathbf{x}_t &\approx \mathbf{C} \mathbf{T}_d \mathbf{z}_t \end{aligned}, \quad \text{where} \quad \mathbf{z} \triangleq \mathbf{S}_d^* \mathbf{\Psi}_{\mathbf{U}}(\mathbf{x}). \tag{9.45}$$

Therefore, the reduced dictionary of observables is given by the components of $\mathbf{S}_d^* \mathbf{\Psi}_{\mathbf{U}} : \mathbb{R}^n \to \mathbb{R}^d$ and the corresponding EDMD approximation of the Koopman operator on its span is given by

$\mathbf{T}_d^*\hat{\mathbf{K}}\mathbf{S}_d$. This set of features is highly observable in that their dynamics strongly influences the full state reconstruction over time through $\mathbf{C}$. And highly controllable in that the features are excited by typical state configurations through $\mathbf{\Sigma}$. The notion of feature excitation corresponding to controllability will be made clear in the next section.

9.5.2 Finite-horizon Gramians and Balanced POD

Typically one would find the infinite horizon Gramians for an overspecified Hurwitz EDMD system Eq. 9.38 by solving the Lyapunov equations

$$\hat{\mathbf{K}}\mathbf{W}_o\hat{\mathbf{K}}^* - \mathbf{W}_o + \mathbf{C}^*\mathbf{C} = 0, \qquad \hat{\mathbf{K}}^*\mathbf{W}_c\hat{\mathbf{K}} - \mathbf{W}_c + \mathbf{\Sigma}^2 = 0. \tag{9.46}$$

In the case of neutrally stable or unstable systems, unique positive definite solutions do not exist and one must use generalized Gramians [300]. When used to form balanced reduced order models, this will always result in the unstable and neutrally stable modes being chosen before the stable modes. This could be problematic for our intended framework since EDMD can identify many spurious and sometimes unstable eigenvalues corresponding to noisy low-amplitude fluctuations. While these noisy modes remain insignificant over finite times of interest, they will dominate EDMD-based predictions over long times. Therefore it makes sense to consider the dominant modes identified by EDMD over a finite time interval of interest. Using finite horizon Gramians reduces the effect of spurious modes on the reduced order model, making it more consistent with the data. The time horizon can be chosen to reflect a desired future prediction time or the number of sequential snapshots in the training data.

The method of Balanced Proper Orthogonal Decomposition or BPOD [225] allows us to find balancing and adjoint modes of the finite horizon system. In BPOD, we observe that the finite-horizon Gramians are empirical covariance matrices formed by evolving the dynamics from impulsive initial conditions for time $\mathcal{T}$. This gives the specific form for matrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{C}^* & \hat{\mathbf{K}}\mathbf{C}^* & \cdots & (\hat{\mathbf{K}})^{\mathcal{T}}\mathbf{C}^* \end{bmatrix}, \qquad \mathbf{B} = \frac{1}{\sqrt{M}} \begin{bmatrix} \mathbf{\Sigma} & \hat{\mathbf{K}}^*\mathbf{\Sigma} & \cdots & (\hat{\mathbf{K}}^*)^{\mathcal{T}}\mathbf{\Sigma} \end{bmatrix}, \tag{9.47}$$

allowing for computation of the balancing and adjoint modes without ever forming the Gramians. This is known as the method of snapshots. Since the output dimension is large, we consider its projection onto the most energetic modes. These are identified by forming the economy sized SVD of the impulse responses $\mathbf{CB} = \mathbf{U}_{OP}\mathbf{\Sigma}_{OP}\mathbf{V}_{OP}^*$. Projecting the output allows us to form the elements

of

$$\mathbf{A}_{OP} = \left[\mathbf{C}^*\mathbf{U}_{OP} \quad \hat{\mathbf{K}}\mathbf{C}^*\mathbf{U}_{OP} \quad \cdots \quad (\hat{\mathbf{K}})^{\mathcal{T}}\mathbf{C}^*\mathbf{U}_{OP}\right], \tag{9.48}$$

from fewer initial conditions than $\mathbf{A}$. In particular, the initial conditions are the first few columns of $\mathbf{U}_{OP}$ with the largest singular values [225].

Observe that the unit impulses place the initial conditions precisely at the σ-points of the data-distribution in features space. If this distribution is Gaussian, then the empirical expectations obtained by evolving the linear system agree with the true expectations taken over the entire data distribution. Therefore, the finite horizon controllability Gramian corresponds to the covariance matrix taken over all time $\mathcal{T}$ trajectories starting at initial data points coming from a Gaussian distribution in feature space. Consequently, controllability in this case corresponds exactly with feature variance or expected square amplitude over time.

We remark that in the infinite-horizon limit $\mathcal{T} \to \infty$, BPOD converges on a transformation which balances the generalized Gramians introduced in [300]. Application of BPOD to unstable systems is discussed in [89] which provides justification for the approach.

Another option to avoid spurious modes from corrupting the long-time dynamics is to consider pre-selection of EDMD modes which are nearly Koopman invariant. The development of such an accuracy criterion for selecting modes is the subject of a forthcoming paper by H. Zhang and C. W. Rowley. One may then apply balanced model reduction to the feature space system containing only the most accurate modes.

9.5.3 Nonlinear reconstruction

In truncating the system, we determined a small subspace of observables whose values evolve linearly in time and are both highly observable and controllable. However, most of the less observable low-amplitude modes are removed. The linear reconstruction only allows us to represent data on a low-dimensional subspace of $\mathbb{R}^n$. While projection onto this subspace aims to explain most of the data variance and dynamics, it may be the case that the data lies near a curved manifold not fully contained in the subspace. The neglected modes contribute to this additional complexity in the shape of the data in $\mathbb{R}^n$. Nonlinear reconstruction of the full state can help account for the complex shape of the data and for neglected modes enslaved to the ones retained.

We consider the regression problem involved in reconstructing the full state $\mathbf{x}$ from a small set of EDMD observables $\mathbf{z}$. Because the previously obtained solution to the EDMD balanced model reduction problem Eq. 9.45 employs linear reconstruction through matrix $\mathbf{C}\mathbf{T}_d$, we expect

nonlinearities in the reconstruction to be small with most of the variance being accounted for by linear terms. Therefore, the regression model,

$$\mathbf{x} = \mathbf{C}_1\mathbf{z} + \mathbf{C}_2\boldsymbol{\Psi}(\mathbf{z}) + \mathbf{e}, \tag{9.49}$$

is formulated based on [87, 289] to include linear and nonlinear components. In the above, $\boldsymbol{\Psi} : \mathbb{C}^d \to \mathcal{H}$ is a nonlinear feature map into reproducing kernel Hilbert space $\mathcal{H}$ and $\mathbf{C}_1 : \mathbb{C}^d \to \mathbb{C}^n$ and $\mathbf{C}_2 : \mathcal{H} \to \mathbb{C}^n$ are linear operators. These operators are found by solving the l^2 regularized optimization problem,

$$\underset{\mathbf{C}_1,\mathbf{C}_2}{\text{minimize}} \;\; J = \|\mathbf{X}^* - \mathbf{Z}^*\mathbf{C}_1^* - \boldsymbol{\Psi}_{\mathbf{Z}}^*\mathbf{C}_2^*\|_F^2 + \gamma \operatorname{Tr}(\mathbf{C}_2\mathbf{C}_2^*), \qquad \gamma \geq 0, \tag{9.50}$$

involving the empirical square error on the training data $\{(\mathbf{z}_j, \mathbf{x}_j)\}_{j=1}^M$ arranged into columns of the matrices $\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \cdots & \mathbf{z}_M \end{bmatrix}$ and $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_M \end{bmatrix}$. The regularization penalty is placed only on the coefficients of nonlinear terms to control over-fitting while the linear term, which we expect to dominate, is not penalized. The operator $\boldsymbol{\Psi}_{\mathbf{Z}} : \mathbb{C}^M \to \mathcal{H}$ forms linear combinations of the data in feature space $\mathbf{v} \mapsto v_1\boldsymbol{\Psi}(\mathbf{z}_1) + \cdots + v_M\boldsymbol{\Psi}(\mathbf{z}_M)$. Since $\mathbf{Z}$ and $\boldsymbol{\Psi}_{\mathbf{Z}}$ are operators with finite ranks r_1 and $r_2 \leq M$, we may consider their economy sized singular value decompositions: $\mathbf{Z} = \mathbf{U}_1\boldsymbol{\Sigma}_1\mathbf{V}_1^*$ and $\boldsymbol{\Psi}_{\mathbf{Z}} = \mathbf{U}_2\boldsymbol{\Sigma}_2\mathbf{V}_2^*$. Observe that it is impossible to infer any components of $\mathcal{R}(\mathbf{C}_1^*)$ orthogonal to $\mathcal{R}(\mathbf{Z})$ since they are annihilated by $\mathbf{Z}^*$. Therefore, we apply Occam's razor and assume that $\mathbf{C}_1^* = \mathbf{U}_1\hat{\mathbf{C}}_1^*$ for some $\hat{\mathbf{C}}_1^* \in \mathbb{C}^{r_1 \times n}$. By the same argument, $\mathcal{R}(\mathbf{C}_2^*)$ cannot have any components orthogonal to $\mathcal{R}(\boldsymbol{\Psi}_{\mathbf{Z}})$ since they are annihilated by $\boldsymbol{\Psi}_{\mathbf{Z}}^*$ and have a positive contribution to the regularization penalty term $\operatorname{Tr}(\mathbf{C}_2\mathbf{C}_2^*)$. Hence, we must also have $\mathbf{C}_2^* = \mathbf{U}_2\hat{\mathbf{C}}_2^*$ for some $\hat{\mathbf{C}}_2^* \in \mathbb{C}^{r_2 \times n}$. Substituting these relationships into Eq. 9.50 allows it to be formulated as the standard least squares problem

$$\begin{aligned} J &= \left\| \mathbf{X}^* - \mathbf{V}_1\boldsymbol{\Sigma}_1\hat{\mathbf{C}}_1^* - \mathbf{V}_2\boldsymbol{\Sigma}_2\hat{\mathbf{C}}_2^* \right\|_F^2 + \gamma \left\| \hat{\mathbf{C}}_2 \right\|_F^2 \\ &= \left\| \begin{bmatrix} \mathbf{X}^* \\ \mathbf{0}_{r_2 \times n} \end{bmatrix} - \begin{bmatrix} \mathbf{V}_1\boldsymbol{\Sigma}_1 & \mathbf{V}_2\boldsymbol{\Sigma}_2 \\ \mathbf{0}_{r_2 \times r_1} & \sqrt{\gamma}\mathbf{I}_{r_2} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{C}}_1^* \\ \hat{\mathbf{C}}_2^* \end{bmatrix} \right\|_F^2 . \end{aligned} \tag{9.51}$$

The block-wise matrix clearly has full column rank $r_1 + r_2$ for $\gamma > 0$ and the normal equation for this least squares problem are found by projecting onto its range. The solution,

$$\begin{bmatrix} \hat{\mathbf{C}}_1^* \\ \hat{\mathbf{C}}_2^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0}_{r_1 \times r_2} \\ \mathbf{0}_{r_2 \times r_1} & \boldsymbol{\Sigma}_2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_{r_1} & \mathbf{V}_1^*\mathbf{V}_2 \\ \mathbf{V}_2^*\mathbf{V}_1 & \mathbf{I}_{r_2} + \gamma\boldsymbol{\Sigma}_2^{-2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{V}_1^*\mathbf{X}^* \\ \mathbf{V}_2^*\mathbf{X}^* \end{bmatrix}, \tag{9.52}$$

corresponds to taking the left pseudoinverse and simplifying the resulting expression. The matrices $\mathbf{V}_{1,2}$ and $\mathbf{\Sigma}_{1,2}$ are found by solving Hermitian eigenvalue problems using the (kernel) matrices of inner products $\mathbf{Z}^*\mathbf{Z} = \mathbf{V}_1\mathbf{\Sigma}_1^2\mathbf{V}_1^*$ and $\mathbf{\Psi}_\mathbf{Z}^*\mathbf{\Psi}_\mathbf{Z} = \mathbf{V}_2\mathbf{\Sigma}_2^2\mathbf{V}_2^*$. At a new point $\mathbf{z}$, the approximate reconstruction using the partially linear kernel regression model is

$$\mathbf{x} \approx \left(\hat{\mathbf{C}}_1\mathbf{\Sigma}_1^{-1}\mathbf{V}_1^*\mathbf{Z}^*\right)\mathbf{z} + \hat{\mathbf{C}}_2\mathbf{\Sigma}_2^{-1}\mathbf{V}_2^*\left(\mathbf{\Psi}_\mathbf{Z}^*\mathbf{\Psi}(\mathbf{z})\right). \tag{9.53}$$

Recall that the kernel matrices are

$$\mathbf{\Psi}_\mathbf{Z}^*\mathbf{\Psi}_\mathbf{Z} = \begin{bmatrix} k(\mathbf{z}_1,\mathbf{z}_1) & \cdots & k(\mathbf{z}_1,\mathbf{z}_M) \\ \vdots & \ddots & \vdots \\ k(\mathbf{z}_M,\mathbf{z}_1) & \cdots & k(\mathbf{z}_M,\mathbf{z}_M) \end{bmatrix}, \qquad \mathbf{\Psi}_\mathbf{Z}^*\mathbf{\Psi}(\mathbf{z}) = \begin{bmatrix} k(\mathbf{z}_1,\mathbf{z}) \\ \vdots \\ k(\mathbf{z}_M,\mathbf{z}) \end{bmatrix} \tag{9.54}$$

for a chosen continuous nonnegative definite mercer kernel function $k : \mathbb{C}^d \times \mathbb{C}^d \to \mathbb{C}$ inducing the feature map $\mathbf{\Psi}$.

The main drawback associated with the kernel method used for encoding eigenfunctions or reconstructing the state is the number of kernel evaluations. Even if the dimension d of the reduced order model is small, the kernel-based inner product of each new example must still be computed with all of the training data in order to encode it and then again to decode it. When the training data sets grow large, this leads to a high cost in making predictions on new data points. An important avenue of future research is to prune the training examples to only a small number of maximally informative "support vectors" for taking inner products. Some possible approaches are discussed in [259, 204].

9.6 Numerical examples

9.6.1 Duffing equation

In our first numerical example, we will consider the unforced Duffing equation in a parameter regime exhibiting two stable spirals. We take this example directly from [280] where the Koopman eigenfunctions are used to separate and parameterize the basins of attraction for the fixed points. The unforced Duffing equation is given by

$$\ddot{x} = -\delta\dot{x} - x(\beta + \alpha x^2), \tag{9.55}$$

where the parameters $\delta = 0.5$, $\beta = -1$, and $\alpha = 1$ are chosen. The equation exhibits stable equilibria at $x = \pm 1$ with eigenvalues $\lambda_{1,2} = \frac{1}{4}\left(-1 \pm \sqrt{31}\imath\right)$ associated with the linearizations at these points. One can show that these (continuous-time) eigenvalues also correspond to Koopman eigenfunctions whose magnitude and complex argument act like action-angle variables parameterizing the entire basins. A non-trivial Koopman eigenfunction with eigenvalue $\lambda_0 = 0$ takes different constant values in each basin, acting like an indicator function to distinguish them.

We will see whether the LRAN and the reduced KDMD model can learn these eigenfunctions from data and use them to predict the dynamics of the unforced Duffing equation as well as to determine which basin of attraction a given point belongs. The training data are generated by simulating the unforced Duffing equation from uniform random initial conditions $(x(0), \dot{x}(0)) \in [-2, 2] \times [-2, 2]$. From each trajectory 11 samples are recorded $\Delta t = 0.25$ apart. The training data for LRAN models consists of $M = 10^4$ such trajectories. Since the KDMD method requires us to evaluate the kernel function between a given example and each training point, we limit the number of training data points to 10^3 randomly selected snapshot pairs from the original set. It is worth mentioning that the LRAN model handles large data sets more efficiently than KDMD since the significant cost goes into training the model which is then inexpensive and fast to evaluate on new examples.

Since three of the Koopman eigenvalues are known ahead of time we train an LRAN model where the transition matrix $\mathbf{K}$ is fixed to have discrete time eigenvalues $\mu_k = \exp(\lambda_k \Delta t)$. We refer to this as the "constrained LRAN" and compare its performance to a "free LRAN" model where $\mathbf{K}$ is learned and a 5th order balanced truncation using KDMD called "KDMD ROM". The hyperparameters of each model are reported in Appendix 9.A.

The learned eigenfunctions for each model are plotted in Figures 9.6.1, 9.6.2, 9.6.3. The corresponding eigenvalues learned or fixed in the model are also reported. The complex eigenfunctions are plotted in terms of their magnitude and phase. In each case, the eigenfunction associated with the continuous-time eigenvalue λ_0 closest to zero appears to partition the phase space into basins of attraction for each fixed point as one would expect. In order to test this hypothesis, we use the median eigenfunction value for each model as a threshold to classify test data points between the basins. The eigenfunction learned by the constrained LRAN was used to correctly classify 0.9274 of the testing data points. The free LRAN eigenfunction and the KDMD balanced reduced order model eigenfunction correctly classified 0.9488 and 0.9650 of the testing data respectively. $M_{\text{test}} = 11 * 10^4$ test data points were used to evaluate the LRAN models, though this number was reduced to a randomly selected $M_{\text{test}} = 1000$ to test the KDMD model due to the exceedingly high computational cost of the kernel evaluations.

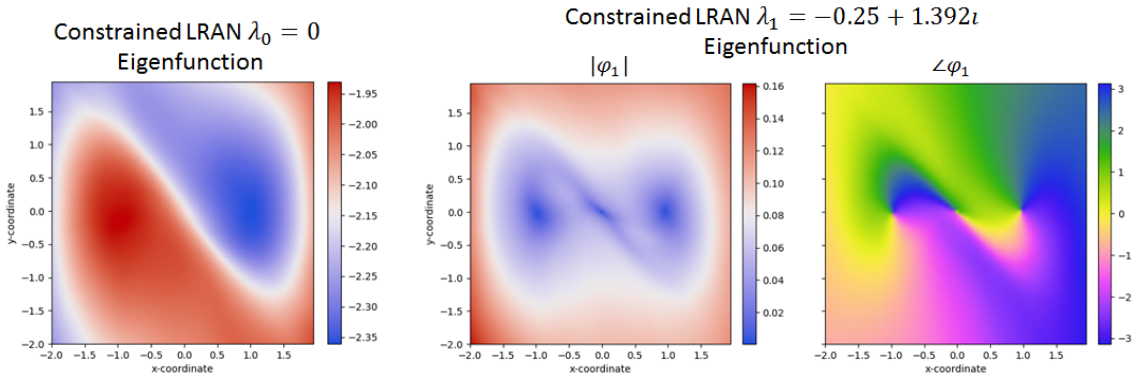


Figure 9.6.1: Unforced Duffing eigenfunctions learned using constrained LRAN

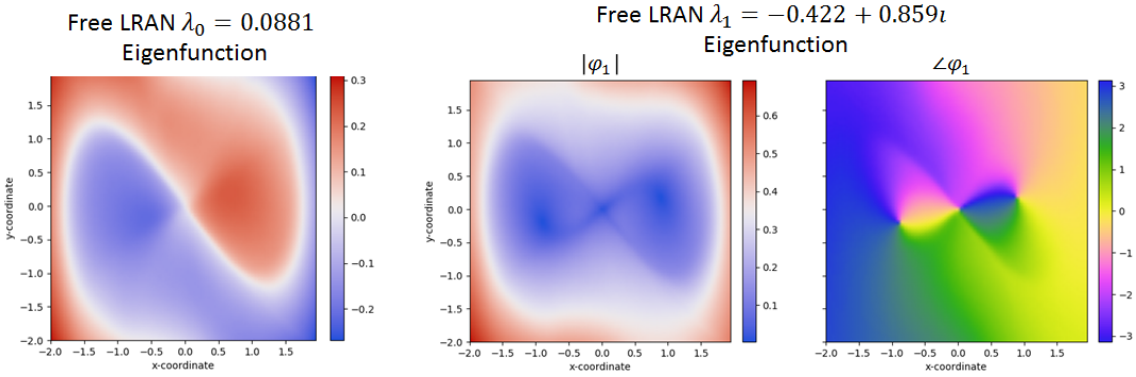


Figure 9.6.2: Unforced Duffing eigenfunctions learned using free LRAN

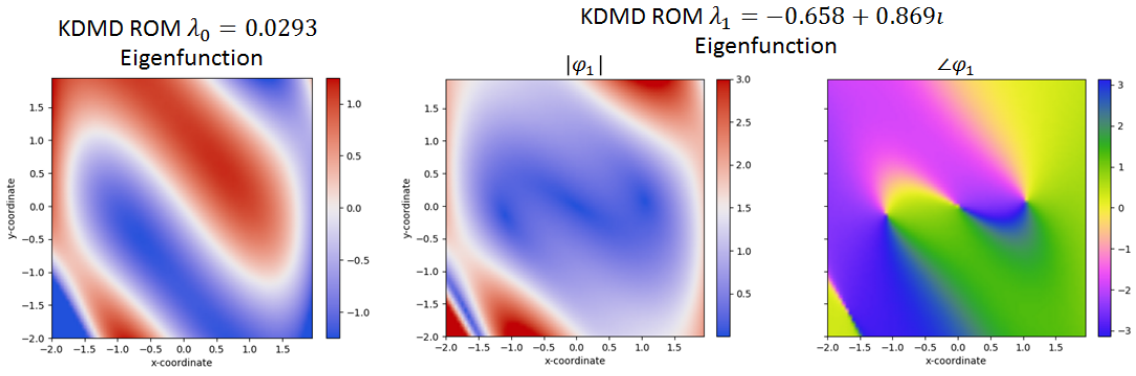


Figure 9.6.3: Unforced Duffing eigenfunctions found using KDMD balanced ROM

The other eigenfunction learned in each case parameterizes the basins of attraction and therefore is used to account for the dynamics in each basin. Each model appears to have learned a similar action-angle parameterization regardless of whether the eigenvalues were specified ahead of time. However, the constrained LRAN shows the best agreement with the true fixed point locations at $x = \pm 1$ where $|\varphi_1| \to 0$. The mean square relative prediction error was evaluated for each model by making predictions on the testing data set at various times in the future. The results plotted in Figure 9.6.4 show that the free LRAN has by far the lowest prediction error likely due to the lack of constraints on the functions it could learn. It is surprising however, that nonlinear reconstruction hurt the performance of the KDMD reduced order model. This illustrates a potential difficulty with this method since the nonlinear part of the reconstruction is prone to over-fit without sufficient regularization.

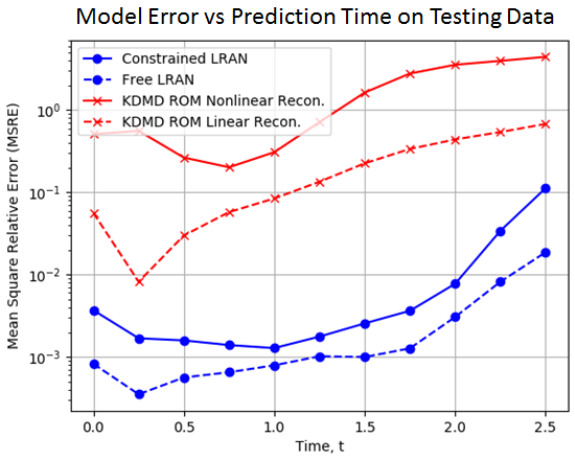
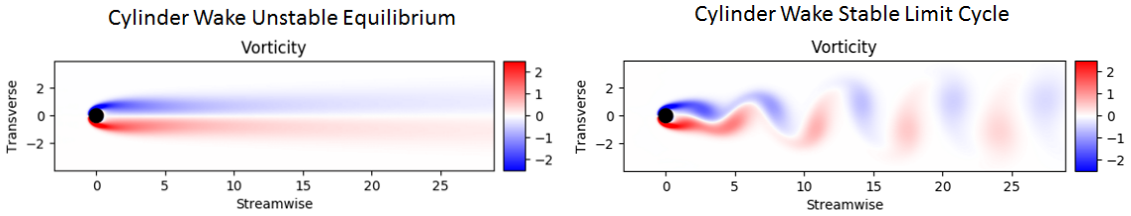


Figure 9.6.4: Unforced Duffing testing data mean square relative prediction errors for each model plotted against the prediction time

9.6.2 Cylinder wake

The next example we consider is the formation of a Kármán vortex sheet downstream of a cylinder in a fluid flow. This problem was chosen since the data has low intrinsic dimensionality due to the simple flow structure but is embedded in high-dimensional snapshots. We are interested in whether the proposed techniques can be used to discover very low dimensional models that accurately predict the dynamics over many time steps. We consider the growth of instabilities near an unstable base flow shown in Figure 9.6.5a at Reynold number $Re = 60$ all the way until a stable limit cycle shown in Figure 9.6.5b is reached. The models will have to learn to make predictions over a range of unsteady flow conditions from the unstable equilibrium to the stable limit cycle.



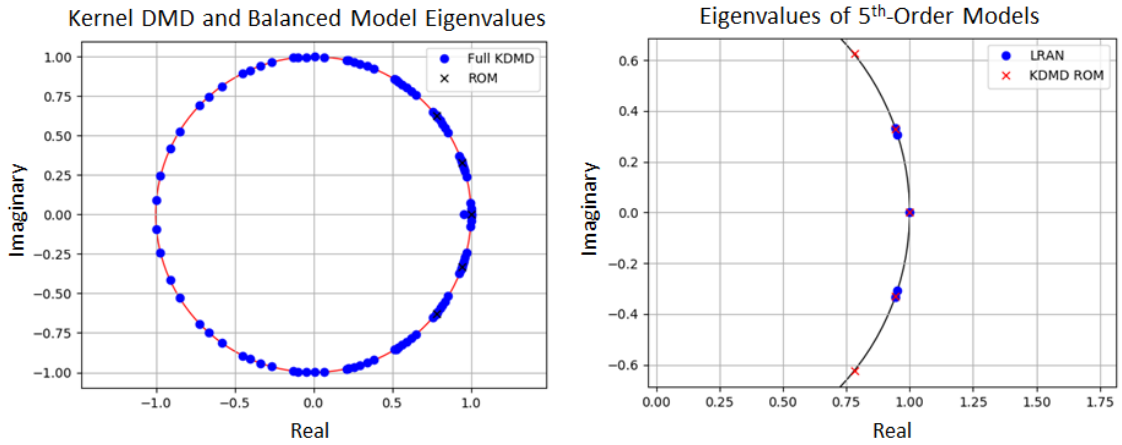
(a) (b)

Figure 9.6.5: Example cylinder wake flow snapshots at the unstable equilibrium and on the stable limit cycle

The raw data consisted of 2000 simulated snapshots of the velocity field taken at time intervals $0.2D/U_\infty$, where D is the cylinder diameter and U_∞ is the free-stream velocity. These data were split into $M_{\text{train}} = 1000$ training, $M_{\text{eval}} = 500$ evaluation, and $M_{\text{test}} = 500$ testing data points. Odd numbered points $\Delta t = 0.4D/U_\infty$ apart were used for training. The remaining 1000 points were divided again into even and odd numbered points $2\Delta t = 0.8D/U_\infty$ apart for evaluation and testing. This enabled training, evaluation, and testing on long data sequences while retaining coverage over the complete trajectory. The continuous-time eigenvalues are found from the discrete-time eigenvalues according to $\lambda = \log(\mu)/\Delta t = \log(\mu)U_\infty/(0.4D)$.

The raw data was projected onto its 200 most energetic POD modes which captured essentially all of the energy in order to reduce the cost of storage and training. 400-dimensional time delay embedded snapshots were formed from the state at time t and $t + \Delta t$. A 5th-order LRAN model and the 5th-order KDMD reduced order model were trained using the hyperparameters in Tables 9.A.3, 9.A.4. In Figure 9.6.6a, many of the discrete-time eigenvalues given by the over-specified KDMD model have approximately neutral stability with some being slightly unstable. However, the finite horizon formulation for balanced truncation allows us to learn the most dynamically salient eigenfunctions over a given length of time, in this case $\mathcal{T} = 20$ steps or $8.0D/U_\infty$. We see in Figure 9.6.6b that three of the eigenvalues learned by the two models are in close agreement and all are approximately neutrally stable.

A side-by-side comparison of the Koopman modes gives some insight into the flow structures whose dynamics the Koopman eigenvalues describe. We notice right away that the Koopman modes in Figure 9.6.8 corresponding to continuous-time eigenvalue λ_1 are very similar for both models and indicate the pattern of vortex shedding downstream. This makes sense since a single frequency and mode will account for most of the amplitude as the limit cycle is approached. Evidently both models discover these limiting periodic dynamics. For the KDMD ROM, λ_2 is almost exactly the higher harmonic $2 * \lambda_1$. The corresponding Koopman mode in Figure 9.6.9 also reflects smaller flow



(a) (b)

Figure 9.6.6: Discrete-time Koopman eigenvalues approximated by the KDMD ROM and the LRAN

structures which oscillate at twice the frequency of λ_1. Interestingly, the LRAN does not learn the same second eigenvalue as the KDMD ROM. The LRAN continuous-time eigenvalue λ_2 is very close to λ_1 which suggest that these frequencies might team up to produce the low-frequency $\lambda_1 - \lambda_2$. The second LRAN Koopman mode in Figure 9.6.9 also bears qualitative resemblance to the first Koopman mode in Figure 9.6.8, but with a slightly narrower pattern in the y-direction. The LRAN may be using the information at these frequencies to capture some of the slower transition process from the unstable fixed point to the limit cycle. The Koopman modes corresponding to $\lambda_0 = 0$ are also qualitatively different indicating that the LRAN and KDMD ROM are extracting different constant features from the data. We must be careful in our interpretation, however, since the LRAN's koopman modes are only a least squares approximations to the nonlinear reconstruction process performed by the decoder.

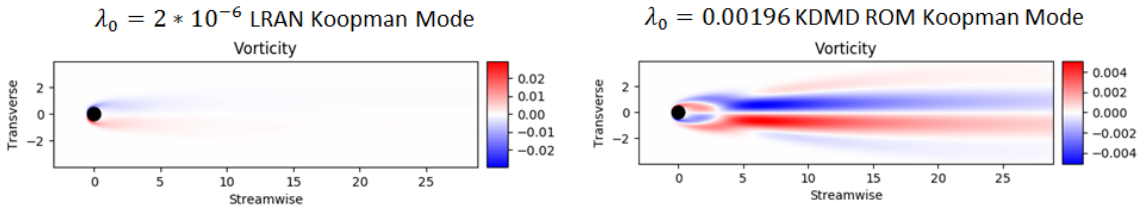


Figure 9.6.7: LRAN and KDMD ROM Koopman modes associated with $\lambda_0 \approx 0$

Plotting the model prediction error Figure 9.6.10 shows that the linear reconstructions using both models have comparable performance with errors growing slowly over time. Therefore, the choice of the second Koopman mode does not seem to play a large role in the reconstruction process. However,

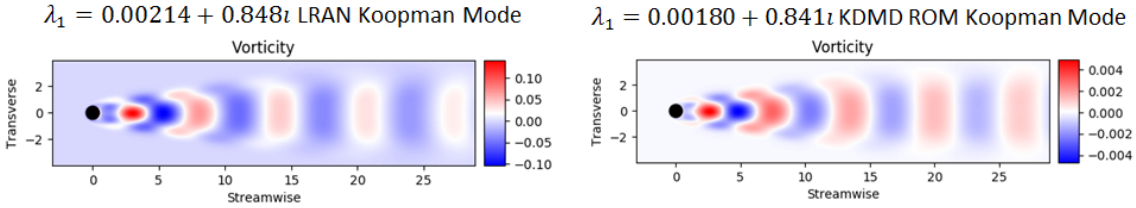


Figure 9.6.8: LRAN and KDMD ROM Koopman modes associated with $\lambda_1 \approx 0.002 + 0.845\imath$

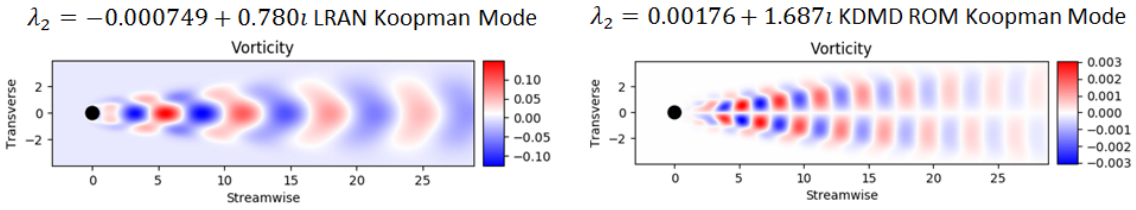


Figure 9.6.9: LRAN and KDMD ROM Koopman modes associated with λ_2 which differs greatly between the models

when the nonlinear decoder is used to reconstruct the LRAN predictions, the mean relative error is roughly an order of magnitude smaller than the nonlinearly reconstructed KDMD ROM over many time steps. The LRAN has evidently learned an advantageous nonlinear transformation for reconstructing the data using the features evolving according to λ_2. The second Koopman mode reflects a linear approximation of this nonlinear transformation in the least squares sense.

Another remark is that nonlinear reconstruction using the KDMD ROM did significantly improve the accuracy in this example. This indicates that many of the complex variations in the data are really enslaved to a small number of modes. This makes sense since the dynamics are periodic on the limit cycle. Finally, it is worth mentioning that the prediction accuracy was achieved on average over all portions of the trajectory from the unstable equilibrium to the limit cycle. Both models therefore have demonstrated predictive accuracy and validity over a wide range of qualitatively different flow conditions. The nonlinearly reconstructed LRAN achieves a constant low prediction error over the entire time interval used for training $\mathcal{T}\Delta t = 8.0D/U_\infty$. The error only begins to grow outside the interval used for training. The high prediction accuracy could likely be extended by training on longer data sequences.

9.6.3 Kuramoto-Sivashinsky equation

We now move on to test our new techniques on a very challenging problem — the Kuramoto-Sivashinsky equation in a parameter regime just beyond the onset of chaos. Since any chaotic

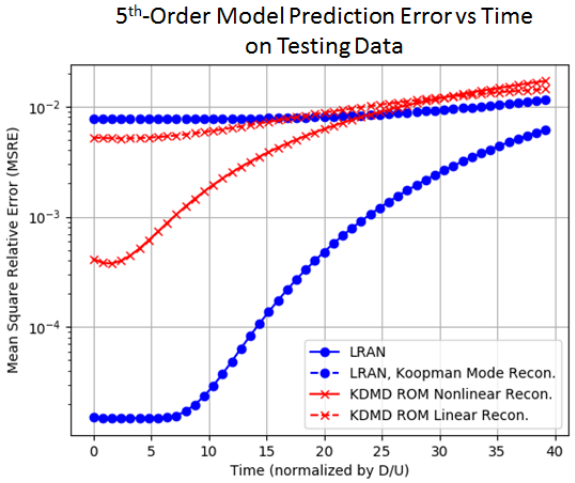


Figure 9.6.10: Cylinder wake testing data mean square relative prediction errors for each model plotted against the prediction time

dynamical system is mixing, it only has trivial Koopman eigenfunctions on its attractor(s). We therefore cannot expect our model to accurately reflect the dynamics of the real system. Rather, we aim to make predictions using very low-dimensional models that are accurate over short times and plausible over longer times.

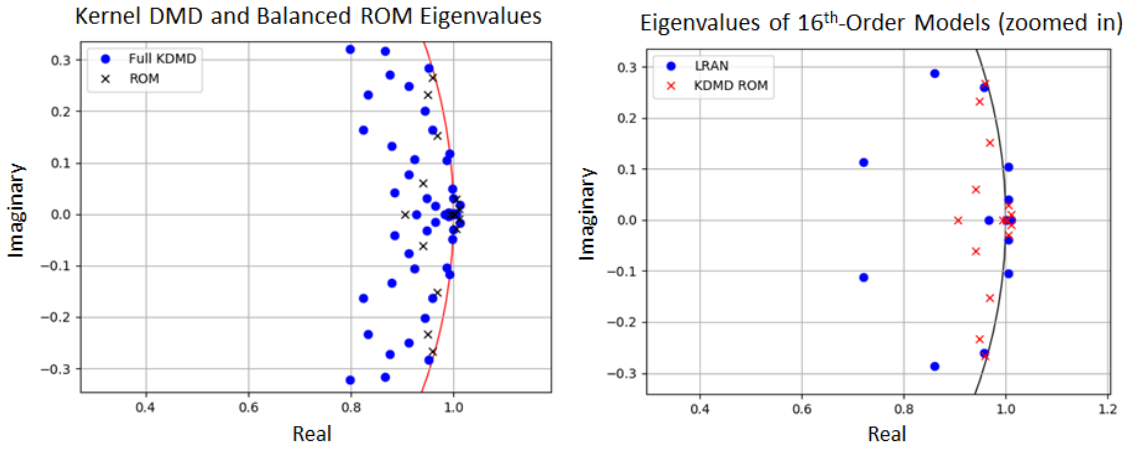
The data was generated by performing direct numerical simulations of the Kuramoto-Sivashinsky equation,

$$u_t + u_{xx} + u_{xxxx} + uu_x = 0, \qquad x \in [0, L], \tag{9.56}$$

using a semi-implicit Fourier pseudo-spectral method. The length $L = 8\pi$ was chosen where the equation first begins to exhibit chaotic dynamics [114]. 128 Fourier modes were used to resolve all of the dissipative scales. Each data set: training, evaluation, and test, consisted of 20 simulations from different initial conditions each with 500 recorded states spaced by $\Delta t = 1.0$. Snapshots consisted of time delay embedded states at t and $t + \Delta t$. The initial conditions were formed by suppling Gaussian random perturbations to the coefficients on the 3 linearly unstable Fourier modes $0 < 2\pi k/L < 1 \implies k = 1, 2, 3$.

An LRAN as well as a KDMD balanced ROM were trained to make predictions over a time horizon $\mathcal{T} = 5$ steps using only $d = 16$ dimensional models. Model parameters are given in Tables 9.A.5, 9.A.6. The learned approximate Koopman eigenvalues are plotted in Figure 9.6.11. We notice that there are some slightly unstable eigenvalues, which makes sense since there are certainly unstable modes including the three linearly unstable Fourier modes. Additionally, Figure 9.6.11b shows that some of the eigenvalues with large magnitude learned by the LRAN and the KDMD

ROM are in near agreement.



(a) (b)

Figure 9.6.11: Discrete-time Koopman eigenvalues for the Kuramoto-Sivashinksy equation approximated by the KDMD ROM and the LRAN

The plot of mean square relative prediction error on the testing data set Figure 9.6.12 indicates that our addition of nonlinear reconstruction from the low dimensional KDMD ROM state does not change the accuracy of the reconstruction. The performance of the KDMD ROM and the LRAN are comparable with the LRAN showing a modest reduction in error over all prediction times. It is interesting to note that the LRAN does not produce accurate reconstructions using the regression-based Koopman modes. In this example, the LRAN's nonlinear decoder is essential for the reconstruction process. Evidently, the dictionary functions learned by the encoder require nonlinearity to reconstruct the state. Again, both models are most accurate over the specified time horizon $\mathcal{T} = 5$ used for training.

Plotting some examples in Figure 9.6.13 of ground truth and predicted test data sequences illustrates the behavior of the models. These examples show that both the LRAN and the KDMD ROM make quantitatively accurate short term predictions. While the predictions after $t \approx 5$ lose their accuracy as one would expect when trying to make linear approximations of chaotic dynamics, they remain qualitatively plausible. The LRAN model in particular is able to model and predict grouping and merging events between traveling waves in the solution. For example in Figure 9.6.13a the LRAN successfully predicts the merging of two wave crests (in red) taking place between $t = 2$ and $t = 5$. The LRAN also predicts the meeting of a peak and trough in Figure 9.6.13b at $t = 5$. These results are encouraging considering the substantial reduction in dimensionality from a time delay embedded state of dimension 256 to a 16-dimensional encoded state having linear time

evolution.

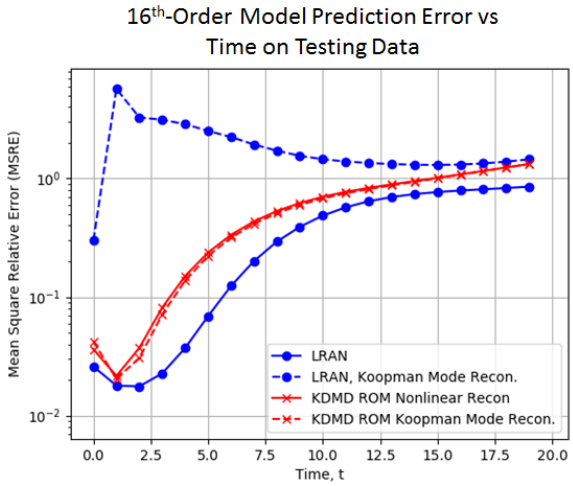
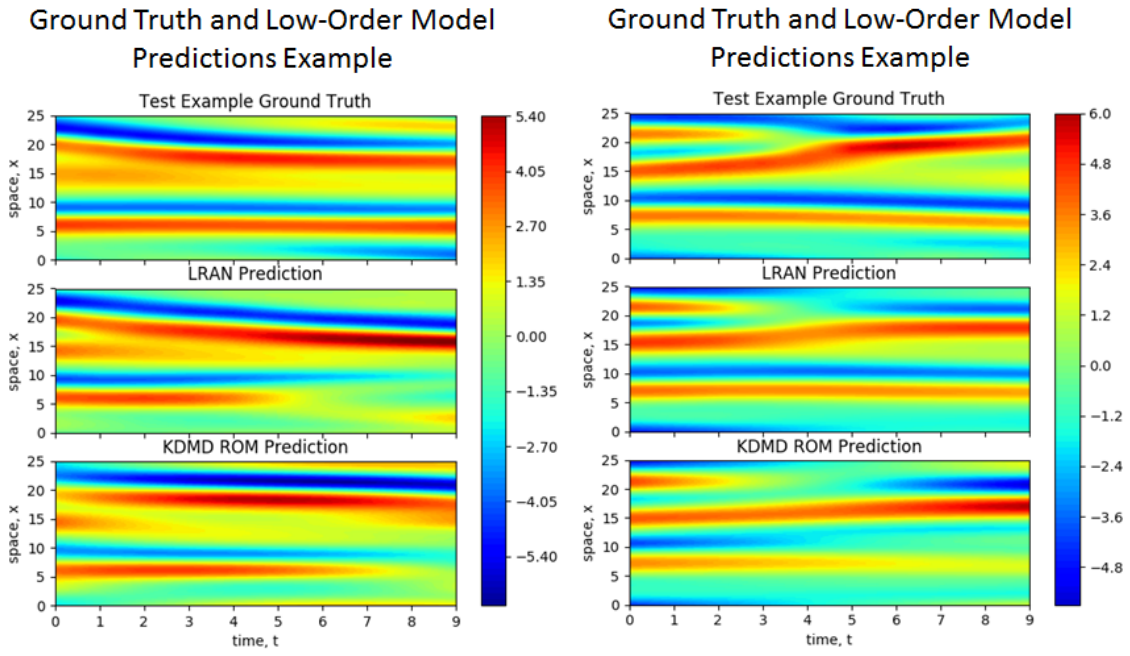


Figure 9.6.12: Kuramoto-Sivashinsky testing data mean square relative prediction errors for each model plotted against the prediction time

9.7 Conclusions

We have illustrated some fundamental challenges with EDMD, in particular highlighting the trade-off between rich dictionaries and over-fitting. The use of adaptive, low-dimensional dictionaries avoids the over-fitting problem while retaining enough capacity to represent Koopman eigenfunctions of complicated systems. This motivates the use of neural networks to learn sets of dictionary observables that are well-adapted to the given problems. By relaxing the constraint that the models must produce linear reconstructions of the state via the Koopman modes, we introduce a decoder neural network enabling the formation of very low-order models utilizing richer observables. Finally, by combining the neural network architecture that is essentially an autoencoder with linear recurrence, the LRAN learns features that are dynamically important rather than just energetic (i.e., large in norm).

Discovering a small set of dynamically important features or states is also the idea behind balanced model reduction. This led us to investigate the identification of low-dimensional models by balanced truncation of over-specified EDMD models, and in particular, KDMD models. Nonlinear reconstruction using a partially linear multi-kernel method was investigated for improving the reconstruction accuracy of the KDMD ROMs from very low-dimensional spaces. Our examples show that in some cases like the cylinder wake example, it can greatly improve the accuracy. We think this is because the data is intrinsically low-dimensional, but curves in such a way as to extend in many dimensions of the embedding space. The limiting case of the cylinder flow is an extreme example



(a) (b)

Figure 9.6.13: LRAN and KDMD ROM model predictions on Kuramoto-Sivashinsky test data examples

where the data becomes one-dimensional on the limit cycle. In some other cases, however, nonlinear reconstruction does not help, is sensitive to parameter choices, or decreases the accuracy due to over-fitting.

Our numerical examples indicate that unfolding the linear recurrence for many steps can improve the accuracy of LRAN predictions especially within the time horizon used during training. This is observed in the error versus prediction time plots in our examples: the error remains low and relatively flat for predictions made inside the training time horizon $\mathcal{T}$. The error then grows for predictions exceeding this length of time. However, for more complicated systems like the Kuramoto-Sivashinsky equation, one cannot unfold the network for too many steps before additional dimensions must be added to retain accuracy of the linear model approximation over time. These observations are also approximately true of the finite-horizon BPOD formulation used to create approximate balanced truncations of KDMD models. One additional consideration in forming balanced reduced-order models from finite-horizon impulse responses of over-specified systems is the problem of spurious eigenvalues whose associated modes only become significant for approximations as $t \to \infty$. The use of carefully chosen finite time horizons allows us to pick features which are the most relevant (observable and excitable) over any time span of interest.

The main drawback of the balanced reduced-order KDMD models becomes evident when making predictions on new evaluation and testing examples. While the LRAN has a high "up-front" cost to train — typically requiring hundreds of thousands of iterations — the cost of evaluating a new example is almost negligible and so very many predictions can be made quickly. On the other hand, every new example whose dynamics we want to predict using the KDMD reduced order model must still have its inner product evaluated with every training data point. Pruning methods like those developed for multi-output least squares support vector machines (LS-SVMs) will be needed to reduce the number of kernel evaluations before KDMD reduced order models can be considered practical for the purpose of predicting dynamics. The same will need to be done for kernel-based nonlinear reconstruction methods.

We conclude by discussing some exciting directions for future work on LRANs. The story certainly does not end with using them to learn linear encoded state dynamics and approximations of the Koopman eigenfunctions. The idea of establishing a possibly complicated transformation into and out of a space where the dynamics are simple is an underlying theme of this work. Human understanding seems to at least partly reside in finding isomorphism. If we can establish that a seemingly complicated system is topologically conjugate to a simple system, it doesn't really matter what the transformation is as long as we can compute it. In this vein, the LRAN architecture is perfectly suited to learning the complicated transformations into and out of a space where the dynamics equations have a simple normal form. For example, this could be accomplished by learning coefficients on homogeneous polynomials of varying degree in addition to a matrix for updating the encoded state. One could also include learned parameter dependencies in order to study bifurcation behavior. Furthermore, any kind of symmetry could be imposed through either the normal form itself or through the neural network's topology. For example, convolutional neural networks can be used in cases where the system is statistically stationary in a spatial coordinate.

Introduction of control terms to the dynamics of the encoded state is another interesting direction for inquiry. In some cases it might be possible to introduce a second autoencoder to perform a state-dependent encoding and decoding of the control inputs at each time step. Depending on the form that the inputs take in the evolution of the encoded state, it may be possible to apply a range of techniques from modern state-space control to nonlinear optimal control or even reinforcement learning-based control with policy and value functions parameterized by neural networks.

Another natural question is how the LRAN framework can be adapted to handle stochastic dynamics and uncertainty. Recent work in the development of structured inference networks for nonlinear stochastic models [142] may offer a promising approach. The low-dimensional dynamics

and reconstruction process could be generalized to nonlinear stochastic processes for generating full state trajectories. Since the inference problem for such nonlinear systems is intractable, the encoder becomes a (bi-directional) recurrent neural network for performing approximate inference on the latent state given our data in analogy with Kalman smoothing. In this manner, many plausible outputs can be generated to estimate the distribution over state trajectories in addition to an inference distribution for quantifying uncertainty about the low-dimensional latent state.

Furthermore, with the above formulation of generative autoencoder networks for dynamics, it might be possible to employ adversarial training [100] in a similar manner to the adversarial autoencoder [165]. Training the generative network used for reconstruction against a discriminator network will encourage the generator to produce more plausible details like turbulent eddies in fluid flows which are not easily distinguished from the real thing.

Acknowledgments

We would like to thank Scott Dawson for providing us with the data from his cylinder wake simulations. We would also like to thank William Eggert for his invaluable help and collaboration on the initial iterations of the LRAN code.

Appendix

9.A Hyperparameters used to train models

The same hyperparameters in Table 9.A.1 were used to train the constrained and free LRANs in the unforced Duffing equation example.

Table 9.A.1: Constrained LRAN hyperparameters for unforced Duffing example

Parameter	**Value(s)**
Time-delays embedded in a snapshot	1
Encoder layer widths (left to right)	2, 32, 32, 16, 16, 8, 3
Decoder layer widths (left to right)	3, 8, 16, 16, 32, 32, 2
Snapshot sequence length, $\mathcal{T}$	10
Weight decay rate, δ	0.8
Relative weight on encoded state, β	1.0
Minibatch size	50 examples
Initial learning rate	10^{-3}
Geometric learning rate decay factor	0.01 per $4 * 10^5$ steps
Number of training steps	$4 * 10^5$

Table 9.A.2 summarizes the hyperparameters used to train the KDMD Reduced Order Model for the unforced Duffing example.

Table 9.A.2: KDMD ROM hyperparameters for unforced Duffing example

Parameter	**Value(s)**
Time-delays embedded in a snapshot	1
EDMD Dictionary kernel function	Gaussian RBF, $\sigma = 10.0$
KDMD SVD rank, r	27
BPOD time horizon, $\mathcal{T}$	10
BPOD output projection rank	2 (no projection)
Balanced model order, d	3
Nonlinear reconstruction kernel function	Gaussian RBF, $\sigma = 10.0$
Multi-kernel linear part truncation rank, r_1	3
Multi-kernel nonlinear part truncation rank, r_2	8
Multi-kernel regularization constant, γ	10^{-4}

The hyperparameters used to train the LRAN model on the cylinder wake data are given in Table 9.A.3.

Table 9.A.3: LRAN hyperparameters for cylinder wake example

Parameter	**Value(s)**
Time-delays embedded in a snapshot	2
Encoder layer widths (left to right)	400, 100, 50, 20, 10, 5
Decoder layer widths (left to right)	5, 10, 20, 50, 100, 400
Snapshot sequence length, $\mathcal{T}$	20
Weight decay rate, δ	0.95
Relative weight on encoded state, β	1.0
Minibatch size	50 examples
Initial learning rate	10^{-3}
Geometric learning rate decay factor	0.01 per $2 * 10^5$ steps
Number of training steps	$2 * 10^5$

Table 9.A.4 summarizes the hyperparameters used to train the KDMD Reduced Order Model on the cylinder wake data.

Table 9.A.5 lists the hyperparameters used to train the LRAN model on the Kuramoto-Sivashinsky equation example.

Table 9.A.6 summarizes the hyperparameters used to train the KDMD Reduced Order Model on the Kuramoto-Sivashinsky equation example data.

Table 9.A.4: KDMD ROM hyperparameters for cylinder wake example

Parameter	**Value(s)**
Time-delays embedded in a snapshot	2
EDMD Dictionary kernel function	Gaussian RBF, $\sigma = 10.0$
KDMD SVD rank, r	100
BPOD time horizon, $\mathcal{T}$	20
BPOD output projection rank	100
Balanced model order, d	5
Nonlinear reconstruction kernel function	Gaussian RBF, $\sigma = 10.0$
Multi-kernel linear part truncation rank, r_1	5
Multi-kernel nonlinear part truncation rank, r_2	15
Multi-kernel regularization constant, γ	10^{-8}

Table 9.A.5: LRAN hyperparameters for Kuramoto-Sivashinsky example

Parameter	**Value(s)**
Time-delays embedded in a snapshot	2
Encoder layer widths (left to right)	256, 32, 32, 16, 16
Decoder layer widths (left to right)	16, 16, 32, 32, 256
Snapshot sequence length, $\mathcal{T}$	5
Weight decay rate, δ	0.9
Relative weight on encoded state, β	1.0
Minibatch size	50 examples
Initial learning rate	10^{-3}
Geometric learning rate decay factor	0.1 per $2 * 10^5$ steps
Number of training steps	$4 * 10^5$

Table 9.A.6: KDMD ROM hyperparameters for Kuramoto-Sivashinsky example

Parameter	**Value(s)**
Time-delays embedded in a snapshot	2
EDMD Dictionary kernel function	Gaussian RBF, $\sigma = 10.0$
KDMD SVD rank, r	60
BPOD time horizon, $\mathcal{T}$	5
BPOD output projection rank	60
Balanced model order, d	16
Nonlinear reconstruction kernel function	Gaussian RBF, $\sigma = 100.0$
Multi-kernel linear part truncation rank, r_1	16
Multi-kernel nonlinear part truncation rank, r_2	60
Multi-kernel regularization constant, γ	10^{-7}

Chapter 10

Inadequacy of Linear Methods for Minimal Sensor Placement and Feature Selection in Nonlinear Systems; a New Approach Using Secants

Samuel E. Otto and Clarence W. Rowley

Sensor placement and feature selection are critical steps in engineering, modeling, and data science that share a common mathematical theme: the selected measurements should enable solution of an inverse problem. Most real-world systems of interest are nonlinear, yet the majority of available techniques for feature selection and sensor placement rely on assumptions of linearity or simple statistical models. We show that when these assumptions are violated, standard techniques can lead to costly over-sensing without guaranteeing that the desired information can be recovered from the measurements. In order to remedy these problems, we introduce a novel data-driven approach for sensor placement and feature selection for a general type of nonlinear inverse problem based on the information contained in secant vectors between data points. Using the secant-based approach, we develop three efficient greedy algorithms that each provide different types of robust, near-minimal

reconstruction guarantees. We demonstrate them on two problems where linear techniques consistently fail: sensor placement to reconstruct a fluid flow formed by a complicated shock-mixing layer interaction and selecting fundamental manifold learning coordinates on a torus.

10.1 Introduction

Reconstructing the state of complex systems like fluid flows, chemical processes, and biological networks from measurements taken by a few carefully chosen sensors is a crucial task for controlling, forecasting, and building simplified models of these systems. In this setting it is important to be able to reconstruct the relevant information about the system using the smallest total number of measurements which includes minimizing the number of sensors to reduce cost, and using the shortest possible measurement histories to shorten response time. Feature selection in statistics and machine learning is a related task where one tries to find a small subset of measured variables (features) in the available data that allow one to reliably predict a quantity of interest.

Nonlinear reconstruction can yield large improvements over linear reconstruction when the sensors or features are carefully selected [107]. Successful nonlinear reconstruction techniques include neural networks [183],[184], deep nonlinear state estimators [115], [142], and convex ℓ^1 minimization to reveal sparse coefficients in learned libraries [284], [43]. The need for nonlinear representation and reconstruction is also recognized in the reduced-order-modeling community where it is called "nonlinear Galerkin" approximation [150], [219], [168]. These methods are necessary because in many systems of interest, the state is found to lie near a low-dimensional underlying manifold that is curved in such a way that it is not contained in any low-dimensional subspace [191]. We will show that the best possible linear reconstruction accuracy is fundamentally limited by the number of measurements (features) and the fraction of the variance that is captured in the principal subspace [116] of that dimension. In essence, any linear representation in a subspace is "too loose" and demands an excessive number of measurements to even have a hope of accurately reconstructing the state using linear functions. Nonlinear reconstruction is much more powerful, as Whitney's celebrated embedding theorem (Theorem. 5, [278]) shows that states on any r-dimensional smooth manifold can be reconstructed using $2r$ carefully chosen measurements. If the measurements must be linear functions of the state on a compact sub-manifold of $\mathbb{R}^n$ then almost any $2r+1$ dimensional projection will provide an embedding [277] — although some embeddings may be better than others in terms of their robustness to disturbances.

With many measurements available from our sensors (though not necessarily ones that achieve Whitney's results), the problem that remains is to properly choose them so that nonlinear reconstruction is possible and robust to noise. While nonlinear reconstruction has proved to be extremely advantageous, the overwhelming majority of sensor placement and feature selection methods rely on measures of linear or Gaussian reconstruction accuracy as an optimization criterion. Such methods include techniques based on sampling modal bases [295], [166], [60], [82], [41], linear dynamical system models [181], [79], [252], [253], [267], [299] Bayesian and maximum likelihood optimality in linear inverse problems [56], [125], [247], information-theoretic criteria under Gaussian or other simple statistical models [141], [55], [54], [248], [245], and sparse linear approximation in dictionaries using LASSO [262], [296] or orthogonal matching pursuit [198], [265]. We provide an overview of a representative collection of these methods that we shall use as a basis for comparison in Section 10.2.

We show that relying on these linear, Gaussian techniques to identify sensors that will be used for nonlinear reconstruction can lead to costly over-sensing when the underlying manifold is low-dimensional, but the data do not lie in an equally low-dimensional subspace. This effect is most pronounced when the most energetic (highest variance) components of the data are actually functions of less-energetic components, but not vice versa. In such cases, the linear techniques are consistently "tricked" into sensing the most energetic components while failing to capture the important less energetic ones that can actually be used for minimal reconstruction. These situations are not merely academic, and they actually abound in physics and in data science. As we shall discuss in Section 10.3, the problem appears in mixing layer fluid flows and in the presence of shock waves, which are both ubiquitous in aerodynamics. The presence of important low-energy sub-harmonic frequencies is also generic behavior after a period-doubling bifurcation, which is a common route to chaos, for instance in ecosystem collapse [268] and cardiac arrhythmia [213]. In data science, the problem is most pronounced when we try to select fundamental nonlinear embedding coordinates for a data set using manifold learning techniques like kernel PCA [244], Laplacian eigenmaps [15], diffusion maps [68], and Isomap [260] as we shall discuss in Section 10.3.3.

In order to address the limitations of linear, Gaussian methods for sensor placement and feature selection demonstrated in the first half of the paper, we develop a novel data-driven approach based on consideration of secant vectors between states in Section 10.4. Related secant-based approaches have been pioneered by [31], [121], [111], [254] for the purpose of finding optimal embedding subspaces. While their considerations of secants lead to continuous optimization problems over subspaces, our considerations of secants lead to combinatorial optimization problems over sets of sensors. We develop three different secant-based objectives together with greedy algorithms that each provide

different types of robust, near-minimal reconstruction guarantees for very general types of nonlinear inverse problems. The guarantees stem from the underlying geometric information that is captured by secants and encoded in our optimization objectives. Moreover, the objectives we consider each have the celebrated diminishing returns property called *submodularity*, allowing us to leverage the classical results of G. L. Nemhauser and L. A. Wolsey et al. [187], [283] to guarantee the performance of efficient greedy algorithms for sensor placement. We also leverage concentration of measure results in order to prove performance guarantees when the secants are randomly down-sampled, enabling computational scalability to very large data sets. Each of these techniques demonstrates greatly improved performance compared to a large collection of linear techniques on a canonical shock-mixing layer flow problem [292] as well as for selecting fundamental manifold learning coordinates.

10.2 Background on Linear, Gaussian, Techniques

The predominant sensor placement, feature selection, and experimental design techniques available today rely on linear and/or Gaussian assumptions about the underlying data: that is, that the data live in a low-dimensional subspace and/or have a Gaussian distribution. Under these assumptions, it becomes easy to quantify the performance of sensors, features, or experiments, using a variety of information theoretic, Bayesian, maximum likelihood, or other optimization criteria. A comprehensive review is beyond the scope of this paper, and of course we do not claim that linear methods always fail. Rather, we argue that because the underlying linear, Gaussian assumptions are violated in many real-world problems, we cannot expect them to find small collections of sensors that enable nonlinear reconstruction of the desired quantities. We shall briefly review the collection of linear techniques that we shall compare to throughout this work and that we think are representative of the current literature.

10.2.1 (Group) LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) method introduced by R. Tibshirani [262] is a highly successful technique for feature selection in machine learning that has found additional applications in compressive sampling recovery [46] and system identification [36]. A generalization by M. Yuan and Y. Lin [296] called group LASSO is especially relevant for sensor placement since it allows measurements to be selected in groups that might come from the same sensor at different instants of time. Suppose we are given a collection of data consisting of available measurements $\boldsymbol{m}_j(\boldsymbol{x}_i)$, $j = 1, \ldots, M$ along with relevant quantities $\boldsymbol{g}(\boldsymbol{x}_i)$ that we wish to reconstruct

over a collection of states $\boldsymbol{x}_i$, $i = 1, \ldots, N$. The group LASSO convex optimization problem takes the form

$$\underset{\boldsymbol{A}_1,\ldots,\boldsymbol{A}_M}{\text{minimize}} \ \sum_{i=1}^{N} \left\| \boldsymbol{g}(\boldsymbol{x}_i) - \sum_{j=1}^{M} \boldsymbol{A}_j \boldsymbol{m}_j(\boldsymbol{x}_i) \right\|_2^2 + \gamma \sum_{j=1}^{M} \|\boldsymbol{A}_j\|_F \tag{10.1}$$

and tries to reconstruct the targets as accurately as possible using a linear combination of the measurements subject to a sparsity-promoting penalty. The strength of the penalty depends on the user-specified parameter $\gamma \geq 0$ and forces the coefficient matrices $\boldsymbol{A}_j$ on many of the measurement groups to be identically zero. Those coefficient matrices with nonzero entries indicate the sensors that should be used to *linearly* reconstruct the target variables with high accuracy.

10.2.2 Determinantal "D"-Optimal Selection

Suppose the state $\boldsymbol{x}$ has a prior probability distribution with covariance $\boldsymbol{C}_{\boldsymbol{x}}$ and the target variables $\boldsymbol{g}(\boldsymbol{x})$ and measurements $\boldsymbol{m}_j(\boldsymbol{x})$, $j = 1, \ldots, M$ are linear functions of the state

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{T}\boldsymbol{x}, \qquad \boldsymbol{m}_j(\boldsymbol{x}) = \boldsymbol{M}_j\boldsymbol{x} + \boldsymbol{n}_j \tag{10.2}$$

where $\boldsymbol{n}_j$ is the mean-zero, state independent, noise from the jth sensor with covariance $\boldsymbol{C}_{\boldsymbol{n}_j}$. Then, if $\boldsymbol{M}_{\mathcal{S}}$ is a matrix with rows given by $\boldsymbol{M}_j$ and $\boldsymbol{C}_{\boldsymbol{n}_{\mathcal{S}}}$ is a block diagonal matrix formed from $\boldsymbol{C}_{\boldsymbol{n}_j}$, for j in a given set of sensors $\mathcal{S}$, then the optimum (least-squares) linear estimate of $\boldsymbol{g}(\boldsymbol{x})$ given $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})$ has error covariance

$$\boldsymbol{C}_{\boldsymbol{e}}(\mathcal{S}) = \boldsymbol{T}\left(\boldsymbol{C}_{\boldsymbol{x}}^{-1} + \boldsymbol{M}_{\mathcal{S}}^T \boldsymbol{C}_{\boldsymbol{n}_{\mathcal{S}}}^{-1} \boldsymbol{M}_{\mathcal{S}}\right)^{-1} \boldsymbol{T}^T. \tag{10.3}$$

If $\boldsymbol{x}$ and the noise are independent Gaussian random variables then Eq. **10.3** is the covariance of the posterior distribution for $\boldsymbol{g}(\boldsymbol{x})$ given $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})$. A low-dimensional representation of the state and its covariance are usually found from data via principal component analysis (PCA) [116] or proper orthogonal decomposition (POD) [23] when an analytical model is not available.

A common technique, referred to as the Bayesian approach in the optimal design of experiments [212] is to quantify performance using functions of $\boldsymbol{C}_{\boldsymbol{e}}(\mathcal{S})$ [56]. In particular, Bayesian determinantal or "D"-optimality entails minimizing $\log\det \boldsymbol{C}_{\boldsymbol{e}}(\mathcal{S})$, which, under Gaussian assumptions, is equivalent to minimizing the conditional entropy [248], [245] or the volumes of confidenece ellipsoids about the maximum a posteriori (MAP) estimate of $\boldsymbol{g}(\boldsymbol{x})$ given $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})$ [125]. This approach is widely used for sensor placement since it readily admits efficient approximations based on convex relaxation [125] and greedy algorithms [247], [267] with guaranteed performance. Similar objectives have been used to quantify observability and controllability for sensor and actuator placement in linear dynamical

systems [252], [253].

When there is no prior probability distribution for $\boldsymbol{x}$ and we want to estimate the full state $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$ from measurements corrupted by Gaussian noise, we can construct the maximum likelihood estimate whose error covariance is

$$\boldsymbol{C_e}(\mathcal{S}) = \left(\boldsymbol{M}_{\mathcal{S}}^T \boldsymbol{C}_{\boldsymbol{n}_{\mathcal{S}}}^{-1} \boldsymbol{M}_{\mathcal{S}}\right)^{-1}. \tag{10.4}$$

Minimizing the volumes of confidence ellipsoids in this setting as is done in [125] is referred to as maximum likelihood "D"-optimality since it entails maximizing $\log\det\left(\boldsymbol{M}_{\mathcal{S}}^T \boldsymbol{C}_{\boldsymbol{n}_{\mathcal{S}}}^{-1} \boldsymbol{M}_{\mathcal{S}}\right)$. In the absence of the regularizing effect the prior distribution has on the estimate, we must have at least as many sensor measurements as state variables in the maximum likelihood setting.

10.2.3 Pivoted QR Factorization

Pivoted matrix factorization techniques, and QR pivoting in particular, have become a popular choice for sensor placement [166], [35] and feature selection in reduced-order modeling [60], [82], where the method is often referred to as the Discrete Empirical Interpolation Method (DEIM). This approach dates back to P. Businger and G. H. Golub's seminal work [41], which introduced Householder-pivoted QR factorization for the purpose of feature selection in least squares fitting problems. The approach is also closely related to orthogonal matching pursuit [198] and simultaneous orthogonal matching pursuit [265], which are widely used sparse approximation algorithms.

In its simplest form, one supposes that the underlying state to be estimated $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$ is low dimensional (e.g., using its PCA or POD coordinate representation) and selects the linear measurements from among the rows of a matrix $\boldsymbol{M}$ by forming a pivoted QR decomposition of the form

$$\boldsymbol{M}^T \left[\begin{array}{c|c} \boldsymbol{P}_1 & \boldsymbol{P}_2 \end{array}\right] = \boldsymbol{Q} \left[\begin{array}{c|c} \boldsymbol{R}_1 & \boldsymbol{R}_2 \end{array}\right], \tag{10.5}$$

where $\left[\begin{array}{c|c} \boldsymbol{P}_1 & \boldsymbol{P}_2 \end{array}\right]$ is a permutation matrix. The first $K = \dim \boldsymbol{x}$ pivot columns forming the submatrix $\boldsymbol{P}_1$ determine a set of sensor measurements $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) = \boldsymbol{M}_{\mathcal{S}}\boldsymbol{x} = \boldsymbol{P}_1^T \boldsymbol{M}\boldsymbol{x}$ from which $\boldsymbol{x}$ can be robustly recovered as

$$\boldsymbol{x} = \left(\boldsymbol{P}_1^T \boldsymbol{M}\right)^{-1} \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) = \boldsymbol{Q} \left(\boldsymbol{R}_1^T\right)^{-1} \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}). \tag{10.6}$$

This approach is successful because at each step of the QR pivoting process, the measurement that maximizes the corresponding diagonal entry of the upper triangular matrix $\boldsymbol{R}_1$ is selected. The

resulting large diagonal entries of $\boldsymbol{R}_1$ mean that measurement errors are not strongly amplified by the linear reconstruction map $\boldsymbol{Q}\left(\boldsymbol{R}_1^T\right)^{-1}$.

10.3 Problems with Linear Techniques

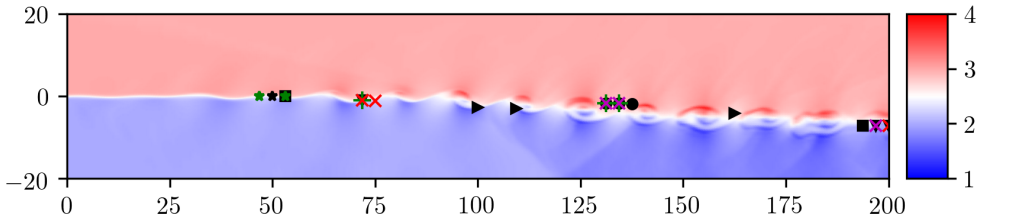
In this section, we illustrate the problems with employing linear state reconstruction and sensor placement techniques for nonlinear systems and data sets by means of an example. We consider the shock-mixing layer interaction proposed by Yee et al. [292], which has become a canonical problem for studying jet noise production as well as high-order numerical methods. This problem captures many key elements of shock wave-turbulent boundary layer interactions that, according to S. Priebe and M. P. Martín [209] "occur in many external and internal compressible flow applications such as transonic aerofoils, high-speed engine inlets, internal flowpaths of scramjets, over-expanded rocket engine nozzles and deflected control surfaces or any other discontinuities in the surface geometry of high-speed vehicles." The resulting pressure and heat transfer fluctuations can be large, so it is important to monitor the state of these flows to ensure the safety of a vehicle.

Our goal will be to choose a small number of sensor locations in this flow at which to measure either the horizontal, u, or vertical, v, velocity component in order to reconstruct the entire velocity field. A snapshot of these velocity fields from the fully-developed flow computed using the high-fidelity local WENO-type characteristic filtering method of S.-C. Lo et al. [159] is shown in Fig. 10.3.1. While the flow is very nearly periodic, and hence lives near a one-dimensional loop in state space, the complicated physics arising from the interaction of the oblique shock with vortices in the spatially-evolving mixing layer results in data that do not lie near any low-dimensional subspace. In addition to being high dimensional, this flow exhibits the low-frequency unsteadiness characteristic of shock wave–turbulent boundary layer interactions [209], [66], [210] and of spatial mixing layer flows in general [113].

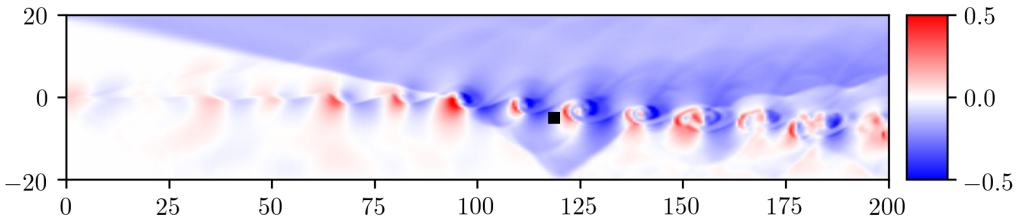
10.3.1 The Need for Nonlinear Reconstruction

Linear reconstruction is fundamentally confined to a subspace whose dimension is at most equal to the total number of sensor measurements. Hence the fraction of the variance that linear reconstruction can capture using d measurements is at most the fraction of the variance along the first d principal components: in particular, the coefficient of determination is bounded by

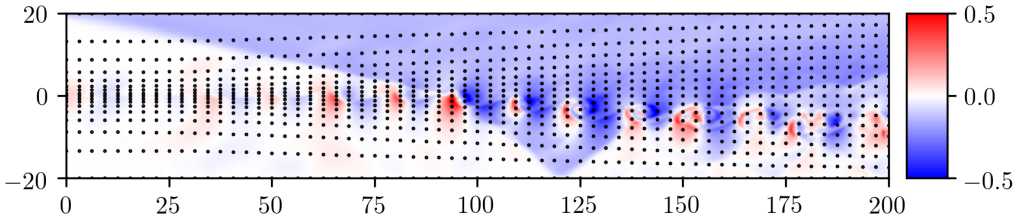
$$R^2 \leq \frac{\sigma_1^2 + \cdots + \sigma_d^2}{\sigma_1^2 + \cdots + \sigma_n^2}. \tag{10.7}$$



(a) stream-wise u velocity component



(b) transverse v velocity component



(c) available sensor locations

Figure 10.3.1: A snapshot of the u and v velocity components in the shock mixing-layer flow is shown in (a) and (b) along with the sensors selected using various methods from among the two components at 1105 available locations shown in (c). These methods include LASSO with PCA (black o), LASSO with Isomap (red x) greedy Bayes D-optimality (magenta x), convex Bayes D-optimality (black >), convex D-optimality for modes 3 and 4 (black v), QR pivoting (green +), and secant-based techniques using detectable differences (#1, #2: green star, #3: black star) and the amplification threshold method (black square).

Examining the fraction of the variance captured by the leading principal subspaces in Figure 10.3.2a leads us to the rather disappointing conclusion that in order to capture 90% of the variance in the shock-mixing layer flow via linear reconstruction, we need at least 11 independent measurements, and to capture 98% we need at least 33.

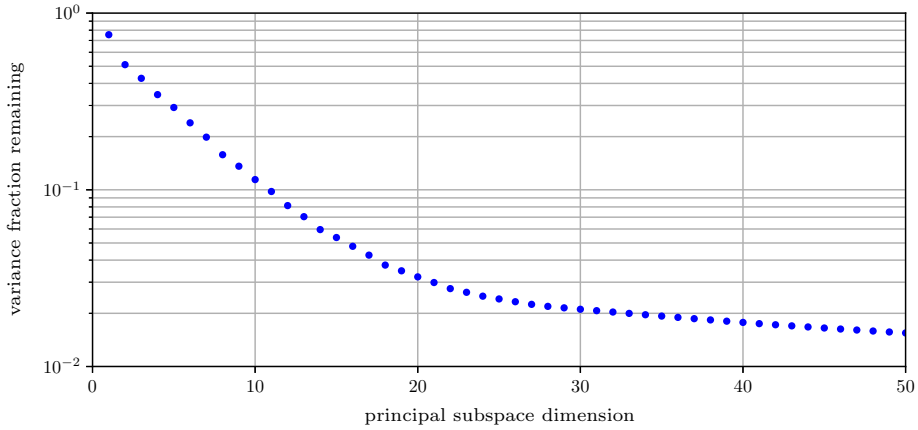
The best possible linear reconstruction performance can be arbitrarily poor even though the underlying manifold is low-dimensional. We illustrate this fact with the following toy model that resembles the phase dependence of principal components in the shock-mixing layer problem shown in Figures 10.3.2b and 10.3.2c. Let θ be uniformly distributed over the interval $[0, 2\pi]$ and let the components of the state vector have sinusoidal dependence on the phase given by

$$x_{2k-1} = \sqrt{2}\cos(k\theta), \quad x_{2k} = \sqrt{2}\sin(k\theta), \quad k = 1, \ldots, n/2. \tag{10.8}$$

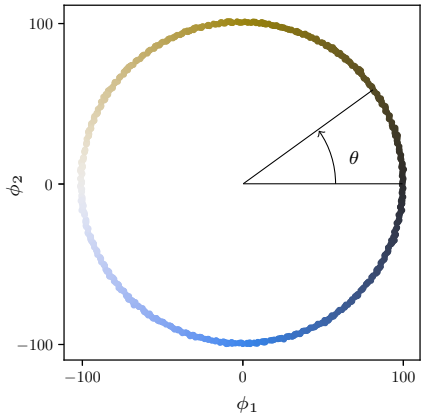
Since these components are orthonormal functions of θ with respect to the uniform probability measure on $[0, 2\pi]$, the state vector has isotropic covariance, $\mathbb{E}\boldsymbol{x}\boldsymbol{x}^T = \boldsymbol{I}_n$, and the fraction of the variance captured by the leading d principal components is d/n. As the dimension increases, the highest possible coefficient of determination for linear reconstruction approaches zero since $R^2 \leq d/n \to 0$ as $n \to \infty$. Meanwhile, it's obvious that the state vector can be perfectly reconstructed as a nonlinear function of x_1 and x_2 alone.

Indeed, it is possible to reconstruct the entire shock-mixing layer flow-field as a nonlinear function of the velocity measurements at two carefully chosen locations. In particular, the measurements made at the locations marked by the two green stars in Figure 10.3.1 are one-to-one with the phase and hence the state of the flow. This is seen in Figure 10.3.4a, where the phase angle (color) — hence the full state — can be determined uniquely from the values of the measurements. Meanwhile, the best possible linear reconstruction performance using two measurements is $R^2 < 0.5$.

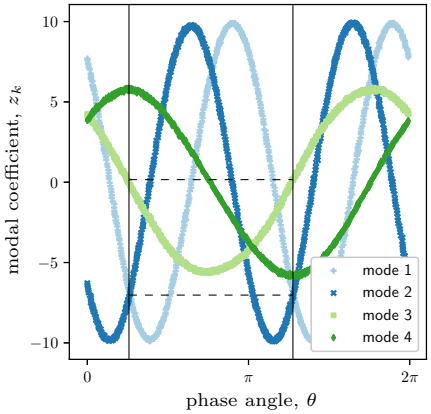
In practice, many nonlinear reconstruction techniques are available including neural networks [183], Gaussian process regression [216], and recurrent neural networks for time-delayed measurements [142]. Using Gaussian process regression and the two sensor locations marked by green stars in Figure 10.3.1, we obtain near-perfect, robust reconstruction of the leading 100 principal components. The resulting reconstruction accuracy for the flow-fields on a held-out set of 250 snapshots is $R^2 = 0.986$. While embedding the state in a subspace of moderately high dimension, as we have done here using 100 principal components, is an essentially unavoidable computational step, the dimension of this subspace need not determine the number of sensor measurements used by the reconstruction as it would for a linear reconstruction technique. In particular, we have nonlinearly



(a) variance orthogonal to principal subspaces



(b) Isomap coordinates



(c) PCA coefficients

Figure 10.3.2: The linear and nonlinear dimension reduction techniques PCA (a.k.a POD) and Isomap are applied to the shock-mixing layer data. (a) shows the remaining fraction of the total variance orthogonal to each leading principal subspace. (b) plots the data in the leading two Isomap embedding coordinates, revealing that it lies very near a loop in state space. (c) shows how the leading principal components (modal coefficients) vary with the phase angle around the loop. The black vertical lines reveal distinct points where the leading three principal components are identical.

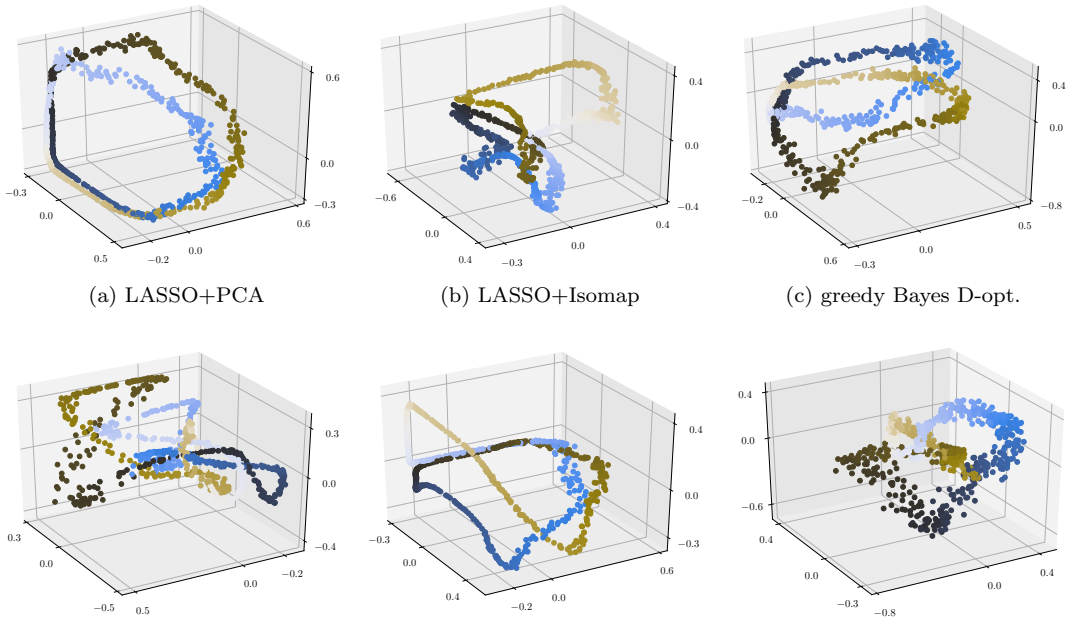
reconstructed the 100 principal components from only two measurements.

10.3.2 The Need for Nonlinear Sensor Placement

With such poor reconstruction afforded by linear techniques, we cannot expect sensor placement methods based on them to perform any better. This is not to say that a practitioner won't ever find lucky sensor locations for nonlinear reconstruction by employing a sensor placement technique that maximizes linear reconstruction accuracy. However, this kind of luck is not guaranteed as illustrated when we apply state of the art linear sensor placement techniques to the shock mixing-layer problem. Indeed Figures 10.3.3a, 10.3.3b, 10.3.3c, 10.3.3d, and 10.3.3e provide visual proof that three sensors chosen using LASSO to reconstruct the leading 100 principal components, LASSO to reconstruct the leading two Isomap coordinates, the greedy Bayes D-optimality approach, the convex Bayes D-optimality approach, and pivoted QR factorization do not produce measurements that are one-to-one with the state. Implementation details can be found in Appendix 10.A. In each case, there are at least two distinct states with different phases on the orbit (color) for which the sensors measure the same values and hence cannot be used to tell them apart.

Even measuring the leading three principal components directly, which are optimal for linear reconstruction, cannot always reveal the state of the shock-mixing layer flow. The black vertical lines in Figure 10.3.2c indicate the phases of two distinct states for which the leading three principal components agree, yet the fourth differs. One may wonder whether the fact that the third and fourth principal components are one-to-one with the state can be leveraged for sensor placement. Even our attempt to place three maximum likelihood D-optimal sensors using the convex optimization approach of [125] to reconstruct the third and fourth principal components fails to produce measurement that can recover the phase of the flow as seen in Figure 10.3.3f.

Despite the failure of linear techniques to find three adequate sensor locations, we have already seen that it is possible to nonlinearly reconstruct the state of the shock-mixing layer flow using the two sensors marked by green stars in Figure 10.3.1. The measurements from these sensors are shown in Figure 10.3.4a, where it is clear that they are one-to-one with the state. It is important to note, however, that the derivative of these measurements (as a linear map of tangent vectors) is not one-to-one. In particular, we have circled two cusps in Figure 10.3.4a where the time derivatives of the measurements vanish on the curve near which the data lie, but the time derivatives of system's states do not vanish. This makes it impossible to reconstruct the time derivatives of the states from the time derivatives of the measurements at the cusps. These cusps pose a problem if we are interested



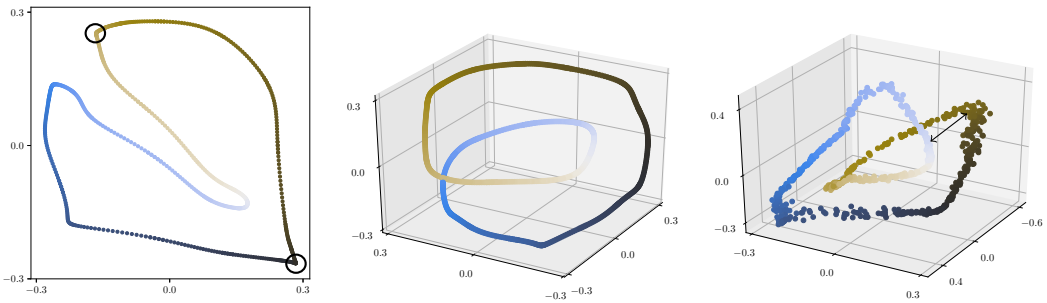
(a) LASSO+PCA (b) LASSO+Isomap (c) greedy Bayes D-opt.

(d) convex Bayes D-opt. (e) pivoted QR (f) convex M.L. D-opt., modes $3, 4$

Figure 10.3.3: these plots show the measurements made by sensors selected using various linear methods on the shock-mixing layer flow problem. Each dot indicates the values measured by the sensors and its color indicates the phase of the corresponding flowfield. Each set of sensors make identical or nearly identical measurements on distinct flowfields, indicated by overlapping points with different colors. These sensors cannot tell those flowfields apart since the measurements are the same.

in constructing a reduced-order model of the system in the measurement space because any such model will have a spurious fixed point at each cusp, causing the modeled dynamics to get stuck. Fortunately, the time derivative can be captured using the three sensors marked by black squares in Figure 10.3.1 and whose measurements are plotted in Figure 10.3.4c. One caveat is that these locations are far apart in space, and so the measurements will be more sensitive to perturbations of the shear-layer thickness which affects the horizontal spacing of vortices.

The linear techniques we have considered fail to reveal the minimum number of sensors needed to reconstruct the state because there is important information about the flow contained in less-energetic principal components. In particular, Figure 10.3.2c shows that the most energetic two principal components oscillate with twice the frequency of the third and fourth most energetic components as one moves around the orbit. In trying to maximize the variance captured by a linear estimator, the linear sensor placement techniques are doomed to choose sensors whose measurements return to the same values twice in one period as in Figures 10.3.3a, 10.3.3c, and 10.3.3e. In addition, the convex Bayesian D-optimal approach finds sensors that achieve a superior value of the objective



(a) secant detect. diffs., $\#(\mathcal{S}) = 2$ (b) secant detect. diffs., $\#(\mathcal{S}) = 3$ (c) secant amplification tol.

Figure 10.3.4: these plots show the measurements made by sensors selected using secant-based greedy optimization methods on the shock-mixing layer flow problem. Each dot indicates the values measured by the sensors and its color indicates the phase of the corresponding flowfield. In each case, the selected sensors make distinct measurements for distinct states, enabling reconstruction of the state from the measurements.

$\log\det \boldsymbol{C_e}(\mathcal{S})$ than the greedy Bayesian D-optimal approach, yet the resulting measurements in Figure 10.3.3d have many more self-intersections than the greedy method in Figure 10.3.3c.

We are forced to conclude that sensor placement based on linear reconstruction is totally unconnected with nonlinear reconstructability when the underlying manifold and principal dimensions do not agree. This can be seen most clearly from the fact that by simply re-scaling each coordinate in the toy model Eq. **10.8** by positive constants $\alpha_1, \ldots, \alpha_n$, we can trick linear techniques into selecting any given collection of coordinates. Under this scaling, the covariance matrix becomes $\mathrm{diag}(\alpha_1^2, \ldots, \alpha_n^2)$ and if we sort the constants in decreasing order $\alpha_{k_1} \geq \alpha_{k_2} \geq \cdots$ then the variance captured by a linear reconstruction from d measurements cannot exceed

$$R^2 \leq \frac{\alpha_{k_1}^2 + \cdots + \alpha_{k_d}^2}{\alpha_1^2 + \cdots + \alpha_n^2}, \tag{10.9}$$

according to the bound in Eq. **10.7**. Equality is achieved by the optimal linear estimator based on measured coordinates $x_{k_1}, \ldots, x_{k_d}$. Meanwhile, the only pair of coordinates needed for nonlinear reconstruction are x_1 and x_2.

The key point is that sensor placement approaches based on linear reconstruction tend to pick sensor locations that have high variance over other choices that can be more informative. The linear approach works well when a small number of principal components contain essentially all of the variance or when all higher modal components are very nearly determined by the lower ones. But as we have shown, linear approaches to sensor placement can fail catastrophically when genuinely informative fluctuations, e.g. sub-harmonics, produce significant variance orthogonal to the leading principal subspace. In order to reveal minimal sensor locations that can be used for

nonlinear reconstruction in such situations, we cannot rely on linear reconstruction performance as an optimization criteria, and an entirely new approach is needed. In Section **10.4** we discuss an approach that can recover the correct coordinates from which all others can be nonlinearly reconstructed.

10.3.3 Selecting Manifold Learning Coordinates

The examples presented in the previous Section 10.3.2 involved data lying near a one-dimensional underlying manifold. Essentially the same problems can occur for data lying near higher-dimensional manifolds, and an especially illustrative and practically useful application where this situation is routinely encountered is manifold learning. In general, manifold learning seeks to find a small collection of nonlinear coordinates that fully describe the structure of a dataset, i.e., that embed it in a lower-dimensional space. Many techniques including kernel PCA [244], Laplacian eigenmaps [15], diffusion maps [68], and Isomap [260] accomplish this via eigen-decomposition of various symmetric matrices

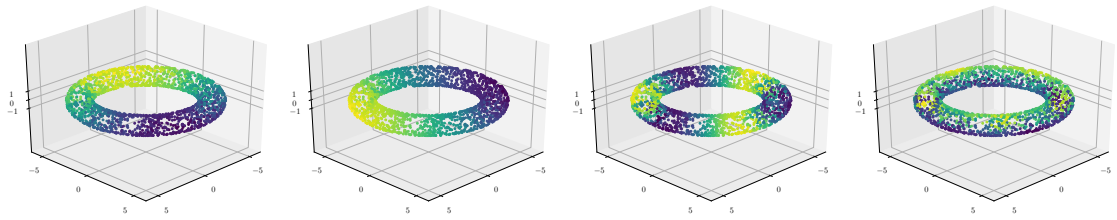
$$\mathbf{G} = \mathbf{\Phi}\mathbf{\Lambda}^2\mathbf{\Phi}^T, \qquad \mathbf{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_r \end{bmatrix} \tag{10.10}$$

derived from pair-wise similarity among data points. The kth eigen-coordinate of each point in the data set is given by the elements of $\boldsymbol{\phi}_k$, which can be viewed as a discrete approximation of an eigenfunction of some kernel integral operator on the underlying manifold. These methods suffer from a well-known issue when the dataset has multiple length scales: namely, there may be several redundant harmonically related eigen-coordinates with higher salience (determined by the eigenvalues) before one encounters a new fundamental eigen-coordinate describing a new set of features. This makes the search for a fundamental set of eigen-coordinates that embed the underlying manifold a potentially large combinatorial search problem.

As a concrete example, consider the Isomap eigen-coordinates shown in Figure 10.3.5 computed from 2000 points lying on the torus in $\mathbb{R}^3$,

$$\mathbf{x} = ((5 + \cos\theta_2)\cos\theta_1,\ (5 + \cos\theta_2)\sin\theta_1,\ \sin\theta_2)\,, \tag{10.11}$$

with (θ_1, θ_2) drawn uniformly at random from the square $[0, 2\pi] \times [0, 2\pi]$. Toroidal dynamics are known to occur in combustion instabilities where multiple incommensurate frequencies are observed [85], [144], producing data that winds around a torus in high-dimensional state space. One may want to build simplified reduced-order models of these dynamics by finding a small set of nonlinear



(a) Isomap ϕ_1 (b) Isomap ϕ_2 (c) Isomap ϕ_3 (d) Isomap ϕ_7

Figure 10.3.5: Isomap coordiantes computed from 2000 randomly sampled points on the torus defined by Eq. **10.11**. The leading six coordinates resemble the real and imaginary components of $e^{ik\theta_1}$, $k = 1, 2, 3$, due to the rotational symmetry, providing redundant information about θ_1 and no information about θ_2. The fundamental coordinates ϕ_1, ϕ_2, and ϕ_7 provide an embedding of the data that captures its toroidal structure.

coordinates that described the state on the torus using manifold learning.

Considering the torus in Eq. **10.11**, the underlying kernel integral operators associated with each manifold learning technique mentioned above are equivariant with respect to rotations about θ_1, meaning that among their eigenfunctions are always those of the symmetry's generator, namely $\phi_k(\boldsymbol{x}) = e^{ik\theta_1(\boldsymbol{x})}$. Unsurprisingly, the leading six Isomap eigen-coordinates, ranked by their associated eigenvalues, are all harmonically related modes resembling the real and imaginary parts of $e^{ik\theta_1}$, which provide redundant information about θ_1 and no information about θ_2. The coordinate θ_1 corresponds to larger spatial variations among points and it is not until we encounter the seventh eigen-coordinate that we learn about the smaller variations associated with θ_2. A naïve user of Isomap might plot the data in the leading three coordinates and falsely conclude that the data lies on a two-dimensional gasket. We'd like to provide an efficient method for selecting the fundamental eigen-coordinates ϕ_1, ϕ_2 and ϕ_7, from which all others can be (nonlinearly) reconstructed; yet again, linear methods fundamentally cannot be used to select them.

Linear methods cannot be used to select manifold learning eigen-coordinates for essentially the same reason why they failed on the toy models in Section 10.3.2: the coordinates are all mutually orthogonal as functions supported on the data! In particular, the covariance among the eigen-coordinates over the data is isotropic, $\mathbb{E}[\phi_i(\mathbf{x})\phi_j(\mathbf{x})] = \frac{1}{m}\boldsymbol{\phi}_i^T\boldsymbol{\phi}_j = \frac{1}{m}\delta_{i,j}$, and so all sub-collections of a given size capture the same fraction of the total eigen-coordinate variance. The methods presented in the following Section 10.4 remedy this issue and are capable of selecting the correct set of fundamental eigen-coordinates on the torus example in Eq. **10.11**.

Method	§	Property
Detectable diffs.	10.4.1	With a fixed sensor budget, this method greedily maximizes the sum of squared differences between relevant quantities over pairs of states whose measurements are separated by a used-defined detection threshold. As more measurements are chosen, more pairs become detectable and the total detectable difference increases.
Error tol.	10.4.2	This method greedily selects nearly the minimum possible number of sensors so that states whose measurements are closer together than a user-specified detection threshold never have associated relevant quantities differing by more than a user-specified error tolerance.
Amplif. tol.	10.4.3	This method greedily selects nearly the minimum possible number of sensors so that the largest ratio of differences between relevant quantities to differences between measurements is below a user-specified level of amplification, i.e., a reconstruction Lipschitz constant.

Table 10.4.1: We provide a summary of the three greedy measurement selection techniques we consider in this paper and their key properties.

10.4 Greedy Algorithms using Secants

With the failure of techniques based on linear reconstruction to select minimal collections of sensors for nonlinear reconstruction, we propose an alternative approach that relies on a collection of "secant" vectors between distinct data points. In this section, we develop this approach, yielding three related greedy selection techniques with classical theoretical guarantees on their performance. These techniques and their key properties are summarized in Table 10.4.1. We also discuss some theoretical results that provide deterministic performance guarantees for the sensors selected by our algorithms on unseen data drawn from an underlying set.

We consider a very general type of sensor placement problem that can be stated as follows. Let the set $\mathcal{X} \subset \mathbb{R}^n$ represent the possible states of the system and suppose that we are interested in some relevant information about the state described by a function $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^q$. The sensors are also described as functions of the state $\boldsymbol{m}_j : \mathcal{X} \to \mathbb{R}^{d_j}$, $j = 1, \ldots, M$ where, with a slight abuse of notation, we will denote the set of all sensors and the set of all sensor indices by $\mathcal{M}$ interchangeably. Our goal is to choose a small subset of sensors $\mathcal{S} = \{j_1, \ldots, j_K\} \subseteq \mathcal{M}$ so that the relevant information $\boldsymbol{g}(\boldsymbol{x})$ about any state $\boldsymbol{x} \in \mathcal{X}$ can be recovered from the combined measurements we have selected

$$\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) = (\boldsymbol{m}_{j_1}(\boldsymbol{x}), \ldots, \boldsymbol{m}_{j_K}(\boldsymbol{x})) \in \mathbb{R}^{d_{\mathcal{S}}}, \tag{10.12}$$

where the measurement dimension is $d_{\mathcal{S}} = \sum_{j \in \mathcal{S}} d_j$. That is, we want to choose $\mathcal{S}$ in such a way

that there exists a reconstruction function $\boldsymbol{\Phi}_{\mathcal{S}} : \mathbb{R}^{d_{\mathcal{S}}} \to \mathbb{R}^q$ so that

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{\Phi}_{\mathcal{S}}\left(\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})\right) \tag{10.13}$$

for every $\boldsymbol{x} \in \mathcal{X}$. This condition is automatically satisfied when the measurement functions $\boldsymbol{m}_{\mathcal{S}}$ are injective. On the other hand, it may be possible to reconstruct a function $\boldsymbol{g}$ containing incomplete information about the state using fewer measurements than are needed to make $\boldsymbol{m}_{\mathcal{S}}$ injective. If $\boldsymbol{g}$ is injective, then a reconstruction function $\boldsymbol{\Phi}_{\mathcal{S}}$ exists if and only if $\boldsymbol{m}_{\mathcal{S}}$ is injective

Remark 10.4.1 (Observability in dynamical systems)**.** *This framework is flexible enough to describe observability properties for dynamical systems through the use of time-delayed measurements. For instance, consider the observability problem for a discrete-time linear system $\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t$, in which we seek to choose from among a collection of observation matrices $\{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_K\}$, a set $\boldsymbol{C}_{\mathcal{S}}^T = \begin{bmatrix} \boldsymbol{C}_{j_1}^T & \cdots & \boldsymbol{C}_{j_K}^T \end{bmatrix}$ that allows the initial state $\boldsymbol{x}_0$ to be recovered from a length τ time-history of observations $\boldsymbol{C}_{\mathcal{S}}\boldsymbol{x}_0, \boldsymbol{C}_{\mathcal{S}}\boldsymbol{x}_1, \ldots, \boldsymbol{C}_{\mathcal{S}}\boldsymbol{x}_{\tau-1}$. In this case, we would let*

$$\boldsymbol{m}_j(\boldsymbol{x}) = \left(\boldsymbol{C}_j\boldsymbol{x}, \boldsymbol{C}_j\boldsymbol{A}\boldsymbol{x}, \ldots, \boldsymbol{C}_j\boldsymbol{A}^{\tau-1}\boldsymbol{x}\right) = \boldsymbol{O}_j\boldsymbol{x} \tag{10.14}$$

and reconstruct the desired state $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$ according to

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{\Phi}_{\mathcal{S}}\left(\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})\right) = \boldsymbol{W}_{\mathcal{S}}^{-1}\boldsymbol{O}_{\mathcal{S}}^T\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}), \tag{10.15}$$

where $\boldsymbol{W}_{\mathcal{S}} = \boldsymbol{O}_{\mathcal{S}}^T\boldsymbol{O}_{\mathcal{S}} = \sum_{t=0}^{\tau-1}(\boldsymbol{A}^t)^T\boldsymbol{C}_{\mathcal{S}}^T\boldsymbol{C}_{\mathcal{S}}\boldsymbol{A}^t = \sum_{j\in\mathcal{S}}\boldsymbol{W}_j$ is the usual time-τ observability Gramian for $(\boldsymbol{A}, \boldsymbol{C}_{\mathcal{S}})$. Sensor placement techniques for linear systems based on the observability Gramian have been developed by T. H. Summers, F. L. Cortesi and J. Lygeros in [252] and [253].

In the general case, for a reconstruction function $\boldsymbol{\Phi}_{\mathcal{S}}$ to exist, we must meet the modest condition that any two states $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ with different target values $\boldsymbol{g}(\boldsymbol{x}) \neq \boldsymbol{g}(\boldsymbol{x}')$ produce different measured values $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) \neq \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')$. This is nothing but the vertical line test for $\boldsymbol{\Phi}_{\mathcal{S}}$, ensuring that it is a true function that does not take multiple values. However, this condition may be met for a variety of different choices of measurements $\mathcal{S}$ and we shall introduce three different ways to quantify their performance and choose among them. In these methods, the notion that $\boldsymbol{\Phi}_{\mathcal{S}}$ should not be sensitive to perturbations of the measurements is key in quantifying the performance of the sensors. The techniques we propose each rely on secants, defined below, to measure the sensitivity of $\boldsymbol{\Phi}_{\mathcal{S}}$.

Definition 10.4.2 (Secant)**.** *A* secant *is a pair of states $(\boldsymbol{x}, \boldsymbol{x}')$, where $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and $\boldsymbol{x} \neq \boldsymbol{x}'$.*

By carefully choosing the objective functions $f : 2^{\mathcal{M}} \to \mathbb{R}$, we can rely on classical results by G. L. Nemhauser and L. A. Wolsey et al. [187], [283] to prove that greedy algorithms can be used to place the sensors with near-optimal performance. In particular, each objective that we propose is normalized so that $f(\emptyset) = 0$, monotone non-decreasing so that $\mathcal{S} \subseteq \mathcal{S}'$ implies $f(\mathcal{S}) \leq f(\mathcal{S}')$, and has a diminishing returns property called *submodularity*.

Definition 10.4.3 (Submodular Function)**.** *Let $\mathcal{M}$ be a finite set and denote the set of all subsets of $\mathcal{M}$ by $2^{\mathcal{M}}$. A real-valued function of the subsets $f : 2^{\mathcal{M}} \to \mathbb{R}$ is called "submodular" when it has the following diminishing returns property: for any element $j \in \mathcal{M}$ and subsets $\mathcal{S}, \mathcal{S}' \subseteq \mathcal{M}$,*

$$\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{M} \setminus \{j\} \quad \Rightarrow \quad f(\mathcal{S} \cup \{j\}) - f(\mathcal{S}) \geq f(\mathcal{S}' \cup \{j\}) - f(\mathcal{S}'). \tag{10.16}$$

That is, adding any new element j to the smaller set $\mathcal{S}$ increases f at least as much as adding the same element to the larger set $\mathcal{S}' \supseteq \mathcal{S}$.

Note that in applications we often do not have direct access to the full set $\mathcal{X}$, which may be continuous. Rather, we have a discrete collection of data $\mathcal{X}_N = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathcal{X}$, which we assume is large enough to achieve suitable approximations of the underlying set.

10.4.1 Maximizing Detectable Differences

As we have seen in the first half of this paper, a set of sensors can be considered good if nearby measurements come only from states whose target variables are also close together. Otherwise a small perturbation to the measurements results in a large change in the quantities of interest. One way to quantify this intuition is to select measurements that minimize the sum of squared differences in the target variables associated with states whose measurements are closer together than a fixed detection threshold $\gamma > 0$, i.e.,

$$F_\gamma(\mathcal{S}) := \sum_{\substack{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_N \ : \\ \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 < \gamma}} \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \,. \tag{10.17}$$

The choice of γ reflects the amount of noise or disturbances the sensor measurements should be able to tolerate without producing large reconstruction errors. For instance, γ might be selected so that the noise coming from a desired number $\#(\mathcal{S}) = K$ of sensors rarely perturbs the measurements by more than γ. If the noise $\boldsymbol{n}$ has $d_{\mathcal{S}}$ independent, identically distributed Gaussian entries, each with

variance σ^2, then the probability that the disturbance $\|\boldsymbol{n}\|_2$ exceeds $\gamma = \sigma\sqrt{d_{\mathcal{S}}} + \delta$ is bounded by

$$\mathbb{P}\left\{\|\boldsymbol{n}\|_2 \geq \gamma\right\} \leq \exp\left(-\frac{d_{\mathcal{S}}\delta^2}{2\sigma^2}\right) \tag{10.18}$$

according to Example 2.28 in M. J. Wainwright [273]. Such bounds on the noise or other measurement disturbances might serve as guidelines for selecting γ, although trying a range of choices may be necessary in order to obtain the best results in practice. Let the sum of squared differences in the target variables along each secant be denoted by

$$F_\infty := \sum_{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}_N} \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \,. \tag{10.19}$$

Then it is clear that minimizing the sum of squared "undetectable" differences given by Eq. **10.17** is equivalent to maximizing an objective function

$$\tilde{f}_\gamma(\mathcal{S}) = F_\infty - F_\gamma(\mathcal{S}) = \sum_{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}_N} \tilde{w}_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2, \tag{10.20}$$

where $\tilde{w}_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})$ is one if $\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 \geq \gamma$ and is zero otherwise. This weight function indicates whether our measurements $\boldsymbol{m}_{\mathcal{S}}$ can distinguish the states $\boldsymbol{x}$ and $\boldsymbol{x}'$ using the detection threshold γ, and may be written

$$\tilde{w}_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) = \mathbb{1}\left\{\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 \geq \gamma\right\}, \tag{10.21}$$

where $\mathbb{1}\{A\} = 1$ if A is true and 0 if A is false. Therefore, we can view the objective in Eq. **10.20** as the sum of squared differences that are "detectable."

Maximizing the objective in Eq. **10.20** over a fixed number of sensors $\#(\mathcal{S}) \leq K$ is a combinatorial optimization problem and to our knowledge does not admit an efficient direct approximation algorithm. However, if we reformulate the objective using a relaxed weight function

$$w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) = \min\left\{\frac{1}{\gamma^2}\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2^2,\ 1\right\}, \tag{10.22}$$

then

$$\boxed{f_\gamma(\mathcal{S}) = \sum_{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}_N} w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \, \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \,,} \tag{10.23}$$

obtained by replacing $\tilde{w}$ with w in Eq. **10.20**, becomes a normalized, monotone, submodular function

on subsets $\mathcal{S} \subseteq \mathcal{M}$ (Lemma 10.B.3 in the Appendix) and a simple greedy approximation algorithm guarantees near-optimal performance on this problem! The greedy algorithm produces a sequence of sets $\mathcal{S}_1, \mathcal{S}_2, \ldots$, by starting with $\mathcal{S}_0 = \emptyset$ and adding the sensor j_k to $\mathcal{S}_{k-1}$ that maximizes the objective $f_\gamma(\mathcal{S}_{k-1} \cup \{j\})$ over all $j \in \mathcal{M} \setminus \mathcal{S}_{k-1}$. If $\mathcal{S}_K^*$ maximizes $f_\gamma(\mathcal{S})$ over all subsets of size $\#(\mathcal{S}) = K$ then the classical result of G. L. Nemhauser et al. [187] states that the objective values attained by the greedily chosen sets satisfy

$$f_\gamma(\mathcal{S}_k) \geq \left(1 - e^{-k/K}\right) f_\gamma(\mathcal{S}_K^*), \qquad k = 1, \ldots, \#(\mathcal{M}). \tag{10.24}$$

The objective function f_γ given by Eq. **10.23** can be viewed as a "submodular relaxation" of the original sum of squared differences $\tilde{f}_\gamma$ given by Eq. **10.20**. While $f_\gamma(\mathcal{S}) \geq \tilde{f}_\gamma(\mathcal{S})$ for every $\mathcal{S} \subseteq \mathcal{M}$, Theorem 10.4.4, below, shows that f_γ also provides a lower bound on $\tilde{f}_{\gamma'}$ at reduced values of the detection threshold $\gamma' < \gamma$. Hence, maximization of f_γ is justified as a proxy for maximizing $\tilde{f}_{\gamma'}$. Moreover, the relaxed objective bounds the total square differences among target variables that are *not detectable* due to corresponding measurement differences smaller than reduced threshold via Eq. **10.26** of Theorem 10.4.4.

Theorem 10.4.4 (Relaxation Bound on Undetectable Differences)**.** *Consider the rigid and relaxed objectives given by Eq.* ***10.20*** *and Eq.* ***10.23****. Then for every* $\mathcal{S} \subseteq \mathcal{M}$ *and constant* $0 < \alpha < 1$*, we have*

$$\tilde{f}_{\alpha\gamma}(\mathcal{S}) \geq \frac{1}{1-\alpha^2} \left[f_\gamma(\mathcal{S}) - \alpha^2 F_\infty \right]. \tag{10.25}$$

Furthermore, the total fluctuation between target variables associated with states whose measurements are closer together than the reduced detection threshold $\alpha\gamma$*, given by Eq.* ***10.17****, is bounded above by*

$$F_{\alpha\gamma}(\mathcal{S}) \leq \frac{1}{1-\alpha^2} \left[F_\infty - f_\gamma(\mathcal{S}) \right]. \tag{10.26}$$

Proof. We observe that

$$\|\boldsymbol{m}_\mathcal{S}(\boldsymbol{x}) - \boldsymbol{m}_\mathcal{S}(\boldsymbol{x}')\|_2 \geq \alpha\gamma \quad \Leftrightarrow \quad w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \geq \alpha^2 \tag{10.27}$$

and so we have

$$\tilde{w}_{\alpha\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) = \mathbb{1}\left\{\|\boldsymbol{m}_\mathcal{S}(\boldsymbol{x}) - \boldsymbol{m}_\mathcal{S}(\boldsymbol{x}')\|_2 \geq \alpha\gamma\right\} \tag{10.28}$$

$$= \mathbb{1}\left\{w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \geq \alpha^2\right\}. \tag{10.29}$$

Since $0 \leq w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \leq 1$, we obtain the following linear lower bound

$$\tilde{w}_{\alpha\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \geq \frac{1}{1-\alpha^2}\left[w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) - \alpha^2\right]. \tag{10.30}$$

Summing this lower bound over all secants gives

$$\tilde{f}_{\alpha\gamma}(\mathcal{S}) \geq \frac{1}{1-\alpha^2}\left[f_\gamma(\mathcal{S}) - \alpha^2 F_\infty\right] \tag{10.31}$$

and subtracting each side from F_∞ yields the final result. □

When applied to the shock-mixing layer problem with the leading Isomap coordinates taken as the target variables $\boldsymbol{g}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}))$, the greedy algorithm maximizing f_γ first reveals the two sensor locations marked by green stars and then the black star in Figure 10.3.1 over the range of $0.02 \leq \gamma \leq 0.06$. These choices produce the measurements shown in Figs. 10.3.4a and 10.3.4b, which can be used to reveal the exact phase of the system. Choosing smaller values of γ yields different sensors that can also be used to reveal the phase, but with reduced robustness to measurement perturbations. This method of maximizing detectable differences also reveals the correct $K = 3$ fundamental Isomap eigen-coordinates from among the leading 100 on the torus example in Eq. **10.11** over a wide range $0.05 \leq \gamma \leq 3.0$. For implementation details, see the Appendix.

10.4.2 Minimal Sensing to Meet an Error Tolerance

The approach presented above relies on an average and so does not guarantee that the target value $\boldsymbol{g}(\boldsymbol{x})$ can be recovered from the selected measurements $\boldsymbol{m}_\mathcal{S}(\boldsymbol{x})$ for every $\boldsymbol{x} \in \mathcal{X}$. In this section, we modify the technique developed above in order to provide such a guarantee by trying to find the minimum number of sensors so that every pair of states in the sampled set $\mathcal{X}_N$ with target values separated by at least ε correspond to measurements separated by at least γ. If our sampled points $\mathcal{X}_N$ come sufficiently close to every point of $\mathcal{X}$ in the sense of Definition 10.4.5, then Proposition 10.4.6, given below, allows us to draw a similar conclusion about the measurements from all points in the underlying set $\mathcal{X}$.

Definition 10.4.5 (ε_0-net)**.** *An ε_0-net of $\mathcal{X}$ is a finite subset $\mathcal{X}_N \subset \mathcal{X}$ satisfying*

$$\forall \boldsymbol{x} \in \mathcal{X}, \quad \exists \boldsymbol{x}_i \in \mathcal{X}_N \quad \textit{such that} \quad \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 < \varepsilon_0. \tag{10.32}$$

We use the subscript N to denote the number of points in $\mathcal{X}_N$.

In particular, if $\mathcal{X}_N$ forms a fine enough ε_0-net of $\mathcal{X}$, then Proposition 10.4.6 guarantees that small measurement differences never correspond to large target value differences.

Proposition 10.4.6 (Separation Guarantee on Underlying Set)**.** *Let $\mathcal{X}_N$ be an ε_0-net of $\mathcal{X}$ (see Definition 10.4.5) and let $\mathcal{S}$ be a subset of $\mathcal{M}$ satisfying*

$$\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}_N \qquad \|\boldsymbol{g}(\boldsymbol{x}_i) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 \geq \varepsilon \quad \Rightarrow \quad \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_i) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2 \geq \gamma. \tag{10.33}$$

If $\boldsymbol{m}_{\mathcal{S}}$ and $\boldsymbol{g}$ are Lipschitz functions with Lipschitz constants $\|\boldsymbol{m}_{\mathcal{S}}\|_{lip}$ and $\|\boldsymbol{g}\|_{lip}$ respectively, then

$$\begin{aligned}\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X} \qquad & \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon + 2\varepsilon_0 \|\boldsymbol{g}\|_{lip} \\ & \Rightarrow \quad \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 > \gamma - 2\varepsilon_0 \|\boldsymbol{m}_{\mathcal{S}}\|_{lip}. \end{aligned} \tag{10.34}$$

Proof. The proof follows immediately from successive applications of the triangle inequality and so we relegate it to Appendix 10.C □

Consequently, the approach described in this section allows one to reconstruct $\boldsymbol{g}(\boldsymbol{x})$ from a perturbed measurement $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) + \boldsymbol{n}$ by taking the value $\boldsymbol{g}(\boldsymbol{x}')$ from its nearest neighbor $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')$ with $\boldsymbol{x}' \in \mathcal{X}$ and achieve small error $\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2$ as long as the perturbation $\|\boldsymbol{n}\|_2$ is below a threshold.

Supposing that the desired separation can be obtained using all of the sensors, i.e., $\mathcal{S} = \mathcal{M}$, then we can take the sum in the objective f_γ given by Eq. **10.23** only over those pairs $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_N$ with targets separated by at least $\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon$, i.e.,

$$\boxed{f_{\gamma,\varepsilon}(\mathcal{S}) = \sum_{\substack{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}_N \ : \\ \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon}} w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2,} \tag{10.35}$$

and state the problem formally as

$$\underset{\mathcal{S} \subseteq \mathcal{M}}{\text{minimize}} \ \ \#(\mathcal{S}) \quad \text{subject to} \quad f_{\gamma,\varepsilon}(\mathcal{S}) = f_{\gamma,\varepsilon}(\mathcal{M}). \tag{10.36}$$

We observe that if all points $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_N$ with $\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon$ can be separated by at least γ using $\mathcal{S} = \mathcal{M}$ then $w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{M}) = 1$ for each term in Eq. **10.35**. On the other hand if there is such a pair $\boldsymbol{x}, \boldsymbol{x}'$ with $\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 < \gamma$ then that term has $w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) < 1$ and $f_{\gamma,\varepsilon}(\mathcal{S}) < f_{\gamma,\varepsilon}(\mathcal{M})$ as a consequence.

One can show, by using the same argument as in Lemma 10.B.3 of the Appendix, that the objective Eq. **10.35** is submodular in addition to being normalized and monotone non-decreasing. It follows that Eq. **10.36** is a classical submodular set cover problem for which a greedy algorithm maximizing $f_{\gamma,\varepsilon}$ and stopping when $f_{\gamma,\varepsilon}(\mathcal{S}_K) = f_{\gamma,\varepsilon}(\mathcal{M})$ will always find, up to a logarithmic factor, the minimum possible number of sensors [283]. In particular, suppose that $\mathcal{S}^*$ is a subset of minimum size with $f_{\gamma,\varepsilon}(\mathcal{S}^*) = f(\mathcal{M})$ and that the greedy algorithm chooses a sequence of subsets $\mathcal{S}_1, \ldots, \mathcal{S}_K$ with $f_{\gamma,\varepsilon}(\mathcal{S}_K) = f_{\gamma,\varepsilon}(\mathcal{M})$. If we define the "increment condition number" to be the ratio of the largest and smallest increments in the objective during greedy optimization

$$\kappa = \frac{f_{\gamma,\varepsilon}(\mathcal{S}_1)}{f_{\gamma,\varepsilon}(\mathcal{S}_K) - f_{\gamma,\varepsilon}(\mathcal{S}_{K-1})}, \tag{10.37}$$

then the classical result of L. A. Wolsey [283] proves that the greedily chosen set is no larger than

$$\#(\mathcal{S}_K) \leq (1 + \ln \kappa)\#(\mathcal{S}^*). \tag{10.38}$$

10.4.3 Minimal Sensing to Meet an Amplification Tolerance

The approaches discussed above are capable of choosing measurements that separate states with distant target values by at least a fixed distance γ. However, we may want the separation between the measurements to grow with the corresponding separation in target values, rather than potentially saturating at the γ threshold. In addition, the nearby measurements separated by less than γ may not adequately capture the local behavior of the target variables as illustrated by the cusps in the measurements made by these sensors in the shock-mixing layer flow shown in Figure 10.3.4a. As we pointed out in Section 10.3.2, this would be a major problem if we wish to build a reduced-order model of this system in the measurement space because such a model would get stuck at spurious fixed points around each cusp. Overcoming this problem is important because it would allow us to build computationally efficient reduced-order models of fluid flows in measurement spaces consisting of easily interpretable fluid velocities at a small number of spatial locations. In particular, models of this kind could be build directly from the governing partial differential equations simply by reconstructing the flow variables in a spatial grid stencil around each selected location, allowing a finite-difference scheme to compute the time derivatives of the fluid velocities at the selected locations and evolve their dynamics forward in time.

Attempting to select sensors $\mathcal{S}$ whose measurements capture both the local and global structure of the target variables leads us to consider disturbance amplification as a performance metric. In

this section, we try to find the minimum number of sensors so that the Lipschitz constant of the reconstruction function does not exceed a user-specified threshold L.

Remark 10.4.7. *In the case of a linear system described in Remark 10.4.1, the reconstruction Lipschitz constant can be described in terms of the smallest eigenvalue of the time-τ observability Gramian $\lambda_{min}(\boldsymbol{W}_{\mathcal{S}})$ as*

$$\|\boldsymbol{\Phi}_{\mathcal{S}}\|_{lip} = \max_{\substack{\boldsymbol{m}\in\mathbb{R}^{d_{\mathcal{S}}}:\\ \boldsymbol{m}\neq\boldsymbol{0}}} \frac{\|\boldsymbol{W}_{\mathcal{S}}^{-1}\boldsymbol{O}_{\mathcal{S}}^{T}\boldsymbol{m}\|_2}{\|\boldsymbol{m}\|_2} = \frac{1}{\sqrt{\lambda_{min}(\boldsymbol{W}_{\mathcal{S}})}}. \tag{10.39}$$

In practice, we do not have access to the true Lipschitz constant, so instead we bound a proxy defined below:

$$\|\boldsymbol{\Phi}_{\mathcal{S}}\|_{\text{lip}} \approx \|\boldsymbol{\Phi}_{\mathcal{S}}\|_{\mathcal{X}_N,\text{lip}} = \max_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}_N} \frac{\|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{g}(\boldsymbol{x}')\|_2}{\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})-\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2} \leq L. \tag{10.40}$$

Proposition 10.4.8, below, shows that it suffices to enforce this condition over an ε_0-net, $\mathcal{X}_N$, of $\mathcal{X}$ (see Definition 10.4.5) in order to bound the amplification over all of $\mathcal{X}$ up to a slight relaxation for measurement differences on the same scale ε_0 as the sampling.

Proposition 10.4.8 (Amplification Guarantee on Underlying Set)**.** *Let $\mathcal{X}_N$ be an ε_0-net of $\mathcal{X}$ and let $\mathcal{S}$ be a subset of $\mathcal{M}$ satisfying*

$$\boldsymbol{x}_i,\boldsymbol{x}_j\in\mathcal{X}_N \qquad \|\boldsymbol{g}(\boldsymbol{x}_i)-\boldsymbol{g}(\boldsymbol{x}_j)\|_2 \leq L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_i)-\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2. \tag{10.41}$$

If $\boldsymbol{m}_{\mathcal{S}}$ and $\boldsymbol{g}$ are Lipschitz functions, with Lipschitz constants $\|\boldsymbol{m}_{\mathcal{S}}\|_{lip}$ and $\|\boldsymbol{g}\|_{lip}$ respectively, then

$$\forall \boldsymbol{x},\boldsymbol{x}'\in\mathcal{X} \qquad \|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{g}(\boldsymbol{x}')\|_2 < L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})+\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 + 2\left(\|\boldsymbol{g}\|_{lip}+L\|\boldsymbol{m}_{\mathcal{S}}\|_{lip}\right)\varepsilon_0. \tag{10.42}$$

Proof. The proof is a direct application of the triangle inequality and so it is relegated Appendix 10.C. □

If the Lipschitz condition in Eq. **10.40** over $\mathcal{X}_N$ can be met using all of the sensors $\mathcal{S}=\mathcal{M}$ then the problem we hope to solve can be stated formally as in Eq. **10.36**, where the condition Eq. **10.40** is imposed using a different normalized, monotone, submodular function

$$\boxed{f_L(\mathcal{S}) = \sum_{\substack{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}_N\\ \boldsymbol{g}(\boldsymbol{x})\neq\boldsymbol{g}(\boldsymbol{x}')}} \min\left\{\frac{\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})-\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2^2}{\|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{g}(\boldsymbol{x}')\|_2^2}, \frac{1}{L^2}\right\}.} \tag{10.43}$$

See Lemma 10.B.4 in the Appendix for proof of these properties.

Remark 10.4.9. *For a linear system described in Remark 10.4.1, the objective Eq.* ***10.43*** *can be written in terms of Rayleigh quotients involving the time-τ observability Gramian as*

$$f_L(\mathcal{S}) = \sum_{\substack{\boldsymbol{\xi} \in \mathcal{X}_N + (-\mathcal{X}_N): \\ \boldsymbol{\xi} \neq \mathbf{0}}} \min \left\{ \frac{\boldsymbol{\xi}^T \boldsymbol{W}_{\mathcal{S}} \boldsymbol{\xi}}{\boldsymbol{\xi}^T \boldsymbol{\xi}}, \ \frac{1}{L^2} \right\}. \tag{10.44}$$

We observe that if there is any secant $(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}_N \times \mathcal{X}_N$ for which Eq. **10.40** is not satisfied for a given $\mathcal{S} \subset \mathcal{M}$, then the corresponding term of Eq. **10.43** is less than $1/L^2$ and $f_L(\mathcal{S}) < f_L(\mathcal{M})$. Otherwise, each term of Eq. **10.43** is $1/L^2$ and we have $f_L(\mathcal{S}) = f_L(\mathcal{M})$. Again, the classical result in [283] shows that a greedy approximation algorithm maximizing Eq. **10.43** and stopping when $f_L(\mathcal{S}_K) = f_L(\mathcal{M})$ finds the minimum possible number of sensors up to a logarithmic factor so that the Lipschitz condition Eq. **10.40** is satisfied. In particular, the same guarantee stated in Eq. **10.38** holds for the Lipschitz objective too.

In some applications, we may instead want to find the measurements that minimize the reconstruction Lipschitz constant $\|\boldsymbol{\Phi}_{\mathcal{S}}\|_{\mathcal{X}_N,\text{lip}}$ using a fixed sensor budget $\#(\mathcal{S}) \leq C$. By running the greedy algorithm repeatedly using different thresholds L it is possible to obtain upper and sometimes lower bounds on this budget-constrained minimum Lipschitz constant L^*. This idea is closely related to the approach of [140]. If the greedy algorithm using Lipschitz constant L chooses sensors $\mathcal{S}$ that meet the budget $\#(\mathcal{S}) \leq C$ then L is obviously an upper bound on L^*. In practice, we can use a bisection search over L to find nearly the smallest L to any given tolerance for which $\#(\mathcal{S}) \leq C$. To get the lower bound, the greedy algorithm is run with a small enough L so that the bound on the minimum possible cost from Eq. **10.38** exceeds the budget

$$C < \#(\mathcal{S})/(1 + \ln \kappa). \tag{10.45}$$

If this is the case, there is no collection of measurements with amplification at most L that meets the cost constraint. Thus, such an L is a lower bound on the minimum possible amplification using measurement budget C. Again, bisection search can be used to find nearly the largest L so that $C < \#(\mathcal{S})/(1 + \ln \kappa)$.

With the leading Isomap coordinates taken as the target variables $\boldsymbol{g}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}))$, a bisection search over L identifies the three sensor locations marked by black squares in Figure 10.3.1 on the shock-mixing layer problem and the correct fundamental Isomap eigenfunctions ϕ_1, ϕ_2, ϕ_7 on the torus example in Eq. **10.11**. The measurements made by these sensors on the shock-mixing layer problem are shown in Figure 10.3.4c and indicate, by the lack of self-intersections, that they

can be used to recover the phase.

The minimum number of sensors selected by the greedy algorithm that allow one to reconstruct both the relevant information $\boldsymbol{g}(\boldsymbol{x})$ and its time derivative is usually persistent over a wide range of Lipschitz constants with fewer sensors not being chosen until L is made extremely large. In the shock-mixing layer problem, three sensors that successfully reveal the underlying phase are found for values of L ranging from 1868 to 47624, above which only two sensors that cannot reveal the underlying phase are selected. The fact that a smaller set of inadequate sensors are selected for extremely large L reflects our use of a discrete approximation $\mathcal{X}_N$ of the continuous set $\mathcal{X}$. Measurements from $\mathcal{X}_N$ will almost never truly overlap to give $\|\boldsymbol{\Phi}_{\mathcal{S}}\|_{\text{lip}} = \infty$ as they would for measurements from $\mathcal{X}$.

We also find that with $L = 129$, the minimum possible number of sensors exceeds $\#(\mathcal{S}_K)/(1 + \ln \kappa) = 3.18 > 3$ on the shock-mixing layer problem. Therefore, the minimum possible reconstruction Lipschitz constant using three sensors that one might find by an exhaustive search over the $\binom{2210}{3} \approx 1.8 \times 10^9$ possible combinations must be greater than 129. For implementation details, see the Appendix.

10.5 Computational Considerations and Down-Sampling

So far, the three secant-based methods we presented involve objectives that sum over $\mathcal{O}(N^2)$ pairs of points from the sampled set $\mathcal{X}_N$. In this section, we discuss how this large collection of secants can be sub-sampled to produce high-probability performance guarantees using a number of secants that scales more favorably with the size of the data set. By sub-sampling we do pay a price in the sense that some "bad" secants may escape our sampling scheme and so we cannot draw the same conclusions about every point in the underlying set as we did in Propositions 10.4.6 and 10.4.8 for the sensors chosen using the methods in Sections 10.4.2 and 10.4.3. Instead, we can bound the size of the set of these "bad" secants with high probability by using a sampled collection of secants that scales linearly with N. In the case of the total detectable difference-based objective discussed in Section 10.4.1, we can prove high-probability bounds for the sum of squared undetectable differences in the target variables using a constant number of secants that doesn't depend on N at all. The down-sampling properties of the various methods we propose are summarized in Table 10.5.1.

Remark 10.5.1 (The curse of dimensionality)**.** *It is important to note that even in the down-sampled setting, the error tolerance and the amplification tolerance methods suffer from the curse of dimensionality when we wish to draw conclusions about an underlying manifold $\mathcal{X}$. In such a case, we must take $\mathcal{X}_N$ to be an ε_0-net of $\mathcal{X}$ (see Definition. 10.4.5), where the number of elements*

Method	**# of secants**	**Property**
Detectable diffs.	$\mathcal{O}\left(\frac{K \ln \#\mathcal{M} - \ln p}{\varepsilon^2}\right)$	The sampled objective differs from the full objective by less than ε for every collection of K or fewer sensors with probability at least $1 - p$ by Lem. 10.5.2. Consequently, sampling does not reduce the worst-case performance of the greedy algorithm by more than 2ε with respect to the full objective by Thm. 10.5.3.
Error tol.	$\mathcal{O}\left(\frac{\#\mathcal{M} - \ln p}{\varepsilon^2}\right)$	When the error tol. method is used to ensure that every down-sampled pair with target variables differing by at least $\sqrt{\varepsilon}$ produces measurements separated by at least γ, then the full normalized sum of squared undetectable differences is less than 2ε with probability at least $1 - p$ by Thm. 10.5.5.
Error tol.	$\mathcal{O}\left(\frac{N(\#\mathcal{M} - \ln p)}{\delta^2}\right)$	The probability measure of the "bad set" of states that have secants not satisfying the desired error tolerance condition is less than δ with probability $1 - p$ (see Thm. 10.5.6).
Amplif. tol.	$\mathcal{O}\left(\frac{N(\#\mathcal{M} - \ln p)}{\delta^2}\right)$	The probability measure of the "bad set" of states that have secants with higher than desired amplification is less than δ with probability $1 - p$ (see Thm. 10.5.7).

Table 10.5.1: We summarize the down-sampling properties of the three greedy measurement selection techniques discussed in Section 10.4 using given number of secants to evaluate the objectives. For the sake of simplicity, the properties described in the last two rows pertain to any discrete set $\mathcal{X}_N$ of size N, whereas Theorems 10.5.6 and 10.5.7 provide results with respect to the underlying set $\mathcal{X}$ of which $\mathcal{X}_N$ is assumed to be an ε_0-net (see Def. 10.4.5). Since $\mathcal{X}_N$ is an ε_0-net of itself for every $\varepsilon_0 > 0$, the results above are easy corollaries of Theorems 10.5.6 and 10.5.7. The properties in the first two rows apply to the true underlying set $\mathcal{X}$.

in such a net scales with $N = \mathcal{O}(\varepsilon_0^{-d})$ and d is the dimension of $\mathcal{X}$. Consequently it will become computationally impractical to use enough points N to draw conclusions about the underlying set using these methods when the dimension of $\mathcal{X}$ becomes large (roughly $d > 5$). On the other hand, the detectable differences method is totally independent of the dimension of $\mathcal{X}$, but has weaker theoretical properties. Therefore, what we exchange for the generality of our approach in handling nonlinear sets is that these sets must be low-dimensional in order to guarantee reconstruction performance.

Before getting started with our discussion of down-sampling, let us first mention that the calculation of each of the objectives formulated in Section 10.4 is easily parallelizable, whether or not they are down-sampled. Even though the computation of each objective function given by Eq. **10.23**, **10.35**, or **10.43** requires $\mathcal{O}(N^2)$ operations, the terms being summed can be distributed among many processors without the need for any communication except at the end when each processor reports the sum over the secants allocated to it. Furthermore, because each secant-based objective we consider in this paper is submodular, it is not actually necessary to evaluate the objectives over all of the remaining sensors during each step of the greedy algorithm. By employing the "accelerated greedy" algorithm of M. Minoux [180], the same set of sensors can be found using a minimal number of evaluations of the objective. We provide a summary of the accelerated greedy algorithm in Section 10.D of the Appendix.

The wall-clock computation times for our secant-based methods using the accelerated greedy algorithm implemented in Python without the aforementioned parallelization and running on a laptop computer were in the 10s of seconds per set of sensors, with bisection searches to minimize amplification on a fixed sensor budget (see Section 10.4.3) taking minutes. These times were comparable to the LASSO method, with the secant-based approach being slower by a factor of about 2, and slower than the convex approaches for D-optimal selection by a factor of about 3. On the other hand, the pivoted QR method and greedy D-optimal selection methods were extremely fast, producing sensors in fractions of a second. For implementation details, see Appendix 10.A

The computational cost of evaluating the objectives in Sections 10.4.2 and 10.4.3 during each step of the greedy algorithm may also be reduced by exploiting the fact that each term in the sum is truncated once the measurements achieve a certain level of separation. This means that only the nearest neighbors within a known distance of each $\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x})$, $\boldsymbol{x} \in \mathcal{X}_N$ need to be computed and rest of the terms all achieve the threshold and need not be computed explicitly. To compute the sum efficiently, fixed-radius near neighbors algorithms [21], [22] could be employed.

10.5.1 Maximizing Detectable Differences

The main results of this section are Theorems 10.5.3 and 10.5.5, which show that with high probability we can obtain guaranteed performance in terms of mean undetectable differences by sampling a constant number of secants (i.e., independent of N) selected at random. In particular, Theorem 10.5.3 bounds the worst-case performance of the greedy algorithm with high probability using the sampled objective. Theorem 10.5.5, on the other hand, shows that if one only considers randomly sampled secants with target variables separated by at least ε (see Section 10.4.2), then the mean square undetectable difference between target values is less than $2\varepsilon^2$ with high probability.

While the original mean square fluctuation objective in Eq. **10.23** was formulated over the discrete set $\mathcal{X}_N$, we can actually prove more versatile approximation results about an objective defined as an average over the entire, possibly continuous, set $\mathcal{X}$ with respect to a probability measure μ. In particular, we assume the target variables $\boldsymbol{g}$ and measurements $\boldsymbol{m}_j$, $j \in \mathcal{M}$ are measurable functions on $\mathcal{X}$ and consider an average detectable difference objective

$$f_\gamma(\mathcal{S}) = \int_{\mathcal{X}\times\mathcal{X}} w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \, \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \; d\mu(\boldsymbol{x}) d\mu(\boldsymbol{x}') \tag{10.46}$$

with $w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})$ defined by Eq. **10.22**. We also denote the average fluctuations between target variables associated with states whose measurements are closer together than the detection threshold γ by

$$F_\gamma(\mathcal{S}) := \int_{\substack{(\boldsymbol{x},\boldsymbol{x}')\in\mathcal{X}\times\mathcal{X} \,:\, \\ \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 < \gamma}} \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \; d\mu(\boldsymbol{x}) d\mu(\boldsymbol{x}') \tag{10.47}$$

and the total fluctuation among target variables by

$$F_\infty := \int_{\mathcal{X}\times\mathcal{X}} \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \; d\mu(\boldsymbol{x}) d\mu(\boldsymbol{x}'). \tag{10.48}$$

Note that the original objective formulated in Section 10.4.1 as well as Eq. **10.17** are special cases of Eq. **10.46** and Eq. **10.47**, up to an irrelevant constant factor, when $\mu = \frac{1}{N}\sum_{\boldsymbol{x}\in\mathcal{X}_N} \delta_{\boldsymbol{x}}$ and $\delta_{\boldsymbol{x}}(A) = \mathbb{1}\{\boldsymbol{x} \in A\}$ is the Dirac measure on Borel sets $A \subseteq \mathcal{X}$. By Lemma 10.B.3, Eq. **10.46** is submodular in addition to being normalized and monotone non-decreasing. Furthermore, by an identical argument to Theorem 10.4.4, we know that the mean square fluctuation between target variables associated with states whose measurements are closer together than a reduced detection

thereshold $\alpha\gamma$ with $0 < \alpha < 1$ is bounded above by

$$F_{\alpha\gamma}(\mathcal{S}) \leq \frac{1}{1-\alpha^2}\left[F_\infty - f_\gamma(\mathcal{S})\right]. \tag{10.49}$$

We begin with Lemma 10.5.2, which shows that by sampling a large enough collection of points $\boldsymbol{x}_1, \boldsymbol{x}_1', \ldots, \boldsymbol{x}_m, \boldsymbol{x}_m' \in \mathcal{X}$ independently according to μ, the objective f_γ can be uniformly approximated by a sample-based average

$$f_{\gamma,m}(\mathcal{S}) = \frac{1}{m}\sum_{i=1}^{m} w_{\gamma,\boldsymbol{x}_i,\boldsymbol{x}_i'}(\mathcal{S})\,\|\boldsymbol{g}(\boldsymbol{x}_i) - \boldsymbol{g}(\boldsymbol{x}_i')\|_2^2 \tag{10.50}$$

over all $\mathcal{S} \subseteq \mathcal{M}$ of size $\#(\mathcal{S}) \leq L$ with high probability over the sample points. Most importantly, the number of sample points needed for this approximation guarantee is independent of the distribution μ. Consequently if we have access to N points making up $\mathcal{X}_N$ that have been sampled independently according to μ, we need only keep the first $2m$ of them to accurately approximate the objective. The number m of such sub-sampled points depends only on the quality of the probabilistic guarantee and not on the size of the data set N.

Lemma 10.5.2 (Accuracy of the Down-Sampled Objective)**.** *Consider the objectives f_γ and $f_{\gamma,m}$ defined according to Eq.* ***10.46*** *and Eq.* ***10.50****. Assume that the target function is bounded over $\mathcal{X}$ so that*

$$D = \operatorname{diam} \boldsymbol{g}(\mathcal{X}) = \sup_{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}} \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 < \infty. \tag{10.51}$$

and that $\boldsymbol{x}_1, \boldsymbol{x}_1', \ldots, \boldsymbol{x}_m, \boldsymbol{x}_m' \in \mathcal{X}$ are sampled independently according to a probability measure μ on $\mathcal{X}$. If the number of sampled pairs is at least

$$m \geq \frac{D^4}{2\varepsilon^2}\left[L\ln\#(\mathcal{M}) - \ln\left((L-1)!\right) - \ln\left(\frac{p}{2}\right)\right], \tag{10.52}$$

then $|f_{\gamma,m}(\mathcal{S}) - f_\gamma(\mathcal{S})| < \varepsilon$ for every $\mathcal{S} \subseteq \mathcal{M}$ of size $\#(\mathcal{S}) \leq L$ with probability at least $1 - p$.

Proof. For simplicity, we will drop γ from the subscripts on our objectives since γ remains fixed throughout the proof. Let us begin by fixing a set $\mathcal{S} \subseteq \mathcal{M}$ of size $\#(\mathcal{S}) \leq L$ and denoting $M = \#(\mathcal{M})$ for short. Under the assumption that the points $\boldsymbol{x}_i, \boldsymbol{x}_i'$ are sampled independently and identically under μ, the random variables

$$Z_i(\mathcal{S}) = w_{\boldsymbol{x}_i,\boldsymbol{x}_i'}(\mathcal{S})\|\boldsymbol{g}(\boldsymbol{x}_i) - \boldsymbol{g}(\boldsymbol{x}_i')\|_2^2, \quad i = 1, \ldots, m, \tag{10.53}$$

are independent and bounded by $0 \leq Z_i(\mathcal{S}) \leq D^2$. The value of the optimization objective is the expectation $f(\mathcal{S}) = \mathbb{E}[Z_i(\mathcal{S})]$ and the value of our sub-sampled objective is the empirical average

$$f_m(\mathcal{S}) = \frac{1}{m} \sum_{i=1}^{m} Z_i(\mathcal{S}). \tag{10.54}$$

Hoeffding's inequality allows us to bound the probability that $f_m(\mathcal{S})$ differs from $f(\mathcal{S})$ by more than ε according to

$$\mathbb{P}\left\{|f_m(\mathcal{S}) - f(\mathcal{S})| \geq \varepsilon\right\} \leq 2\exp\left(-\frac{2m\varepsilon^2}{D^4}\right). \tag{10.55}$$

We want the objective to be accurately approximated with tolerance ε uniformly over all collections of sensors of size $\#(\mathcal{S}) \leq L$. We unfix $\mathcal{S}$ by taking the union bound

$$\mathbb{P} \bigcup_{\substack{\mathcal{S}\subseteq\mathcal{M}: \\ \#(\mathcal{S})\leq L}} \left\{|f_m(\mathcal{S}) - f(\mathcal{S})| \geq \varepsilon\right\} \leq \sum_{\substack{\mathcal{S}\subseteq\mathcal{M}: \\ \#(\mathcal{S})\leq L}} 2\exp\left(-\frac{2m\varepsilon^2}{D^4}\right). \tag{10.56}$$

The combinatorial inequality

$$\#\left(\{\mathcal{S} \subseteq \mathcal{M} \ : \ \#(\mathcal{S}) \leq L\}\right) = \sum_{k=1}^{L} \binom{M}{k} \leq \sum_{k=1}^{L} \frac{M^k}{k!} \leq L\frac{M^L}{L!} = \frac{M^L}{(L-1)!} \tag{10.57}$$

yields the bound

$$\mathbb{P} \bigcup_{\substack{\mathcal{S}\subseteq\mathcal{M}: \\ \#(\mathcal{S})\leq L}} \left\{|f_m(\mathcal{S}) - f(\mathcal{S})| \geq \varepsilon\right\} \leq 2\exp\left(L\ln M - \ln\left((L-1)!\right) - \frac{2m\varepsilon^2}{D^4}\right) \leq p \tag{10.58}$$

when the number of sampled pairs $\boldsymbol{x}_i, \boldsymbol{x}_i'$ satisfies Eq. **10.52**. □

The uniform accuracy of the sampled objective $f_{\gamma,m}$ over the feasible subsets $\mathcal{S}$ in our optimization problem

$$\underset{\mathcal{S}\subseteq\mathcal{M} \ : \ \#(\mathcal{S})\leq K}{\text{maximize}} f_\gamma(\mathcal{S}) \tag{10.59}$$

established in Lemma 10.5.2 leads to performance guarantees for the greedy approximation algorithm when the sampled objective $f_{\gamma.m}$ is used in place of f_γ. In particular, Theorem 10.5.3 shows that the greedy algorithm can be applied to the sampled objective Eq. **10.50** and still achieve near-optimal performance with respect to the original objective Eq. **10.46** on the underlying set $\mathcal{X}$ with high probability. This sampling-based approach therefore completely eliminates the $\mathcal{O}(N^2)$ dependence of the computational complexity involved in evaluating the objective at a penalty on the worst case

performance that can be made arbitrarily small by sampling more points.

Theorem 10.5.3 (Greedy Performance using Sampled Objective)**.** *Assume the same hypotheses as Lemma 10.5.2 and let $\mathcal{S}^*$ denote an optimal solution of*

$$\underset{\mathcal{S}\subseteq\mathcal{M}\,:\,\#(\mathcal{S})\leq K}{\text{maximize}} f_\gamma(\mathcal{S}), \tag{10.60}$$

with f_γ given by Eq. ***10.46*** *and $K \leq L$. If $\mathcal{S}_1, \ldots, \mathcal{S}_L$ are the sequence of subsets selected by the greedy algorithm using the sampled objective $f_{\gamma,m}$ given by Eq.* ***10.50****, then*

$$f_\gamma(\mathcal{S}_k) \geq \left(1 - e^{-k/K}\right) f_\gamma(\mathcal{S}^*) - \left(2 - e^{-k/K}\right)\varepsilon, \qquad k = 1, \ldots, L, \tag{10.61}$$

with probability at least $1 - p$ over the sample points.

Proof. For simplicity, we will drop γ from the subscripts on our objectives since γ remains fixed throughout the proof. Let $\mathcal{S}_m^*$ denote the optimal solution of

$$\underset{\mathcal{S}\subseteq\mathcal{M}\,:\,\#(\mathcal{S})\leq K}{\text{maximize}} f_m(\mathcal{S}), \tag{10.62}$$

using the sampled objective and assume that $|f(\mathcal{S}) - f_m(\mathcal{S})| < \varepsilon$ for every subset $\mathcal{S}$ of $\mathcal{M}$ with $\#(\mathcal{S}) \leq L$. According to Lemma 10.5.2, this happens with probability at least $1 - p$ over the sample points. Using this uniform approximation and the guarantee on the performance of the greedy algorithm for f_m, we have

$$f(\mathcal{S}_k) \geq f_m(\mathcal{S}_k) - \varepsilon \geq \left(1 - e^{-k/K}\right) f_m(\mathcal{S}_m^*) - \varepsilon. \tag{10.63}$$

Since $\mathcal{S}_m^*$ is the optimal solution using the sampled objective, we must have $f_m(\mathcal{S}_m^*) \geq f_m(\mathcal{S}^*)$. Using this fact and the uniform approximation gives

$$f(\mathcal{S}_k) \geq \left(1 - e^{-k/K}\right) f_m(\mathcal{S}^*) - \varepsilon \tag{10.64}$$

$$\geq \left(1 - e^{-k/K}\right) (f(\mathcal{S}^*) - \varepsilon) - \varepsilon. \tag{10.65}$$

Combining the terms on ε completes the proof. □

Remark 10.5.4. *While Theorem 10.5.3 tells us that down-sampling has a small effect on the worst-case performance of the greedy algorithm, unfortunately, we cannot say much beyond that. It may be*

the case that the greedy solution using the sampled objective $f_{\gamma,m}$ produces a very different value of f_γ than the greedy solution using f_γ directly, even though these functions are both submodular and differ by no more than an arbitrarily small $\varepsilon > 0$. Consider the following example in Table 10.5.2 where we have two submodular objectives, f and $\tilde{f}$, that differ by no more than $\varepsilon \ll 1$, yet the greedy algorithm applied to f and $\tilde{f}$ yield results that differ by $\mathcal{O}(1)$. One can easily verify that both

$\mathcal{S}$	$f(\mathcal{S})$	$\tilde{f}(\mathcal{S})$
$\emptyset$	0	0
$\{a\}$	$2+\varepsilon$	2
$\{b\}$	2	$2+\varepsilon$
$\{c\}$	1	1
$\{a,b\}$	$2+\varepsilon$	$2+2\varepsilon$
$\{a,c\}$	$3+\varepsilon$	3
$\{b,c\}$	2	$2+\varepsilon$
$\{a,b,c\}$	3	3

Table 10.5.2: Two submodular functions are given that differ by no more than $\varepsilon \ll 1$, yet produce very different greedy solutions and objective values.

functions in Table 10.5.2 are normalized, monotone, and submodular. When selecting subsets of size 2, the greedy algorithm for f picks $\emptyset \to \{a\} \to \{a,c\}$ and the greedy algorithm for for $\tilde{f}$ picks $\emptyset \to \{b\} \to \{a,b\}$. The values of f on the chosen sets, $f(\{a,c\}) = 3+\varepsilon$ and $f(\{a,b\}) = 2+2\varepsilon$, differ by $1-\varepsilon \gg \varepsilon$, and similarly for $\tilde{f}(\{a,c\}) = 3$ and $\tilde{f}(\{a,b\}) = 2+\varepsilon$, which also differ by $1-\varepsilon \gg \varepsilon$. Thus the performance of the greedy algorithm can be sensitive to small perturbations of the objective even though the lower bound on performance is not sensitive.

It turns out that by solving the error tolerance problem in Section 10.4.2 greedily using a down-sampled objective, we can provide high probability bounds directly on the mean square undetectable differences in Eq. **10.47**. We will use the down-sampled objective

$$f_{\gamma,\varepsilon,m}(\mathcal{S}) = \frac{1}{m} \sum_{\substack{i\in\{1,\ldots,m\}:\\ \|\boldsymbol{g}(\boldsymbol{x}_i)-\boldsymbol{g}(\boldsymbol{x}'_i)\|_2\geq\varepsilon}} w_{\gamma,\boldsymbol{x}_i,\boldsymbol{x}'_i}(\mathcal{S})\|\boldsymbol{g}(\boldsymbol{x}_i) - \boldsymbol{g}(\boldsymbol{x}'_i)\|_2^2, \tag{10.66}$$

with the relaxed weight function in Eq.**10.22** in a greedy approximation algorithm for the submodular set-cover problem

$$\underset{\mathcal{S}\subseteq\mathcal{M}}{\text{minimize}} \;\; \#(\mathcal{S}) \quad \text{subject to} \quad f_{\gamma,\varepsilon,m}(\mathcal{S}) = f_{\gamma,\varepsilon,m}(\mathcal{M}). \tag{10.67}$$

Using the resulting greedy solution $\mathcal{S}_K$ that satisfies $f_{\gamma,\varepsilon,m}(\mathcal{S}_K) = f_{\gamma,\varepsilon,m}(\mathcal{M}) = \tilde{f}_{\gamma,\varepsilon,m}(\mathcal{M})$, Theorem 10.5.5 provides a high-probability bound on the mean square undetectable difference in the

target variables, Eq. **10.47**, over the entire set $\mathcal{X} \times \mathcal{X}$ rather than merely $\mathcal{X}_N \times \mathcal{X}_N$.

Theorem 10.5.5 (Sample Separation Bound on Undetectable Differences)**.** *Consider the functions $f_{\gamma,\varepsilon,m}$ and F_γ defined by Eqs.* ***10.67*** *and* ***10.47*** *and assume that the condition $\|\boldsymbol{m}_{\mathcal{M}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{M}}(\boldsymbol{x}')\|_2 \geq \gamma$ holds for μ-almost every $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ such that $\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon$. Suppose that the target function is bounded over $\mathcal{X}$ so that*

$$D = \operatorname{diam} \boldsymbol{g}(\mathcal{X}) = \sup_{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}} \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 < \infty. \tag{10.68}$$

and that $\boldsymbol{x}_1, \boldsymbol{x}'_1, \ldots, \boldsymbol{x}_m, \boldsymbol{x}'_m \in \mathcal{X}$ are sampled independently according to the probability measure μ on $\mathcal{X}$. If the number of sampled pairs is at least

$$m \geq \frac{D^4}{2\varepsilon^4} \left(\#(\mathcal{M}) \ln 2 - \ln p\right), \tag{10.69}$$

and the greedy approximation of Eq. ***10.67*** *produces a set $\mathcal{S}_K$, then*

$$F_\gamma(\mathcal{S}_K) < 2\varepsilon^2 \tag{10.70}$$

with probability at least $1 - p$.

Proof. For simplicity, we will drop γ, ε from the subscripts on our objectives since γ and ε remain fixed throughout the proof. Let

$$\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X} \times \mathcal{X} \ : \ \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon\} \tag{10.71}$$

and

$$\tilde{f}(\mathcal{S}) = \mathbb{E}\left[\tilde{f}_m(\mathcal{S})\right] = \int_{\mathcal{X}\times\mathcal{X}} \chi_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{x}') \tilde{w}_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \, d\mu(\boldsymbol{x}) d\mu(\boldsymbol{x}'), \tag{10.72}$$

where $\chi_{\mathcal{D}}$ is the characteristic function of the set $\mathcal{D}$. From our assumption that $\|\boldsymbol{m}_{\mathcal{M}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{M}}(\boldsymbol{x})\|_2 \geq \gamma$ for μ-almost every $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ with $\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon$, it follows

$$\tilde{f}(\mathcal{M}) = \int_{\mathcal{X}\times\mathcal{X}} \chi_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{x}') \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \, d\mu(\boldsymbol{x}) d\mu(\boldsymbol{x}'). \tag{10.73}$$

Expanding our definition of F_γ in Eq.**10.47**, we find

$$F_\gamma(\mathcal{S}) = \tilde{f}(\mathcal{M}) - \tilde{f}(\mathcal{S}) + \int_{\mathcal{X}\times\mathcal{X}} \chi_{\mathcal{D}^c}(\boldsymbol{x},\boldsymbol{x}')\left[1 - \tilde{w}_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})\right] \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \, d\mu(\boldsymbol{x}) d\mu(\boldsymbol{x}') \tag{10.74}$$

and therefore

$$F_\gamma(\mathcal{S}) \leq \tilde{f}(\mathcal{M}) - \tilde{f}(\mathcal{S}) + \varepsilon^2. \tag{10.75}$$

We shall now use a similar Hoeffding and union bound argument as in Thm. 10.5.2 to relate $\tilde{f}(\mathcal{M}) - \tilde{f}(\mathcal{S})$ to $\tilde{f}_m(\mathcal{M}) - \tilde{f}_m(\mathcal{S})$ uniformly over every subset $\mathcal{S} \subseteq \mathcal{M}$. Fixing such $\mathcal{S} \subset \mathcal{M}$, the one-sided Hoeffding inequality tells us that

$$\mathbb{P}\left\{\left[\tilde{f}(\mathcal{M}) - \tilde{f}(\mathcal{S})\right] - \left[\tilde{f}_m(\mathcal{M}) - \tilde{f}_m(\mathcal{S})\right] \geq \varepsilon^2\right\} \leq \exp\left(-\frac{2m\varepsilon^4}{D^4}\right). \tag{10.76}$$

Unfixing $\mathcal{S}$ using the union bound tells us that

$$\tilde{f}(\mathcal{M}) - \tilde{f}(\mathcal{S}) < \tilde{f}_m(\mathcal{M}) - \tilde{f}_m(\mathcal{S}) + \varepsilon^2 \tag{10.77}$$

uniformly over all $\mathcal{S} \subset \mathcal{M}$ with probability at least $1 - p$. Since the greedy algorithm terminates when $\tilde{f}_m(\mathcal{S}_K) = \tilde{f}_m(\mathcal{M})$, it follows by substitution into Eq. **10.75** that

$$F_\gamma(\mathcal{S}) < 2\varepsilon^2 \tag{10.78}$$

with probability at least $1 - p$ over the sample points. □

10.5.2 Minimal Sensing to Meet Separation or Amplification Tolerances

If we want to draw stronger conclusions about the underlying set $\mathcal{X}$ than are captured by the mean square (un)detectable differences, then we must increase the number of sample points. The following Theorems 10.5.6 and 10.5.7 show that similar conclusions about the separation of points as in Propositions 10.4.6 and 10.4.8 can be achieved over large subsets of $\mathcal{X}$ with high probability by considering secants between a randomly chosen set of "base points" and the full data set. More precisely, we will consider secants between an ε_0-net $\mathcal{X}_N$ of $\mathcal{X}$ and a collection of base point $\mathcal{B}_m \subset \mathcal{X}$ with size m independent of N. This leads to linear $\mathcal{O}(N)$ scaling of the cost to evaluate the down-sampled versions of the objectives given by Eqs. **10.35** and **10.43** in Sections 10.4.2 and 10.4.3 to achieve these relaxed guarantees.

The strong guarantee of Proposition 10.4.6 requires that we use an objective like Eq. **10.35** in the submodular set-cover problem Eq. **10.36** where the sum in Eq. **10.35** is taken over $\mathcal{X}_N \times \mathcal{X}_N$ and $\mathcal{X}_N$ is an ε_0-net of the underlying set $\mathcal{X}$. The problem is that the ε_0-net $\mathcal{X}_N$ may be quite large and the number of operations needed to evaluate the sum in the objective scales with the square of the size of $\mathcal{X}_N$. Here we will prove that a similar guarantee as in Proposition 10.4.6 holds with high probability over a large subset of $\mathcal{X}$ when the sum in Eq. **10.35** is taken over secants between a randomly chosen collection of base points $\mathcal{B}_m = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\} \subseteq \mathcal{X}$ and the ε_0-net $\mathcal{X}_N$. Most importantly, the number of base points depends on the quality of the guarantee and not on size of the ε_0-net, so that the computational cost can be reduced to linear dependence on the size of $\mathcal{X}_N$.

Specifically, in place of Eq. **10.35**, we can consider the sampled objective

$$f_{\gamma,\varepsilon,m}(\mathcal{S}) = \frac{1}{mN} \sum_{\substack{1 \le i \le m,\ 1 \le j \le N: \\ \|\boldsymbol{g}(\boldsymbol{b}_i) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 \ge \varepsilon}} w_{\gamma,\boldsymbol{b}_i,\boldsymbol{x}_j}(\mathcal{S}) \|\boldsymbol{g}(\boldsymbol{b}_i) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2^2 \tag{10.79}$$

with $w_{\gamma,\boldsymbol{b}_i,\boldsymbol{x}_j}(\mathcal{S})$ defined by Eq. **10.22** in the optimization problem Eq. **10.36**. The greedy approximation algorithm produces a set of sensors $\mathcal{S}_K$ such that

$$\|\boldsymbol{g}(\boldsymbol{b}_i) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 \ge \varepsilon \quad \Rightarrow \quad \|\boldsymbol{m}_{\mathcal{S}_K}(\boldsymbol{b}_i) - \boldsymbol{m}_{\mathcal{S}_K}(\boldsymbol{x}_j)\|_2 \ge \gamma \tag{10.80}$$

for every $\boldsymbol{b}_i \in \mathcal{B}_m$ and $\boldsymbol{x}_j \in \mathcal{X}_N$. Theorem 10.5.6 guarantees that with high probability, only a small subset of points in $\mathcal{X}$ have target values that cannot be distinguished from the rest by measurements separated by a relaxed detection threshold. This size of this "bad set" is determined by its μ-measure, which can be made arbitrarily small with high probability by taking more sample base points m.

Theorem 10.5.6 (Sampled Separation Guarantee)**.** *Let $\mathcal{X}_N$ be an ε_0-net of $\mathcal{X}$ and let the base points $\mathcal{B}_m$ be sampled independently according to a probability measure μ on $\mathcal{X}$ with*

$$m \ge \frac{1}{2\delta^2} \left(\#(\mathcal{M}) \ln 2 - \ln p \right), \tag{10.81}$$

*where $p, \delta \in (0,1)$. Consider the objective $f_{\gamma,\varepsilon,m}$ given by Eq. **10.79** for a certain choice of $\gamma > 0$ and $\varepsilon > 0$ for which every $\boldsymbol{b}_i \in \mathcal{B}_m$ and $\boldsymbol{x}_j \in \mathcal{X}_N$ satisfies*

$$\|\boldsymbol{g}(\boldsymbol{b}_i) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 \ge \varepsilon \quad \Rightarrow \quad \|\boldsymbol{m}_{\mathcal{M}}(\boldsymbol{b}_i) - \boldsymbol{m}_{\mathcal{M}}(\boldsymbol{x}_j)\|_2 \ge \gamma. \tag{10.82}$$

Suppose also that $\boldsymbol{g}$ and the measurement functions $\boldsymbol{m}_k$, $k \in \mathcal{M}$ are all Lipschitz over $\mathcal{X}$. If

$f_{\gamma,\varepsilon,m}(\mathcal{S}) = f_{\gamma,\varepsilon,m}(\mathcal{M})$, *then the* μ *measure of points* $\boldsymbol{x} \in \mathcal{X}$ *such that*

$$\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon + \varepsilon_0 \|\boldsymbol{g}\|_{lip} \Rightarrow \quad \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 > \gamma - \varepsilon_0 \|\boldsymbol{m}_{\mathcal{S}}\|_{lip} \tag{10.83}$$

for every $\boldsymbol{x}' \in \mathcal{X}$ *is at least* $1 - \delta$ *with probability at least* $1 - p$.

Proof. For simplicity, we will drop γ, ε from the subscript on our objective since γ and ε remain fixed throughout the proof. Let us begin by fixing a set $\mathcal{S} \subseteq \mathcal{M}$ and define the random variables

$$Z_{\mathcal{S}}(\boldsymbol{b}_i) = \max_{\boldsymbol{x} \in \mathcal{X}_N} \mathbb{1}\big\{ \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{b}_i) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})\|_2 < \gamma \quad \text{and} \quad \|\boldsymbol{g}(\boldsymbol{b}_i) - \boldsymbol{g}(\boldsymbol{x})\|_2 \geq \varepsilon \big\}. \tag{10.84}$$

If $Z_{\mathcal{S}}(\boldsymbol{b}_i) = 0$ then every $\boldsymbol{x} \in \mathcal{X}_N$ with $\|\boldsymbol{g}(\boldsymbol{b}_i) - \boldsymbol{g}(\boldsymbol{x})\|_2 \geq \varepsilon$ also satisfies $\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{b}_i) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})\|_2 \geq \gamma$, otherwise $Z_{\mathcal{S}}(\boldsymbol{b}_i) = 1$. We observe that $Z_{\mathcal{S}}(\boldsymbol{b}_i)$, $i = 1, \ldots, m$ are independent, identically distributed Bernoulli random variables whose expectation

$$\mathbb{E}\left[Z_{\mathcal{S}}(\boldsymbol{b}_i)\right] = \mu\big(\big\{\boldsymbol{x} \in \mathcal{X} \,:\, \exists \boldsymbol{x}' \in \mathcal{X}_N \quad \text{s.t.} \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 < \gamma, \quad \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon \big\}\big) \tag{10.85}$$

is the μ-measure of points in $\mathcal{X}$ for which target values differing by at least ε with points of $\mathcal{X}_N$ are separated by measurements differing by less than γ. Suppose that for a fixed $\boldsymbol{x} \in \mathcal{X}$ we have

$$\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 \geq \varepsilon \quad \Rightarrow \quad \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2 \geq \gamma \tag{10.86}$$

for every $\boldsymbol{x}_j \in \mathcal{X}_N$. For any $\boldsymbol{x}' \in \mathcal{X}$, there is an $\boldsymbol{x}_j \in \mathcal{X}_N$ with $\|\boldsymbol{x}' - \boldsymbol{x}_j\|_2 < \varepsilon_0$ and so we have

$$\begin{aligned} \varepsilon + \varepsilon_0 \|\boldsymbol{g}\|_{\text{lip}} &\leq \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \\ &\leq \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 + \|\boldsymbol{g}(\boldsymbol{x}_j) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \\ &< \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 + \varepsilon_0 \|\boldsymbol{g}\|_{\text{lip}}. \end{aligned} \tag{10.87}$$

Hence, $\varepsilon \leq \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2$, which implies that $\gamma \leq \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2$ by assumption. From this we obtain

$$\begin{aligned} \gamma &\leq \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 + \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}') - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2 \\ &< \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 + \varepsilon_0 \|\boldsymbol{m}_{\mathcal{S}}\|_{\text{lip}}. \end{aligned} \tag{10.88}$$

Therefore, for such an $\boldsymbol{x} \in \mathcal{X}$ we have

$$\forall \boldsymbol{x}' \in \mathcal{X} \qquad \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon + \varepsilon_0 \|\boldsymbol{g}\|_{\text{lip}} \Rightarrow \quad \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 > \gamma - \varepsilon_0 \|\boldsymbol{m}_{\mathcal{S}}\|_{\text{lip}}. \quad (10.89)$$

It follows that $\mathbb{E}\left[Z_{\mathcal{S}}(\boldsymbol{b}_i)\right]$ is an upper bound on the μ-measure of points in $\mathcal{X}$ for which there is another point in $\mathcal{X}$ with a close measurement and distant target value, that is

$$\mathbb{E}\left[Z_{\mathcal{S}}(\boldsymbol{b}_i)\right] \geq \mu\big(\big\{\boldsymbol{x} \in \mathcal{X} \ : \ \exists \boldsymbol{x}' \in \mathcal{X} \quad \text{s.t.} \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 \leq \gamma - \varepsilon_0 \|\boldsymbol{m}_{\mathcal{S}}\|_{\text{lip}}, \\ \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon + \varepsilon_0 \|\boldsymbol{g}\|_{\text{lip}}\big\}\big). \quad (10.90)$$

By assumption, we have a set $\mathcal{S} \subset \mathcal{M}$ so that $Z_{\mathcal{S}}(\boldsymbol{b}_i) = 0$ for each $i = 1, \ldots, m$. And so it remains to bound the difference between the empirical and true expectation of $Z_{\mathcal{S}}(\boldsymbol{b}_i)$ uniformly over every subset $\mathcal{S} \subset \mathcal{M}$. For fixed $\mathcal{S}$, the one-sided Hoeffding inequality gives

$$\mathbb{P}\Big\{\frac{1}{m}\sum_{i=1}^{m}\left(\mathbb{E}[Z_{\mathcal{S}}(\boldsymbol{b}_i)] - Z_{\mathcal{S}}(\boldsymbol{b}_i)\right) \geq \delta\Big\} \leq e^{-2m\delta^2}. \quad (10.91)$$

Unfixing $\mathcal{S}$ via the union bound over all $\mathcal{S} \subset \mathcal{M}$ and applying our assumption about the number of base points m yields

$$\mathbb{P}\bigcup_{\mathcal{S}\subseteq\mathcal{M}}\Big\{\frac{1}{m}\sum_{i=1}^{m}\left(\mathbb{E}[Z_{\mathcal{S}}(\boldsymbol{b}_i)] - Z_{\mathcal{S}}(\boldsymbol{b}_i)\right) \geq \delta\Big\} \leq \exp\left[\#(\mathcal{M})\ln 2 - 2m\delta^2\right] \leq p. \quad (10.92)$$

Since our assumed choice of $\mathcal{S}$ has $f_m(\mathcal{S}) = f_m(\mathcal{M})$ it follows that all $Z_{\mathcal{S}}(\boldsymbol{b}_i) = 0$, $i = 1, \ldots, m$, hence we have

$$\mathbb{E}[Z_{\mathcal{S}}(\boldsymbol{b}_i)] < \delta \quad (10.93)$$

with probability at least $1 - p$. Combining this with Eq. **10.90** completes the proof. □

It is also possible to use a down-sampled objective to greedily choose sensors that satisfy a similarly relaxed version of the amplification guarantee given by Proposition 10.4.8 with high probability over a large subset of $\mathcal{X}$. In order to do this, we take the sum in Eq. **10.43** over secants between a randomly chosen collection of base points $\mathcal{B}_m = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\} \subseteq \mathcal{X}$ and the ε_0-net $\mathcal{X}_N$. Again, the number of base points depends on the quality of the guarantee and not on size of the ε_0-net, so that the computational cost can be reduced to linear dependence on the size of $\mathcal{X}_N$.

Specifically, in place of Eq. **10.43**, we consider

$$f_{L,m}(\mathcal{S}) = \sum_{\substack{1\le i\le m,\ 1\le j\le N,\\ \boldsymbol{g}(\boldsymbol{b}_i)\neq\boldsymbol{g}(\boldsymbol{x}_j)}} \min\left\{\frac{\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{b}_i)-\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2^2}{\|\boldsymbol{g}(\boldsymbol{b}_i)-\boldsymbol{g}(\boldsymbol{x}_j)\|_2^2},\ \frac{1}{L^2}\right\}. \tag{10.94}$$

In Theorem 10.5.7 we show that when a sufficiently small set of sensors $\mathcal{S}$ is found, e.g., using the greedy algorithm with the sampled objective $f_{L,m}$, that satisfies the amplification tolerance over $\mathcal{B}_m \times \mathcal{X}_N$, we can conclude that a slightly relaxed amplification bound holds with high probability over a large subset of $\mathcal{X}$. In particular, the subset of "bad points" in $\boldsymbol{x} \in \mathcal{X}$ for which there is another point $\boldsymbol{x}' \in \mathcal{X}$ with a different target value, but not a sufficiently different measured value, has small μ-measure with high probability.

Theorem 10.5.7 (Sampled Amplification Guarantee)**.** *Let $\mathcal{X}_N$ be an ε_0-net of $\mathcal{X}$ and let the base points $\mathcal{B}_m$ be sampled independently according to a probability measure μ on $\mathcal{X}$ with*

$$m \ge \frac{1}{2\delta^2}\left(\#(\mathcal{M})\ln 2 - \ln p\right). \tag{10.95}$$

*Consider the objective f_m given by Eq. **10.94** for a certain choice of $L > 0$ for which*

$$\|\boldsymbol{g}(\boldsymbol{b}_i)-\boldsymbol{g}(\boldsymbol{x}_j)\|_2 \le L\|\boldsymbol{m}_{\mathcal{M}}(\boldsymbol{b}_i)-\boldsymbol{m}_{\mathcal{M}}(\boldsymbol{x}_j)\|_2 \tag{10.96}$$

is achieved for all $\boldsymbol{b}_i \in \mathcal{B}_m$, $\boldsymbol{x}_j \in \mathcal{X}_N$. Suppose also that $\boldsymbol{g}$ and the measurement functions $\boldsymbol{m}_k$, $k \in \mathcal{M}$ are all Lipschitz functions over $\mathcal{X}$. If $f_{L,m}(\mathcal{S}) = f_{L,m}(\mathcal{M})$, then the μ-measure of points $\boldsymbol{x} \in \mathcal{X}$ such that

$$\|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{g}(\boldsymbol{x}')\|_2 < L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})+\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 + \left(\|\boldsymbol{g}\|_{lip} + L\|\boldsymbol{m}_{\mathcal{S}}\|_{lip}\right)\varepsilon_0 \tag{10.97}$$

for every $\boldsymbol{x}' \in \mathcal{X}$ is at least $1-\delta$ with probability at least $1-p$.

Proof. The proof is analogous to Theorem 10.5.6 and so we relegate it to Appendix 10.C. □

10.6 Working with Noisy Data

So far, we have considered maximizing different measures of robust reconstructability given a collection of noiseless data. That is, the resulting sensors are selected in order to be noise robust, but we have assumed that the measurements $\boldsymbol{m}_j(\boldsymbol{x}_i)$, $j \in \mathcal{M}$ and target variables $\boldsymbol{g}(\boldsymbol{x}_i)$ used during

the sensor selection process are noiseless over the sampled states $\boldsymbol{x}_i \in \mathcal{X}_N$. In many applications, however, our data may contain noisy measurements, target variables, or both. In this section, we study the effect of noisy data on the performance of our proposed secant-based greedy algorithms. By "noise" we mean specifically that we are given a collection of available measurements $\{\tilde{\boldsymbol{m}}_{i,\mathcal{M}} = \boldsymbol{m}_{\mathcal{M}}(\boldsymbol{x}_i) + \boldsymbol{u}_{i,\mathcal{M}}\}_{i=1}^N$ that are corrupted by unknown noise $\boldsymbol{u}_{i,\mathcal{M}}$ together with the corresponding target values $\{\tilde{\boldsymbol{g}}_i = \boldsymbol{g}(\boldsymbol{x}_i) + \boldsymbol{v}_i\}_{i=1}^N$ that are also corrupted by unknown noise $\boldsymbol{v}_i$. That is, we do not have access to the measurement functions $\boldsymbol{m}_{\mathcal{M}}$ or the target function $\boldsymbol{g}$ and must rely solely on noisy data generated by them.

First, we mention that the minimal sensing method to meet an error tolerance discussed in Section 10.4.2 is robust to bounded noise in the measurements and target variables. In particular, since the selected sensors $\mathcal{S}$ using the approach described in Section 10.4.2 automatically satisfy Eq. **10.99**, Proposition 10.6.1, below, shows that the true measurements coming from states with sufficiently distant true target values must also be separated by the measurements.

Proposition 10.6.1 (Noisy Separation Guarantee)**.** *Let $\mathcal{X}_N$ be an ε_0-net of $\mathcal{X}$ (see Definition 10.4.5) and let $\boldsymbol{v}_i \in \mathbb{R}^{\dim \boldsymbol{g}}$, $\boldsymbol{u}_{i,\mathcal{S}} \in \mathbb{R}^{d_{\mathcal{S}}}$, $i = 1, \ldots, N$ be bounded vectors with*

$$\forall i = 1, \ldots, N \qquad \|\boldsymbol{u}_{i,\mathcal{S}}\|_2 \leq \delta_u, \qquad \|\boldsymbol{v}_i\|_2 \leq \delta_v. \tag{10.98}$$

Suppose that there exists $\epsilon > 0$ and $\gamma > 0$ such that

$$\begin{aligned} \forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}_N \qquad & \| \left(\boldsymbol{g}(\boldsymbol{x}_i) + \boldsymbol{v}_i\right) - \left(\boldsymbol{g}(\boldsymbol{x}_j) + \boldsymbol{v}_j\right) \|_2 \geq \varepsilon \\ & \Rightarrow \quad \| \left(\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_i) + \boldsymbol{u}_{i,\mathcal{S}}\right) - \left(\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j) + \boldsymbol{u}_{j,\mathcal{S}}\right) \|_2 \geq \gamma. \end{aligned} \tag{10.99}$$

If $\boldsymbol{m}_{\mathcal{S}}$ and $\boldsymbol{g}$ are Lipschitz functions with Lipschitz constants $\|\boldsymbol{m}_{\mathcal{S}}\|_{lip}$ and $\|\boldsymbol{g}\|_{lip}$ respectively, then

$$\begin{aligned} \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X} \qquad & \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon + 2\delta_v + 2\varepsilon_0 \|\boldsymbol{g}\|_{lip} \\ & \Rightarrow \quad \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 > \gamma - 2\delta_u - 2\varepsilon_0 \|\boldsymbol{m}_{\mathcal{S}}\|_{lip}. \end{aligned} \tag{10.100}$$

Proof. The proof is analogous to Proposition 10.4.6 and has been relegated to Appendix 10.C. □

As a consequence of Proposition 10.6.1, the reconstruction error for the desired quantities using these sensors can still be bounded if the thresholds ε and γ exceed twice the noise level of the target variable and measurements respectively (with a little extra padding based on the sampling fineness).

On the other hand, the minimal sensing method to meet an amplification tolerance discussed in Section 10.4.3 is very sensitive to noisy data. This is because measurement noise can bring two nearby measurements $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})$ and $\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')$ arbitrarily close together while the corresponding target variables $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x}')$ remain separated. Such terms can result in arbitrarily large data-driven estimates of the reconstruction Lipschitz constant. Consequently it may not be possible to find a small set of sensors $\mathcal{S}$ such that

$$\max_{1\leq i<j\leq N} \frac{\left\|\tilde{\boldsymbol{g}}_i - \tilde{\boldsymbol{g}}_j\right\|_2}{\left\|\tilde{\boldsymbol{m}}_{i,\mathcal{S}} - \tilde{\boldsymbol{m}}_{j,\mathcal{S}}\right\|_2} \leq L \tag{10.101}$$

for acceptable values of L.

One way to deal with this problem is to smooth out the target variables. For instance, given the available noisy measurement and target pairs $\{(\tilde{\boldsymbol{m}}_{i,\mathcal{M}},\ \tilde{\boldsymbol{g}}_i)\}_{i=1}^N$, one can find an approximation of the reconstruction function $\boldsymbol{\Phi}_{\mathcal{M}}$ via regression. Using the predicted target variables

$$\hat{\boldsymbol{g}}_i := \boldsymbol{\Phi}_{\mathcal{M}}\left(\tilde{\boldsymbol{m}}_{i,\mathcal{M}}\right) \tag{10.102}$$

in place of the noisy data $\tilde{\boldsymbol{g}}_i$ fixes the problem of infinite Lipschitz constants. This is because the amplification-based approach using these data seeks to find the minimal set of sensors $\mathcal{S}$ such that

$$\max_{1\leq i<j\leq N} \frac{\left\|\hat{\boldsymbol{g}}_i - \hat{\boldsymbol{g}}_j\right\|_2}{\left\|\tilde{\boldsymbol{m}}_{i,\mathcal{S}} - \tilde{\boldsymbol{m}}_{j,\mathcal{S}}\right\|_2} \leq L \tag{10.103}$$

rather than satisfying Eq. **10.101**.

We use a similar type of smoothing approach for the shock-mixing layer problem by choosing the leading two Isomap coordinates $\boldsymbol{g}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}),\ \phi_2(\boldsymbol{x}))$ rather than simply taking $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$. This is because the full state $\boldsymbol{x}$ contains some small noise, meaning that it does not lie exactly on the one-dimensional loop in state space, but rather on a very thin manifold with full dimensionality. If we were to use the Lipschitz-based approach to reconstruct $\boldsymbol{x}$ directly, we would need enough sensors to reconstruct this noise. By seeking to reconstruct the leading Isomap coordinates instead, we have regularized our selection algorithm to choose only those sensors that are needed to reconstruct the dominant periodic behavior.

Reconstructing smoothed target variables turns out to be a robust method for sensor placement, as we show by introducing increasing levels of noise in the shock-mixing layer problem. We added independent Gaussian noise with standard deviations $\sigma_{\text{noise}} = 0.01$, 0.02, 0.03, 0.04, and 0.05 to each velocity component at every location on the computational grid, yielding noisy snapshots like

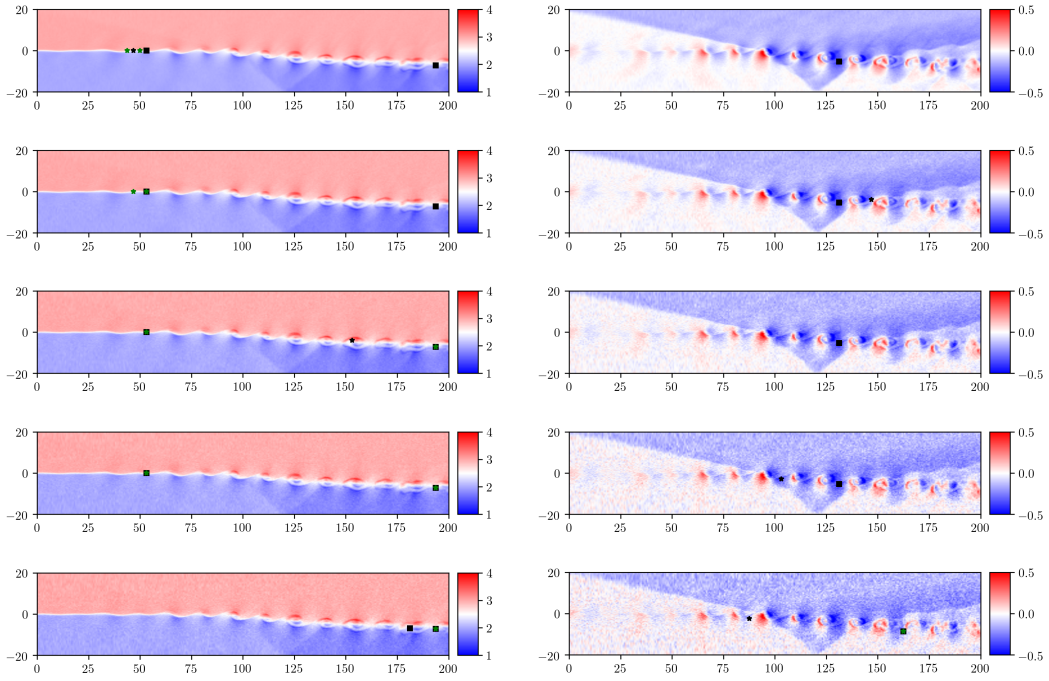


Figure 10.6.1: We show the stream-wise (first column) and transverse (second column) components of velocity for a single snapshot of the shock-mixing layer flow with increasing levels of noise added in each successive row. Independent Gaussian noise with standard deviations $\sigma_{\text{noise}} = 0.01$, 0.02, 0.03, 0.04, and 0.05 are added to each velocity component at each location on the computational grid. The first two sensors chosen by detectable difference method of Section 10.4.1 are indicated by green stars and the third is indicated by a black star. The three sensors selected using the amplification tolerance method of Section 10.4.3 with bisection search over L are indicated by black squares.

the one shown in Figure 10.6.1. This reflects the typical situation when the underlying data given to us are noisy. At each noise level we selected three sensors using the detectable difference-based method of Section 10.4.1 as well as the amplification tolerance-based method of Section 10.4.3, with a bisection search over the threshold Lipschitz constant L, to reconstruct the leading two Isomap coordinates of the noisy data. Despite the noise, the leading two Isomap coordinates continued to accurately capture the dominant periodic behavior of the underlying system, making them good reconstruction target variables. The thresholds for the detectable difference method were fixed at $\gamma = 0.04$ except in the $\sigma_{\text{noise}} = 0.02$ case, where better performance was achieved using $\gamma = 0.02$.

We found that the amplification tolerance-based method identified the same sensors across each of the first four noise levels $\sigma_{\text{noise}} = 0.01$, 0.02, 0.03, and 0.04. While these sensor locations differed slightly from the ones selected without noise (shown in Figure 10.3.1), they too were capable of robustly recovering the underlying phase of the system as illustrated by their corresponding measurements in the third column of Figure 10.6.2. At the largest noise level $\sigma_{\text{noise}} = 0.5$, the sensors selected using this method changed, but were still capable of revealing the phase as shown in the

bottom right plot of Figure 10.6.2. The detectable difference-based method selected the same three sensors as in the zero noise case when $\sigma_{\text{noise}} = 0.01$ with the first two remaining the same up to $\sigma_{\text{noise}} = 0.02$. At these noise levels the first two sensors are sufficient to reveal the underlying phase of the system as shown in the first two plots in the first column of Figure 10.6.2. Beyond this level of noise, the first two sensors were no longer able to reveal the phase as illustrated by the self-intersections in the last three plots in the first column of Figure 10.6.2. While it is admittedly difficult to see from the last three plots in the middle column of Figure 10.6.2, the third sensor eliminated these self-intersections by raising one of the two intersecting branches and allowing the phase to be determined.

10.7 Conclusion

In this paper we have identified a common type of nonlinear structure that causes techniques for sensor placement relying on linear reconstruction accuracy as an optimization criterion to consistently fail to identify minimal sets of sensors. Specifically, these techniques break down and lead to costly over-sensing when the data is intrinsically low dimensional, but is curved in such a way that energetic components are functions of less energetic ones, but not vice versa. This problem occurs commonly in fluid flows, period-doubling bifurcations in ecology and cardiology, as well as in spectral methods for manifold learning. We demonstrated that a representative collection of linear techniques fail to identify sensors from which the state of a shock-mixing layer flow can be reconstructed, and we provide a simple example that illustrates that the performance of the linear techniques can be arbitrarily bad. In addition, we demonstrated that it is impossible to use linear feature selection methods to choose fundamental nonlinear eigen-coordinates in manifold learning problems.

To remedy these issues, we proposed a new approach for sensor placement that relies on the information contained in secant vectors between data points to quantify nonlinear reconstructability of desired quantities from measurements. The resulting secant-based optimization problems turn out to have useful diminishing returns properties that enable efficient greedy approximation algorithms to achieve guaranteed high levels of performance. We also describe how down-sampling can be used to improve the computational scaling of these algorithms while still providing guarantees regarding the reconstructability of states in the underlying set from which the available data is sampled. Finally, these methods prove to be capable of selecting minimal collections of sensors in the shock-mixing layer problem as well as selecting the minimal set of fundamental manifold learning coordinates on

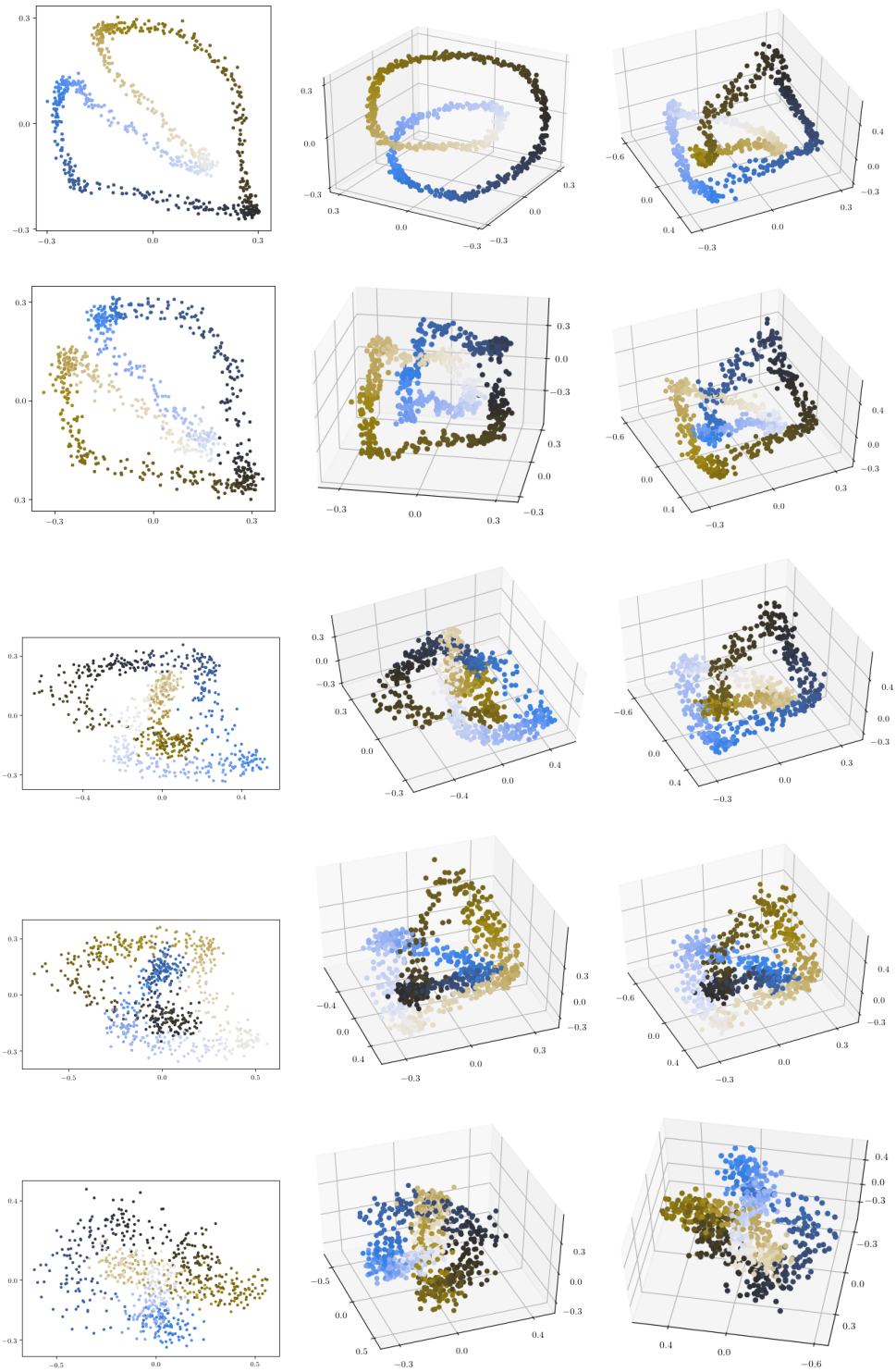


Figure 10.6.2: These plots show the measurements made by sensors selected using the detectable difference method of Section 10.4.1 with two (first column) and three (second column) sensors along with the amplification tolerance method of Section 10.4.3 with three sensors (third column) on the shock-mixing layer flow problem with various levels of added noise. Each row shows the result of adding independent Gaussian noise with standard deviations $\sigma_{\text{noise}} = 0.01$, 0.02, 0.03, 0.04, and 0.05 to each velocity component at each location on the computational grid.

a torus — both of which are problems where the linear techniques fail.

Acknowledgments

The authors would like to thank Gregory Blaisdell, Shih-Chieh Lo, Tasos Lyrintzis, and Kurt Aikens for providing the code used to compute the shock-mixing layer interaction. We also want to thank Alberto Padovan and Anastasia Bizyaeva for providing key references that motivate our main example, provide connections with period doubling, and reveal how linear methods can fail to find adequate sensor and actuator locations in real-world problems.

Appendix

10.A Implementation Details

10.A.1 Principal Component Analysis (PCA) and Isomap

In this paper, we used principal component analysis (PCA) [116] in order to find a modal basis for pivoted QR factorization and to identify a low-dimensional representation of the state and its covariance for determinantal D-optimal selection techniques on the shock-mixing layer flow. In order to perform PCA, one needs an appropriate inner product on the space in which the data lives. In the case of the shock-mixing layer problem, we use the energy-based inner product for compressible flows developed in [227] together with trapezoidal quadrature weights to approximate the integrals of the spatial fields over a stretched computational grid. In this problem, the data consists of vectors $\boldsymbol{z}$ whose elements are the streamwise velocity u, transverse velocity v, and the local speed of sound a over a 321×81 computational grid. The inner product between two snapshots $\boldsymbol{z}$ and $\boldsymbol{z}'$ is defined by

$$\begin{aligned}\langle \boldsymbol{z},\ \boldsymbol{z}' \rangle = \boldsymbol{z}^T \boldsymbol{W} \boldsymbol{z}' = \sum_{i=1}^{321} \sum_{j=1}^{81} w_{i,j} \left(u_{i,j}^2 + v_{i,j}^2 + a_{i,j}^2 \right) \\ \approx \int_\Omega \left[u(\xi_1, \xi_2)^2 + v(\xi_1, \xi_2)^2 + a(\xi_1, \xi_2)^2 \right] d\xi_1 d\xi_2, \quad (10.104)\end{aligned}$$

where the weights $\{w_{i,j}\}$ are selected to perform trapezoidal quadrature. Principal component analysis is performed by computing an economy-sized singular value decomposition of the mean-

subtracted data matrix

$$\tilde{\boldsymbol{U}}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{W}^{1/2}\left[(\boldsymbol{z}_1 - \bar{\boldsymbol{z}}) \quad \cdots \quad (\boldsymbol{z}_N - \bar{\boldsymbol{z}})\right], \quad \bar{\boldsymbol{z}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{z}_i \tag{10.105}$$

and forming the matrix of principal vectors $\boldsymbol{U} = \boldsymbol{W}^{-1/2}\tilde{\boldsymbol{U}}$. These vectors, making up the columns of $\boldsymbol{U}$, are orthonormal with respect to the $\boldsymbol{W}$-weighted inner product. If we represent the states in this basis so that $\boldsymbol{z}_i = \bar{\boldsymbol{z}} + \boldsymbol{U}\boldsymbol{x}_i$ then $\boldsymbol{x}$ has empirical covariance $\boldsymbol{C}_{\boldsymbol{x}} = \frac{1}{N}\boldsymbol{\Sigma}^2$.

The same weighted inner product was used to compute the distances between each data point $\boldsymbol{z}_i$ and its 10 nearest neighbors in order to compute the leading 50 Isomap coordinates using scikit learn's implementation found at `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html`.

10.A.2 (Group) LASSO

We use the Python implementation of group LASSO [296] by Yngve Mardal Moe at the University of Oslo that can be found at `https://group-lasso.readthedocs.io/en/latest/index.html`. We select among 2210 sensor measurements of u and v velocity components over a grid of 1105 spatial locations taken directly from the shock-mixing layer snapshot data. We tried two different kinds of target variables to be reconstructed via group LASSO. For the method we call "LASSO+PCA", the target variables were the data's leading 100 principal components which capture over 99% of the data's variance. For the method we call "LASSO+Isomap", the target variables were the leading two Isomap coordinates $\boldsymbol{g}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}))$, which reveal the phase angle θ. The sparsity-promoting regularization parameter was found using a bisection search in each case and was the smallest value, to within a tolerance of 10^{-5}, for which group LASSO selected 3 sensors.

10.A.3 Bayesian D-Optimal Selection

We use two different approaches for Bayesian D-optimal sensor placement: the greedy technique of [247] and the convex relaxation approach by [125]. In the greedy approach, we leverage the submodularity of the objective in the case when $\boldsymbol{T} = \boldsymbol{I}$ in order to use the accelerated greedy algorithm of M. Minoux [180]. For the convex approach, we wrote a direct Python translation of a MATLAB code written by S. Joshi and S. Boyd that implements a Newton method with line search, and may be found at `https://web.stanford.edu/~boyd/papers/matlab/sensor_selection/`. We use the gradient and Hessian matrices for the Bayesian D-optimal objective from their paper [125].

In both the greedy and convex approach for the shock-mixing layer problem, we take the state to be its representation using 100 principal components with covariance given by $\boldsymbol{C}_{\boldsymbol{x}} = \frac{1}{N}\boldsymbol{\Sigma}^2$ as computed by PCA. These principal components were also used as the relevant information to be reconstructed, i.e., $\boldsymbol{T} = \boldsymbol{I}$. The sensor noise was assumed to be isotropic with covariance $\boldsymbol{C}_{\boldsymbol{n}_s} = \sigma^2 \boldsymbol{I}_{d_s}$ with $\sigma = 0.02$. We tried many other values of σ, yielding different sensor locations, none of which could be used for nonlinear reconstruction. The ones we show at $\sigma = 0.02$ are representative.

10.A.4 Maximum Likelihood D-Optimal Selection

We used the maximum likelihood D-optimal selection technique based on convex relaxation found in [125] in order to choose sensors to try to reconstruct only the 3rd and 4th principal components of the shock-mixing layer snapshots. That is, if $\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots \end{bmatrix}$ is the matrix of principal components, we model the state as a linear combination of $\boldsymbol{u}_3$ and $\boldsymbol{u}_4$ together with isotropic Gaussian noise. We try to find the sensors so that the correct coefficients on $\boldsymbol{u}_3$ and $\boldsymbol{u}_4$ can be recovered with high confidence from the measurements. The rationale for doing so is the fact that these two components are sufficient to nonlinearly reconstruct the state of the system if they can be measured. As in Section 10.A.3 above, we use a direct Python translation of a MATLAB code written by S. Joshi and S. Boyd, which may be found at `https://web.stanford.edu/~boyd/papers/matlab/sensor_selection/`.

10.A.5 Pivoted QR Factorization

For the pivoted QR factorization method [82, 41] applied to the shock-mixing layer flow, we represent the state approximately as a linear combination of the leading three principal components. Scipy's implementation of pivoted QR factorization found at `https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.qr.html` was used to select among the 2210 allowable sensors those that allow robust reconstruction of these first three principal components. We also tried representing the state using more principal components and taking the first three sensor locations chosen via pivoted QR factorization. As with the case when only three principal components are used, these sensors do not enable nonlinear reconstruction of the state.

10.A.6 Secant-Based Detectable Differences

The secant-based detectable difference method was implemented using the accelerated greedy algorithm of M. Minoux [180] to optimize the objective computed over all secants between points

in the training data set consisting of $N = 750$ snapshots of the shock-mixing layer velocity field. We select among the 2210 sensor measurements of u and v velocity components on a grid of 1105 spatial locations taken directly from the shock-mixing layer snapshot data. The target variables were chosen to be the leading two Isomap coordinates $\boldsymbol{g}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}))$, which reveal the phase angle θ. The greedy algorithm first reveals the two sensor locations marked by green stars and then the black star in Figure 10.3.1 over the range of $0.02 \leq \gamma \leq 0.06$, which can be used to reveal the exact phase of the system. Choosing smaller values of γ produce different sensors that can also be used to reveal the phase, but with reduced robustness to measurement perturbations. Gaussian process regression [216] was used to reconstruct the leading 100 principal components of the flowfields from the sensor measurements. We used scikit learn's implementation which can be found at `https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html` together with a Matérn and white noise kernel whose parameters were optimized during the fit.

For the torus example, the relevant information we wish to reconstruct are the leading 100 Isomap eigen-coordinates $\boldsymbol{g}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_{100}(\boldsymbol{x}))$ computed from 2000 points sampled from the torus according to Eq. **10.11**. The objective function was evaluated using secants between $\#(\mathcal{B}) = 100$ randomly sampled base points and the original set of $N = 2000$ points. The correct three coordinates ϕ_1, ϕ_2, ϕ_7 are selected from among the first 100 consistently across a wide range of measurement separation values $0.05 \leq \gamma \leq 3.0$. We note that these values vary slightly with the selected base points and these particular values hold only for one instance.

10.A.7 Secant-Based Amplification Tolerance

Like the secant-based detectable difference method described above, the secant-based amplification tolerance method was implemented using the same data, secant vectors, and target variables with the accelerated greedy algorithm. A bisection search was used to find the smallest Lipschitz constant $L = 1868$ to within a tolerance of 1 for which the algorithm selects three sensors on the shock-mixing layer flow. Three (different) sensors that correctly reveal the state of the flow are selected by this algorithm over a range $1868 \leq L \leq 47624$, above which only two sensors that cannot reveal the state are selected. We also find that with $L = 129$, the minimum possible number of sensors exceeds $\#(\mathcal{S}_K)/(1 + \ln \kappa) = 3.18 > 3$. Therefore, the minimum possible reconstruction Lipschitz constant using three sensors that one might find by an exhaustive combinatorial search must be greater than 129. We admit that this is likely a rather pessimistic bound, but we cannot check it as there are

$\binom{2210}{3} \approx 1.8 \times 10^9$ possible choices for three sensors in this problem.

When applied to select from among the leading 100 Isomap eigen-coordinates on the torus example with the same setup as the secant-based detectable differences method, the amplification tolerance method selects the appropriate collection ϕ_1, ϕ_2, ϕ_7 over the range $7.1 \leq L \leq 25$. We note that these value vary slightly with the selected base points and these particular values hold only for one instance.

10.B Submodularity of Objectives

We will need the definition of a modular function given below.

Definition 10.B.1 (Modular Function)**.** *Denote the set of all subsets of* $\mathcal{M}$ *by* $2^{\mathcal{M}}$*. A real-valued function of the subsets* $f : 2^{\mathcal{M}} \to \mathbb{R}$ *is called "modular" when it can be written as a sum*

$$f(\mathcal{S}) = \sum_{j \in \mathcal{S}} a_j \tag{10.106}$$

of constants a_j*,* $j \in \mathcal{M}$*.*

The key ingredient needed to prove submodularity for the objectives described in Section 10.4 is the following lemma.

Lemma 10.B.2 (Concave Composed with Modular is Submodular)**.** *Let* $h : \mathbb{R} \to \mathbb{R}$ *be a concave function and let* $a : 2^{\mathcal{M}} \to \mathbb{R}$ *defined by*

$$a(\mathcal{S}) = \sum_{j \in \mathcal{S}} a_j \tag{10.107}$$

be a modular function (Def. 10.B.1) of subsets $\mathcal{S} \subseteq \mathcal{M}$ *with* $a_j \geq 0$ *for all* $j \in \mathcal{M}$*. Then the function* $f : 2^{\mathcal{M}} \to \mathbb{R}$ *defined by*

$$f(\mathcal{S}) = h(a(\mathcal{S})) \tag{10.108}$$

is submodular.

Proof. Suppose that $\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{M} \setminus \{j\}$. By concavity of h we have

$$h_\alpha = h((1-\alpha)a(\mathcal{S}) + \alpha(a(\mathcal{S}') + a_j)) \geq (1-\alpha)h_0 + \alpha h_1 \tag{10.109}$$

for every $\alpha \in [0, 1]$, where we note that $h_0 = f(\mathcal{S})$ and $h_1 = f(\mathcal{S}' \cup \{j\})$.

Since $\{a_l\}$ are non-negative we have $a(\mathcal{S}) \leq a(\mathcal{S}) + a_j \leq a(\mathcal{S}') + a_j$ and $a(\mathcal{S}) \leq a(\mathcal{S}') \leq a(\mathcal{S}') + a_j$. We can therefore find

$$\alpha_1 = \frac{a_j}{a(\mathcal{S}') + a_j - a(\mathcal{S})}, \quad \alpha_2 = \frac{a(\mathcal{S}') - a(\mathcal{S})}{a(\mathcal{S}') + a_j - a(\mathcal{S})} \tag{10.110}$$

so that $h_{\alpha_1} = f(\mathcal{S} \cup \{j\})$ and $h_{\alpha_2} = f(\mathcal{S}')$. Note that $\alpha_1 + \alpha_2 = 1$.

We now use Eq. **10.109** at α_1 and α_2 to bound the increments of f:

$$f(\mathcal{S} \cup \{j\}) - f(\mathcal{S}) = h_{\alpha_1} - h_0 \geq \alpha_1(h_1 - h_0), \tag{10.111}$$

$$f(\mathcal{S}' \cup \{j\}) - f(\mathcal{S}') = h_1 - h_{\alpha_2} \leq (1 - \alpha_2)(h_1 - h_0) \tag{10.112}$$

Combining the bounds Eq. **10.111** and Eq. **10.112** on the increments using $1 - \alpha_2 = \alpha_1$ we conclude that f is submodular

$$f(\mathcal{S} \cup \{j\}) - f(\mathcal{S}) \geq f(\mathcal{S}' \cup \{j\}) - f(\mathcal{S}'). \tag{10.113}$$

□

Using Lemma 10.B.2 it suffices to observe that each of the objectives described in Section 10.4 can be written as the composition of a concave function and a modular function. We carry this out below in addition to proving normalization and monotonicity for these objectives.

Lemma 10.B.3 (Detectable Difference Objective is Submodular)**.** *Suppose that the target variables $\boldsymbol{g}$ and measurements $\boldsymbol{m}_j$, $j \in \mathcal{M}$ are measurable functions. If μ and ν are measures on $\mathcal{X}$, then the function defined by*

$$f(\mathcal{S}) = \int_{\substack{(\boldsymbol{x},\boldsymbol{x}') \in \mathcal{X} \times \mathcal{X}: \\ \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \geq \varepsilon}} w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2 \, d\mu(\boldsymbol{x}) \nu(d\boldsymbol{x}'), \tag{10.114}$$

for any $\varepsilon \geq 0$ with

$$w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) = \min\left\{ \frac{1}{\gamma^2} \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2^2, \ 1 \right\}, \tag{10.115}$$

is normalized so that $f(\emptyset) = 0$, monotone non-decreasing so that $\mathcal{S} \subseteq \mathcal{S}' \ \Rightarrow \ f(\mathcal{S}) \leq f(\mathcal{S}')$, and submodular (Def. 10.4.3).

Proof. Normalization is obvious. It suffices to prove that the function $w_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})$ is monotone and

submodular for any fixed $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$. For if we suppose that

$$\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{M} \setminus \{j\} \quad \Rightarrow \quad w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S} \cup \{j\}) - w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \geq w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}' \cup \{j\}) - w_{\gamma,\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}'), \quad (10.116)$$

then multiplying both sides of the inequality by $\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2$ and integrating proves that f is submodular. The same argument also proves monotonicity.

Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ be fixed. The squared separation between the measurements is given by a modular (Def. 10.B.1) sum

$$\mathcal{S} \;\mapsto\; \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2^2 = \sum_{j \in \mathcal{S}} \|\boldsymbol{m}_j(\boldsymbol{x}) - \boldsymbol{m}_j(\boldsymbol{x}')\|_2^2 \quad (10.117)$$

of non-negative constants $\|\boldsymbol{m}_j(\boldsymbol{x}) - \boldsymbol{m}_j(\boldsymbol{x}')\|_2^2$ over each $j \in \mathcal{S}$. Since $x \mapsto \min\{x/\gamma^2,\ 1\}$ is a non-decreasing function, it follows that $\mathcal{S} \subseteq \mathcal{S}' \;\Rightarrow\; w_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \leq w_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}')$, proving monotonicity.

Submodularity of $w_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})$ follows from Lemma 10.B.2 since $w_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})$ is the composition of a concave function $x \mapsto \min\{x/\gamma^2,\ 1\}$ with the modular function in Eq. **10.117**. □

Lemma 10.B.4 (Lipschitz Objective is Submodular)**.** *Suppose that the target variables $\boldsymbol{g}$ and measurements $\boldsymbol{m}_j$, $j \in \mathcal{M}$ are measurable functions. If μ and ν are measures on $\mathcal{X}$, then the function defined by*

$$f(\mathcal{S}) = \int_{\substack{(\boldsymbol{x},\boldsymbol{x}') \in \mathcal{X}\times\mathcal{X}: \\ \boldsymbol{g}(\boldsymbol{x}) \neq \boldsymbol{g}(\boldsymbol{x}')}} g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})\ d\mu(\boldsymbol{x})\nu(d\boldsymbol{x}'), \quad (10.118)$$

with

$$g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) = \min\left\{ \frac{\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2^2}{\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2},\ \frac{1}{L^2} \right\}, \quad (10.119)$$

is normalized so that $f(\emptyset) = 0$, monotone non-decreasing so that $\mathcal{S} \subseteq \mathcal{S}' \;\Rightarrow\; f(\mathcal{S}) \leq f(\mathcal{S}')$, and submodular (Def. 10.4.3).

Proof. Normalization is obvious. It suffices to prove that the function $g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})$ is monotone and submodular for any fixed $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$. For if we suppose that

$$\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{M} \setminus \{j\} \quad \Rightarrow \quad g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S} \cup \{j\}) - g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \geq g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}' \cup \{j\}) - g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}'), \quad (10.120)$$

then integrating both sides of the inequality proves that f is submodular. The same argument also proves monotonicity.

Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ be fixed. The squared separation between the measurements is given by a modular

(Def. 10.B.1) sum

$$\mathcal{S} \;\mapsto\; \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2^2 = \sum_{j\in\mathcal{S}} \|\boldsymbol{m}_j(\boldsymbol{x}) - \boldsymbol{m}_j(\boldsymbol{x}')\|_2^2 \tag{10.121}$$

of non-negative constants $\|\boldsymbol{m}_j(\boldsymbol{x}) - \boldsymbol{m}_j(\boldsymbol{x}')\|_2^2$ over each $j \in \mathcal{S}$. Since

$$x \;\mapsto\; \min\left\{ \frac{x}{\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2^2},\; \frac{1}{L^2} \right\} \tag{10.122}$$

is a non-decreasing function, it follows that $\mathcal{S} \subseteq \mathcal{S}' \;\Rightarrow\; g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}) \leq g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S}')$, proving monotonicity.

Submodularity of $g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})$ follows from Lemma 10.B.2 since $g_{\boldsymbol{x},\boldsymbol{x}'}(\mathcal{S})$ is the composition of the concave function in Eq. **10.122** with the modular function in Eq. **10.121**. □

10.C Proofs

Proposition 10.4.6: Separation Guarantee on Underlying Set. The result follows immediately from the triangle inequality. Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}_N$ so that $\|\boldsymbol{x} - \boldsymbol{x}_i\|_2 < \varepsilon_0$ and $\|\boldsymbol{x}' - \boldsymbol{x}_j\|_2 < \varepsilon_0$. Then $\varepsilon + 2\varepsilon_0\|\boldsymbol{g}\|_{\text{lip}} \leq \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2$ implies that

$$\begin{aligned} \varepsilon + 2\varepsilon_0\|\boldsymbol{g}\|_{\text{lip}} &\leq \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \\ &\leq \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}_i)\|_2 + \|\boldsymbol{g}(\boldsymbol{x}') - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 + \|\boldsymbol{g}(\boldsymbol{x}_i) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 \\ &< \|\boldsymbol{g}(\boldsymbol{x}_i) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 + 2\varepsilon_0\|\boldsymbol{g}\|_{\text{lip}}, \end{aligned} \tag{10.123}$$

hence, $\|\boldsymbol{g}(\boldsymbol{x}_i) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 \geq \varepsilon$. By assumption, this implies that $\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_i) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2 \geq \gamma$ and

$$\begin{aligned} \gamma &\leq \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_i) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2 \\ &\leq \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_i) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})\|_2 + \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}') - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2 + \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 \\ &< 2\varepsilon_0\|\boldsymbol{m}_{\mathcal{S}}\|_{\text{lip}} + \|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2, \end{aligned} \tag{10.124}$$

hence, $\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 > \gamma - 2\varepsilon_0\|\boldsymbol{m}_{\mathcal{S}}\|_{\text{lip}}$ as claimed. □

Proposition 10.4.8: Amplification Guarantee on Underlying Set. The result follows immediately from the triangle inequality. Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}_N$ so that $\|\boldsymbol{x} - \boldsymbol{x}_i\|_2 < \varepsilon_0$ and $\|\boldsymbol{x}' - \boldsymbol{x}_j\|_2 < \varepsilon_0$,

then

$$\begin{aligned}\|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{g}(\boldsymbol{x}')\|_2 &\leq \|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{g}(\boldsymbol{x}_i)\|_2+\|\boldsymbol{g}(\boldsymbol{x}')-\boldsymbol{g}(\boldsymbol{x}_j)\|_2+\|\boldsymbol{g}(\boldsymbol{x}_i)-\boldsymbol{g}(\boldsymbol{x}_j)\|_2\\ &< 2\varepsilon_0\|\boldsymbol{g}\|_{\text{lip}}+L\|\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_i)-\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_j)\|_2\\ &\leq 2\varepsilon_0\|\boldsymbol{g}\|_{\text{lip}}+L\|\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x})-\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_i)\|_2+L\|\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}')-\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_j)\|_2\\ &\qquad +L\|\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x})-\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}')\|_2\\ &< 2\varepsilon_0\|\boldsymbol{g}\|_{\text{lip}}+2L\varepsilon_0\|\boldsymbol{m}_{\mathbb{S}}\|_{\text{lip}}+L\|\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x})-\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}')\|_2.\end{aligned} \tag{10.125}$$

Gathering terms on ε_0 completes the proof. □

Proposition 10.6.1: Noisy Separation Guarantee. Choose $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}_N$ and suppose that

$$\|\boldsymbol{g}(\boldsymbol{x}_i)-\boldsymbol{g}(\boldsymbol{x}_j)\|_2 \geq \varepsilon+2\delta_v. \tag{10.126}$$

Then we have

$$\begin{aligned}\|\left(\boldsymbol{g}(\boldsymbol{x}_i)+\boldsymbol{v}_i\right)-\left(\boldsymbol{g}(\boldsymbol{x}_j)+\boldsymbol{v}_j\right)\|_2 &\geq \|\boldsymbol{g}(\boldsymbol{x}_i)-\boldsymbol{g}(\boldsymbol{x}_j)\|_2-\|\boldsymbol{v}_i\|-\|\boldsymbol{v}_j\|\\ &\geq \|\boldsymbol{g}(\boldsymbol{x}_i)-\boldsymbol{g}(\boldsymbol{x}_j)\|_2-2\delta_v\\ &\geq \varepsilon\end{aligned} \tag{10.127}$$

By our assumption, this implies

$$\|\left(\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_i)+\boldsymbol{u}_{i,\mathbb{S}}\right)-\left(\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_j)+\boldsymbol{u}_{j,\mathbb{S}}\right)\|_2 \geq \gamma, \tag{10.128}$$

and so we have

$$\begin{aligned}\|\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_i)-\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_j)+\boldsymbol{u}_{j,\mathbb{S}}\|_2 &\geq \|\left(\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_i)+\boldsymbol{u}_{i,\mathbb{S}}\right)-\left(\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_j)+\boldsymbol{u}_{j,\mathbb{S}}\right)\|_2-\|\boldsymbol{u}_{i,\mathbb{S}}\|-\|\boldsymbol{u}_{j,\mathbb{S}}\|\\ &\geq \gamma-2\delta_u.\end{aligned} \tag{10.129}$$

Therefore, we have established that

$$\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}_N \qquad \|\boldsymbol{g}(\boldsymbol{x}_i)-\boldsymbol{g}(\boldsymbol{x}_j)\|_2 \geq \varepsilon+2\delta_v \Rightarrow \quad \|\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_i)-\boldsymbol{m}_{\mathbb{S}}(\boldsymbol{x}_j)+\boldsymbol{u}_{j,\mathbb{S}}\|_2 \geq \gamma-2\delta_u. \tag{10.130}$$

The conclusion follows immediately by Proposition 10.4.6. □

Theorem 10.5.7: Down-Sampled Amplification Guarantee. For simplicity, we will drop L from the subscript on our objective since the threshold L for the Lipschitz constant remains fixed throughout the proof. Let us begin by fixing a set $\mathcal{S} \subseteq \mathcal{M}$ and define the random variables

$$Z_{\mathcal{S}}(\boldsymbol{b}_i) = \max_{\boldsymbol{x} \in \mathcal{X}_N} \mathbb{1}\big\{\|\boldsymbol{g}(\boldsymbol{b}_i) - \boldsymbol{g}(\boldsymbol{x})\|_2 > L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{b}_i) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})\|_2\big\}, \tag{10.131}$$

for $i = 1, \ldots, m$. If $Z_{\mathcal{S}}(\boldsymbol{b}_i) = 0$ then every secant between $\boldsymbol{b}_i$ and points of $\mathcal{X}_N$ satisfies the desired bound on the amplification. Otherwise, there is some point $\boldsymbol{x} \in \mathcal{X}$ for which

$$\|\boldsymbol{g}(\boldsymbol{b}_i) - \boldsymbol{g}(\boldsymbol{x})\|_2 > L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{b}_i) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x})\|_2 \tag{10.132}$$

and so $Z_{\mathcal{S}}(\boldsymbol{b}_i) = 1$. We observe that $Z_{\mathcal{S}}(\boldsymbol{b}_1), \ldots, Z_{\mathcal{S}}(\boldsymbol{b}_m)$ are independent, identically distributed Bernoulli random variables whose expectation

$$\mathbb{E}[Z_{\mathcal{S}}(\boldsymbol{b}_i)] = \mu\big(\big\{\boldsymbol{x} \in \mathcal{X} \;:\; \exists \boldsymbol{x}_j \in \mathcal{X}_N \quad \text{s.t.} \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 > L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2\big\}\big) \tag{10.133}$$

is the μ-measure of points in $\mathcal{X}$ that are not adequately separated from points in the ε_0-net $\mathcal{X}_N$ by the measurements $\boldsymbol{m}_{\mathcal{S}}$. Suppose that for a fixed $\boldsymbol{x} \in \mathcal{X}$ we have

$$\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 \leq L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2 \tag{10.134}$$

for every $\boldsymbol{x}_j \in \mathcal{X}_N$. By definition of $\mathcal{X}_N$, for any $\boldsymbol{x}' \in \mathcal{X}$, there is an $\boldsymbol{x}_j \in \mathcal{X}_N$ with $\|\boldsymbol{x}' - \boldsymbol{x}_j\|_2 < \varepsilon_0$ and so we have

$$\begin{aligned}
\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 &\leq \|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}_j)\|_2 + \|\boldsymbol{g}(\boldsymbol{x}_j) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \\
&< L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2 + \varepsilon_0\|\boldsymbol{g}\|_{\text{lip}} \\
&\leq L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 + L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}') - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}_j)\|_2 + \varepsilon_0\|\boldsymbol{g}\|_{\text{lip}} \\
&< L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 + (\|\boldsymbol{g}\|_{\text{lip}} + L\|\boldsymbol{m}_{\mathcal{S}}\|_{\text{lip}})\,\varepsilon_0.
\end{aligned} \tag{10.135}$$

It follows that $\mathbb{E}[Z_{\mathcal{S}}(\boldsymbol{b}_i)]$ is an upper bound on the μ-measure of points in $\mathcal{X}$ for which the relaxed amplification threshold is exceeded, that is,

$$\begin{aligned}
\mathbb{E}[Z_{\mathcal{S}}(\boldsymbol{b}_i)] \geq \mu\big(\big\{\boldsymbol{x} \in \mathcal{X} \;:\; \exists \boldsymbol{x}' \in \mathcal{X} \quad \text{s.t.} \quad &\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x}')\|_2 \\
&\geq L\|\boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}) - \boldsymbol{m}_{\mathcal{S}}(\boldsymbol{x}')\|_2 + (\|\boldsymbol{g}\|_{\text{lip}} + L\|\boldsymbol{m}_{\mathcal{S}}\|_{\text{lip}})\,\varepsilon_0\big\}\big).
\end{aligned} \tag{10.136}$$

By assumption, we have a set $\mathcal{S} \subseteq \mathcal{M}$ so that $Z_{\mathcal{S}}(\boldsymbol{b}_i) = 0$ for each $i = 1, \ldots, m$. And so it remains to bound the difference between the empirical and true expectation of $Z_{\mathcal{S}}(\boldsymbol{b}_i)$ uniformly over every subset $\mathcal{S} \subseteq \mathcal{M}$. For fixed $\mathcal{S}$, the one-sided Hoeffding inequality gives

$$\mathbb{P}\Big\{\frac{1}{m}\sum_{i=1}^{m}\left(\mathbb{E}[Z_{\mathcal{S}}(\boldsymbol{b}_i)] - Z_{\mathcal{S}}(\boldsymbol{b}_i)\right) \geq \delta\Big\} \leq e^{-2m\delta^2}. \tag{10.137}$$

Unfixing $\mathcal{S}$ via the union bound over all $\mathcal{S} \subseteq \mathcal{M}$ and applying our assumption about the number of base points m yields

$$\mathbb{P}\bigcup_{\mathcal{S}\subseteq\mathcal{M}}\Big\{\frac{1}{m}\sum_{i=1}^{m}\left(\mathbb{E}[Z_{\mathcal{S}}(\boldsymbol{b}_i)] - Z_{\mathcal{S}}(\boldsymbol{b}_i)\right) \geq \delta\Big\} \leq e^{\#(\mathcal{M})\ln 2 - 2m\delta^2} \leq p. \tag{10.138}$$

Since our assumed choice of $\mathcal{S}$ has $f_m(\mathcal{S}) = f_m(\mathcal{M})$ it follows that all $Z_{\mathcal{S}}(\boldsymbol{b}_i) = 0$, $i = 1, \ldots, m$, hence we have

$$\mathbb{E}[Z_{\mathcal{S}}(\boldsymbol{b}_i)] < \delta \tag{10.139}$$

with probability at least $1 - p$. Combining this with Eq. **10.136** completes the proof. □

10.D Description of the Accelerated Greedy Algorithm

Since each objective function f presented in Section 10.4 is submodular, it is possible to use an "accelerated greedy" (AG) algorithm to obtain the same solution as the naive greedy algorithm with a provably minimal number of objective function evaluations compared to a broad class of algorithms [180]. Let the increase in the objective function obtained by adding the sensor j to the set $\mathcal{S}$ be called $\Delta_j(\mathcal{S}) = f(\mathcal{S} \cup \{j\}) - f(\mathcal{S})$. Instead of evaluating $\Delta_j(\mathcal{S}_{k-1})$ for every measurement in $\mathcal{M} \backslash \mathcal{S}_{k-1}$, AG keeps track of an upper bound $\hat{\Delta}_j \geq \Delta_j(\mathcal{S}_{k-1})$ on the increments for each sensor. Since submodularity of f means that the increments $\Delta_j(\mathcal{S})$ can only decrease as the size of $\mathcal{S}$ increases, it is sufficient to have the maximum upper bound $\hat{\Delta}_{j^*} \geq \hat{\Delta}_j$, $\forall j \in \mathcal{M} \setminus \mathcal{S}_{k-1}$ be tight $\hat{\Delta}_{j^*} = \Delta_{j^*}(\mathcal{S}_{k-1})$ in order to conclude that $\Delta_{j^*}(\mathcal{S}_{k-1})$ is the largest increment. The rest of the upper bounds on the increments can remain loose since they are smaller than the tight maximum upper bound. The AG algorithm finds largest upper bound $\hat{\Delta}_{j^*}$ and updates it so that it is tight. If $\hat{\Delta}_{j^*}$ is still the greatest upper bound, then $j^* = j_k$ achieves the largest increment and is added to $\mathcal{S}_{k-1}$. Otherwise if $\hat{\Delta}_{j^*}$ is no longer the largest upper bound, the new largest upper bound is selected and process repeated until a tight maximum upper bound is obtained.

Bibliography

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

[2] P.-A. Absil and J. Malick. Projection-like retractions on matrix manifolds. *SIAM J. Optim.*, 22(1):135–158, 2012.

[3] D. Achlioptas. Database-friendly random projections: Johnson–Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.

[4] R. L. Adler, J.-P. Dedieu, J. Y. Margulies, M. Martens, and M. Shub. Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA J. Numer. Anal.*, 22(3):359–390, 2002.

[5] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

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

[7] A. H. Al-Mohy and N. J. Higham. Computing the fréchet derivative of the matrix exponential, with an application to condition number estimation. *SIAM Journal on Matrix Analysis and Applications*, 30(4):1639–1657, 2009.

[8] D. Amsallem, M. J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, 2012.

[9] T. Ando and F. Hiai. Operator log-convex functions and operator means. *Mathematische Annalen*, 350(3):611–630, 2011.

[10] A. C. Antoulas. *Approximation of large-scale dynamical systems*. SIAM, 2005.

[11] H. Arbabi, M. Korda, and I. Mezić. A data-driven Koopman model predictive control framework for nonlinear partial differential equations. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6409–6414. IEEE, 2018.

[12] R. G. Baraniuk and M. B. Wakin. Random projections of smooth manifolds. *Foundations of computational mathematics*, 9(1):51–77, 2009.

[13] A. Barbagallo, D. Sipp, and P. J. Schmid. Closed-loop control of an open cavity flow using reduced-order models. *J. Fluid Mech.*, 641:1, 2009.

[14] U. Baur, P. Benner, and L. Feng. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. *Arch. Comput. Meth. Eng.*, 21(4):331–358, 2014.

[15] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[16] T. Bendokat, R. Zimmermann, and P.-A. Absil. A Grassmann manifold handbook: Basic geometry and computational aspects. *arXiv preprint arXiv:2011.13699*, 2020.

[17] P. Benner and T. Breiten. Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems. *SIAM J. Matrix Anal. Appl.*, 33(3):859–885, 2012.

[18] P. Benner and T. Breiten. Two-sided projection methods for nonlinear model order reduction. *SIAM J. Sci. Comput.*, 37(2):B239–B260, 2015.

[19] P. Benner, P. Goyal, and S. Gugercin. $\mathcal{H}_2$-quasi-optimal model order reduction for quadratic-bilinear control systems. *SIAM J. Matrix Anal. Appl.*, 39(2):983–1032, 2018.

[20] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.*, 57(4):483–531, 2015.

[21] J. L. Bentley. A survey of techniques for fixed radius near neighbor searching. Technical report, Stanford University, Stanford, CA, USA, 1975.

[22] J. L. Bentley, D. F. Stanat, and E. H. Williams Jr. The complexity of finding fixed-radius near neighbors. *Information Processing Letters*, 6(6):209–212, 1977.

[23] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.*, 25(1):539–575, 1993.

[24] A. A. Bian, J. M. Buhmann, A. Krause, and S. Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 498–507, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[25] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, 2001.

[26] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[27] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[28] S. Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Trans. Automat. Contr.*, 58(9):2217–2229, 2013.

[29] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[30] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

[31] D. Broomhead and M. Kirby. Dimensionality reduction using secant-based projection methods: The induced dynamics in projected systems. *Nonlinear Dynamics*, 41(1-3):47–67, 2005.

[32] G. L. Brown and A. Roshko. On density effects and large structure in turbulent mixing layers. *J. Fluid Mech.*, 64(4):775–816, 1974.

[33] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2), 2016.

[34] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz. Modern Koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*, 2021.

[35] S. L. Brunton and J. N. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[36] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Nat. Acad. Sci.*, 113(15):3932–3937, 2016.

[37] M. Budišić, R. Mohr, and I. Mezić. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.

[38] M. Budišić, R. Mohr, and I. Mezić. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.

[39] J. V. Burke. Line search methods. Lecture notes for MATH 408, Nonlinear Optimization, University of Washington, published online at `https://sites.math.washington.edu/~burke/crs/408/notes/nlp/line.pdf`, 2004.

[40] J. V. Burke. Continuity and differentiability of solutions. Lecture notes for MATH 555, Linear Analysis, University of Washington, published online at `https://sites.math.washington.edu/~burke/crs/555/555_notes/continuity.pdf`, 2015.

[41] P. Businger and G. H. Golub. Linear least squares solutions by householder transformations. *Numerische Mathematik*, 7(3):269–276, 1965.

[42] M. Y. Byron, K. V. Shenoy, and M. Sahani. Derivation of Kalman filtering and smoothing equations. In *Technical report*. Stanford University, 2004.

[43] J. L. Callaham, K. Maeda, and S. L. Brunton. Robust flow reconstruction from limited measurements via sparse representation. *Physical Review Fluids*, 4(10):103907, 2019.

[44] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592, 2008.

[45] E. J. Candès and Y. Plan. A probabilistic and RIPless theory of compressed sensing. *IEEE transactions on information theory*, 57(11):7235–7254, 2011.

[46] E. J. Candès, Y. Plan, et al. Near-ideal model selection by ℓ_1 minimization. *The Annals of Statistics*, 37(5A):2145–2177, 2009.

[47] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[48] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[49] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006.

[50] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[51] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ^1 minimization. *Journal of Fourier analysis and applications*, 14(5):877–905, 2008.

[52] E. Carlen. Trace inequalities and quantum entropy: an introductory course. *Entropy and the quantum*, 529:73–140, 2010.

[53] R. Caron and T. Traynor. The zero set of a polynomial. *WSMR Report*, 2005.

[54] W. F. Caselton, L. Kan, and J. V. Zidek. Quality data networks that minimize entropy. In *Statistics in the Environmental and Earth Sciences*, pages 10–38. Halsted Press, 1992.

[55] W. F. Caselton and J. V. Zidek. Optimal monitoring network designs. *Statistics & Probability Letters*, 2(4):223–227, 1984.

[56] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[57] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[58] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *2008 IEEE international conference on acoustics, speech and signal processing*, pages 3869–3872. IEEE, 2008.

[59] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, pages 808–817, 2000.

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

[61] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[62] Y. Chen. Compressed sensing and sparse recovery. Lecture notes for ELE 520: Mathematics of Data Science, Princeton University, published online at `https://yuxinchen2020.github.io/ele520_math_data/lectures/compressed_sensing.pdf`, 2020.

[63] E. Chiavazzo, C. W. Gear, C. J. Dsilva, N. Rabin, and I. G. Kevrekidis. Reduced models in chemical kinetics via nonlinear data-mining. *Processes*, 2(1):112–140, 2014.

[64] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28:2980–2988, 2015.

[65] K. L. Clarkson. Tighter bounds for random projections of manifolds. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, pages 39–48, 2008.

[66] N. T. Clemens and V. Narayanaswamy. Low-frequency unsteadiness of shock wave/turbulent boundary layer interactions. *Annual Review of Fluid Mechanics*, 46:469–492, 2014.

[67] D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by Exponential Linear Units (ELUs). *arXiv preprint arXiv:1511.07289*, 2015.

[68] R. R. Coifman and S. Lafon. Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21(1):5–30, 2006.

[69] D. Crommelin and A. Majda. Strategies for model reduction: Comparing different optimal bases. *Journal of the Atmospheric Sciences*, 61(17):2206–2217, 2004.

[70] Y.-H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.*, 10(1):177–182, 1999.

[71] A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 45–54, 2008.

[72] A. Das and D. Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

[73] A. Das and D. Kempe. Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection. *The Journal of Machine Learning Research*, 19(1):74–107, 2018.

[74] S. Das and D. Giannakis. Delay-coordinate maps and the spectra of Koopman operators. *Journal of Statistical Physics*, 175(6):1107–1145, 2019.

[75] S. Das, D. Giannakis, and J. Slawinska. Reproducing kernel Hilbert space compactification of unitary evolution groups. arXiv:1808.01515, 2018.

[76] M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[77] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010.

[78] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[79] N. K. Dhingra, M. R. Jovanović, and Z.-Q. Luo. An ADMM algorithm for optimal sensor and actuator selection. In *53rd IEEE Conference on Decision and Control*, pages 4039–4044. IEEE, 2014.

[80] M. Dihlmann, M. Drohmann, and B. Haasdonk. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. *Proc. of ADMOS*, 2011:64, 2011.

[81] D. L. Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[82] Z. Drmac and S. Gugercin. 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, 2016.

[83] Z. Drmac and A. K. Saibaba. The discrete empirical interpolation method: Canonical structure and formulation in weighted inner product spaces. *SIAM Journal on Matrix Analysis and Applications*, 39(3):1152–1180, 2018.

[84] G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory : a Convex Approach*. Springer New York, 2000.

[85] W. J. Dunstan, R. R. Bitmead, and S. M. Savaresi. Fitting nonlinear low-order models for combustion instability control. *Control Engineering Practice*, 9(12):1301–1317, 2001.

[86] K.-J. Engel and R. Nagel. *One-parameter semigroups for linear evolution equations*, volume 194. Springer Science & Business Media, 1999.

[87] M. Espinoza, A. K. Suykens, and B. D. Moor. Kernel based partially linear models and nonlinear identification. *IEEE Transactions on Automatic Control*, 50(10):1602–6, oct 2005.

[88] G. Flagg and S. Gugercin. Multipoint Volterra series interpolation and $\mathcal{H}_2$ optimal model reduction of bilinear systems. *SIAM J. Matrix Anal. Appl.*, 36(2):549–579, 2015.

[89] T. L. B. Flinois, A. S. Morgans, and P. J. Schmid. Projection-free approximate balanced truncation of large unstable systems. *Physical Review E*, 92(2):023012, 2015.

[90] C. Foias, M. S. Jolly, I. G. Kevrekidis, G. R. Sell, and E. S. Titi. On the computation of inertial manifolds. *Physics Letters A*, 131(7-8):433–436, 1988.

[91] C. Foias, O. Manley, and R. Temam. Modelling of the interaction of small and large eddies in two dimensional turbulent flows. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 22(1):93–118, 1988.

[92] C. Foias, G. R. Sell, and R. Temam. Inertial manifolds for nonlinear evolutionary equations. *J. Diff. Eq.*, 73(2):309–353, 1988.

[93] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013.

[94] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[95] Z. Ghahramani and S. T. Roweis. Learning nonlinear dynamical systems using an EM algorithm. *Advances in neural information processing systems*, pages 431–437, 1999.

[96] K. Glashoff and M. M. Bronstein. Optimization on the biorthogonal manifold. *arXiv preprint arXiv:1609.04161*, 2016.

[97] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013.

[98] F. J. Gonzalez and M. Balajewicz. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. *arXiv preprint arXiv:1808.01346*, 2018.

[99] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[100] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[101] D. Goswami and D. A. Paley. Global bilinearization and controllability of control-affine nonlinear systems: A Koopman spectral approach. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 6107–6112. IEEE, 2017.

[102] M. D. Graham, P. H. Steen, and E. S. Titi. Computational efficiency and approximate inertial manifolds for a Bénard convection system. *Journal of Nonlinear Science*, 3(1):153–167, 1993.

[103] K.-G. Grosse-Erdmann and A. P. Manguillot. *Linear chaos*. Springer Science & Business Media, 2011.

[104] J. Guckenheimer and P. Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42. Springer Science & Business Media, 2013.

[105] S. Gugercin, A. C. Antoulas, and C. Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM J. Matrix Anal. Appl.*, 30(2):609–638, 2008.

[106] V. Guillemin and A. Pollack. *Differential topology*, volume 370. American Mathematical Society, 2010.

[107] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

[108] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53:217–288, 2011.

[109] B. Hall. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer, 2015.

[110] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009.

[111] C. Hegde, A. C. Sankaranarayanan, W. Yin, and R. G. Baraniuk. Numax: A convex approach for learning near-isometric linear embeddings. *IEEE Transactions on Signal Processing*, 63(22):6109–6121, 2015.

[112] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[113] C.-M. Ho and L.-S. Huang. Subharmonics and vortex merging in mixing layers. *Journal of Fluid Mechanics*, 119:443–473, 1982.

[114] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge Univ. Press, 2012.

[115] S. Hosseinyalamdary. Deep Kalman filter: Simultaneous multi-sensor integration and modelling; a GNSS/IMU case study. *Sensors*, 18(5):1316, 2018.

[116] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

[117] W. Huang, K. A. Gallivan, and P.-A. Absil. A Broyden class of quasi-Newton methods for Riemannian optimization. *SIAM J. Optim.*, 25(3):1660–1685, 2015.

[118] M. Ilak, S. Bagheri, L. Brandt, C. W. Rowley, and D. S. Henningson. Model reduction of the nonlinear complex Ginzburg–Landau equation. *SIAM J. Appl. Dyn. Sys.*, 9(4):1284–1302, 2010.

[119] M. Ilak and C. W. Rowley. Modeling of transitional channel flow using balanced proper orthogonal decomposition. *Physics of Fluids*, 20(3):034103, 2008.

[120] S. J. Illingworth, A. S. Morgans, and C. W. Rowley. Feedback control of flow resonances using balanced reduced-order models. *J. Sound Vib.*, 330(8):1567–1581, 2011.

[121] A. A. Jamshidi and M. J. Kirby. Towards a black box algorithm for nonlinear function approximation over high-dimensional domains. *SIAM Journal on Scientific Computing*, 29(3):941–963, 2007.

[122] Y.-L. Jiang and K.-L. Xu. Riemannian modified Polak–Ribière–Polyak conjugate gradient order reduced model by tensor techniques. *SIAM J. Matrix Anal. Appl.*, 41(2):432–463, 2020.

[123] W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a Hilbert space 26. *Contemporary mathematics*, 26, 1984.

[124] M. S. Jolly, I. Kevrekidis, and E. S. Titi. Approximate inertial manifolds for the Kuramoto-Sivashinsky equation: analysis and computations. *Physica D: Nonlinear Phenomena*, 44(1-2):38–60, 1990.

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

[126] M. R. Jovanović and B. Bamieh. Componentwise energy amplification in channel flows. *J. Fluid Mech.*, 534:145–183, 2005.

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

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

[129] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960.

[130] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz. Time-delay observables for Koopman: Theory and applications. *SIAM Journal on Applied Dynamical Systems*, 19(2):886–917, 2020.

[131] K. Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*, volume 37. Sana, 1947.

[132] M. Karl, M. Soelch, J. Bayer, and P. Van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.

[133] S. Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*, volume 1, pages 413–418. IEEE, 1998.

[134] W. G. Kelly and A. C. Peterson. *The Theory of Differential Equations, Classical and Qualitative*. Pearson Prentice Hall, 2004.

[135] I. G. Kevrekidis, B. Nicolaenko, and J. C. Scovel. Back in the saddle again: a computer assisted study of the Kuramoto-Sivashinsky equation. *SIAM Journal on Applied Mathematics*, 50(3):760–790, 1990.

[136] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

[137] L. B. Koralov and Y. G. Sinai. *Theory of probability and random processes*. Springer, 2 edition, 2012.

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

[139] M. Korda and I. Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *J. Nonlin. Sci.*, 28:687–710, 2018.

[140] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec):2761–2801, 2008.

[141] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.

[142] R. Krishnan, U. Shalit, and D. Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[143] C. Kuehn. *Multiple Time Scale Dynamics*. Springer International, 2015.

[144] A. Lamraoui, F. Richecoeur, S. Ducruix, and T. Schuller. Experimental analysis of simultaneous non-harmonically related unstable modes in a swirled combustor. In *Proceedings of the ASME 2011 Turbo Expo: Turbine Technical Conference and Exposition*, volume 2, pages 1289–1299, 2011.

[145] Y. Lan and I. Mezić. Linearization in the large of nonlinear systems and Koopman operator spectrum. *Phys. D*, 242(1):42–53, 2013.

[146] A. J. Laub. *Matrix analysis for scientists and engineers*, volume 91. Siam, 2005.

[147] P. D. Lax. *Functional analysis*. John Wiley & Sons, Inc., 2002.

[148] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2007.

[149] J. M. Lee. *Introduction to Smooth Manifolds: Second Edition*. Springer New York, 2013.

[150] K. Lee and K. T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.*, 404, 2020.

[151] J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.

[152] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103111, 2017.

[153] H. W. Lin, M. Tegmark, and D. Rolnick. Why does deep and cheap learning work so well? *arXiv preprint arXiv:1608.08225*, aug 2016.

[154] D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005, 1956.

[155] A. J. Linot and M. D. Graham. Deep learning to discover and predict dynamics on an inertial manifold. *Physical Review E*, 101(6):062209, 2020.

[156] X. Liu and B. Sinopoli. On partial observability of large scale linear systems: A structured systems approach. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4655–4661. IEEE, 2018.

[157] X. Liu, S. Weerakkody, and B. Sinopoli. Sensor placement for reliable observability: a structured systems approach. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 5414–5421. IEEE, 2016.

[158] J. T.-H. Lo. Global bilinearization of systems with control appearing linearly. *SIAM Journal on Control*, 13(4):879–885, 1975.

[159] S.-C. Lo, G. A. Blaisdell, and A. S. Lyrintzis. High-order shock capturing schemes for turbulence calculations. *International Journal for Numerical Methods in Fluids*, 62(5):473–498, 2010.

[160] M. Loève. *Probability Theory II*, volume 46. Springer-Verlag, 1978.

[161] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20(2):130–141, 1963.

[162] D. M. Luchtenburg and C. W. Rowley. Model reduction using snapshot-based realizations. *Bulletin of the American Physical Society*, Volume 56, Number 18, 2011.

[163] J. L. Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation*, 1967.

[164] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018.

[165] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

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

[167] A. Mardt, L. Pasquali, H. Wu, and F. Noé. VAMPnets for deep learning of molecular kinetics. *Nature communications*, 9(1):1–11, 2018.

[168] M. Marion and R. Temam. Nonlinear Galerkin methods. *SIAM J. Numer. Anal.*, 26(5):1139–1157, 1989.

[169] R. Mathias. A chain rule for matrix functions and applications. *SIAM J. Matrix Anal. Appl.*, 17(3):610–620, 1996.

[170] H. G. Matthies and M. Meyer. Nonlinear galerkin methods for the model reduction of nonlinear dynamical systems. *Computers & Structures*, 81(12):1277–1286, 2003.

[171] A. Mauroy and I. Mezić. On the use of Fourier averages to compute the global isochrons of (quasi) periodic dynamics. *Chaos*, 22(3):033112, 2012.

[172] A. Mauroy and I. Mezić. Global stability analysis using the eigenfunctions of the Koopman operator. *IEEE Trans. Automat. Contr.*, 61(11):3356–3369, 2016.

[173] A. Mauroy, I. Mezić, and J. Moehlis. Isostables, isochrons, and Koopman spectrum for the action-angle representation of stable fixed point dynamics. *Phys. D*, 261:19–30, 2013.

[174] B. J. McKeon and A. S. Sharma. A critical-layer framework for turbulent pipe flow. *J. Fluid Mech.*, 658:336–382, 2010.

[175] N. Mehr and R. Horowitz. A submodular approach for optimal sensor placement in traffic networks. In *2018 Annual American Control Conference*, pages 6353–6358. IEEE, 2018.

[176] L. Mescheder, S. Nowozin, and A. Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*, 2017.

[177] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlin. Dyn.*, 41:309–325, 2005.

[178] I. Mezić. Koopman operator spectrum and data analysis. *arXiv preprint arXiv:1702.07597*, 2017.

[179] P. W. Michor. *Topics in differential geometry*, volume 93. American Mathematical Society, 2008.

[180] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978.

[181] V. Mons, J.-C. Chassaing, and P. Sagaut. Optimal sensor placement for variational data assimilation of unsteady flows past a rotationally oscillating cylinder. *Journal of Fluid Mechanics*, 823:230–277, 2017.

[182] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Contr.*, 26(1):17–32, 1981.

[183] N. J. Nair and A. Goza. Integrating sensor data into reduced-order models with deep learning. *Bulletin of the American Physical Society*, 64, 2019.

[184] N. J. Nair and A. Goza. Leveraging reduced-order models for state estimation using deep learning. *arXiv preprint arXiv:1912.10553*, 2019.

[185] I. Najfeld and T. F. Havel. Derivatives of the matrix exponential and their computation. *Advances in applied mathematics*, 16(3):321–375, 1995.

[186] A. Narasingam and J. S.-I. Kwon. Koopman Lyapunov-based model predictive control of nonlinear chemical process systems. *AIChE Journal*, 65(11):e16743, 2019.

[187] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[188] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

[189] L. Noakes. The takens embedding theorem. *International Journal of Bifurcation and Chaos*, 1(04):867–872, 1991.

[190] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[191] M. Ohlberger and S. Rave. Reduced basis methods: Success, limitations and future challenges. In *Proceedings of Algoritmy*, pages 1–12, 2016.

[192] S. E. Otto, A. Padovan, and C. W. Rowley. Optimizing oblique projections for nonlinear systems using trajectories. *arXiv preprint arXiv:2106.01211*, 2021.

[193] S. E. Otto and C. W. Rowley. A discrete empirical interpolation method for interpretable immersion and embedding of nonlinear manifolds. *arXiv preprint arXiv:1905.07619*, 2019.

[194] S. E. Otto and C. W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

[195] S. E. Otto and C. W. Rowley. Inadequacy of linear methods for minimal sensor placement and feature selection in nonlinear systems; a new approach using secants. *arXiv preprint arXiv:2101.11162*, 2021.

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

[197] A. Padovan, S. E. Otto, and C. W. Rowley. Analysis of amplification mechanisms and cross-frequency interactions in nonlinear flows via the harmonic resolvent. *Journal of Fluid Mechanics*, 900, 2020.

[198] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

[199] A. Pazy. *Semigroups of linear operators and applications to partial differential equations*, volume 44. Springer, 1983.

[200] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1):A168–A192, 2014.

[201] S. Peitz and S. Klus. Koopman operator-based model reduction for switched-system control of PDEs. *Automatica*, 106:184–191, 2019.

[202] S. Peitz, S. E. Otto, and C. W. Rowley. Data-driven model predictive control using interpolated Koopman generators. *SIAM J. Appl. Dyn. Sys.*, 19(3):2162–2193, 2020.

[203] L. Peng and K. Mohseni. Nonlinear model reduction via a locally weighted pod method. *International Journal for Numerical Methods in Engineering*, 106(5):372–396, 2016.

[204] W. Peng and Y. Ai-Jun. Improved pruning algorithm using quadratic renyi entropy for ls-svm modeling. In *24th Chinese Control and Decision Conference (CCDC)*, pages 3471–3474. IEEE, 2012.

[205] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 1999.

[206] J. B. Perot. An analysis of the fractional step method. *J. Comput. Phys.*, 108(1):51–58, 1993.

[207] H. Poincaré. Mémoire sur les courbes définies par une équation différentielle (i). *Journal de Mathématiques Pures et Appliquées*, 7:375–422, 1881.

[208] H. Poincaré. Mémoire sur les courbes définies par une équation différentielle (ii). *Journal de Mathématiques Pures et Appliquées*, 8:251–296, 1882.

[209] S. Priebe and M. P. Martín. Low-frequency unsteadiness in shock wave–turbulent boundary layer interaction. *Journal of Fluid Mechanics*, 699:1–49, 2012.

[210] S. Priebe, J. H. Tu, C. W. Rowley, and M. P. Martín. Low-frequency dynamics in a shock-induced separated flow. *Journal of Fluid Mechanics*, 807:441–477, 2016.

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

[212] F. Pukelsheim. *Optimal Design of Experiments*. SIAM, 2006.

[213] T. Quail, A. Shrier, and L. Glass. Predicting the onset of period-doubling bifurcations in noisy cardiac systems. *Proceedings of the National Academy of Sciences*, 112(30):9358–9363, 2015.

[214] J. F. Queiró. On the interlacing property for singular values and eigenvalues. *Linear Algebra and Its Applications*, 97:23–28, 1987.

[215] S. Rao, S. P. Chepuri, and G. Leus. Greedy sensor selection for non-linear models. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 241–244. IEEE, 2015.

[216] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[217] H. Rauch. Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 8(4):371–372, 1963.

[218] M. Reed and B. Simon. *Methods of modern mathematical physics, Volume I: Functional Analysis*. Academic press, 1980.

[219] G. Rega and H. Troger. Dimension reduction of dynamical systems: methods, models, applications. *Nonlin. Dyn.*, 41(1-3):1–15, 2005.

[220] R. Rico-Martines, I. Kevrekidis, M. Kube, and J. Hudson. Discrete-vs. continuous-time nonlinear signal processing: Attractors, transitions and parallel implementation issues. In *1993 American Control Conference*, pages 1475–1479. IEEE, 1993.

[221] R. Rico-Martinez and I. G. Kevrekidis. Continuous time modeling of nonlinear systems: A neural network-based approach. In *IEEE International Conference on Neural Networks*, pages 1522–1525. IEEE, 1993.

[222] W. Ring and B. Wirth. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM J. Optim.*, 22(2):596–627, 2012.

[223] D. Rolnick and M. Tegmark. The power of deeper networks for expressing natural functions. *arXiv preprint arXiv:1705.05502*, 2017.

[224] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[225] C. W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *Int. J. Bifurcation Chaos*, 15(03):997–1013, 2005.

[226] C. W. Rowley. Data-driven methods for identifying nonlinear models of fluid flows. In *KITP Program: Recurrent Flows: The Clockwork Behind Turbulence*. Kavli Institute for Theoretical Physics, University of California, Santa Barbara, 2017.

[227] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1-2):115–129, 2004.

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

[229] C. W. Rowley, I. G. Kevrekidis, J. E. Marsden, and K. Lust. Reduction and reconstruction for self-similar dynamical systems. *Nonlinearity*, 16(4):1257, 2003.

[230] C. W. Rowley and J. E. Marsden. Reconstruction equations and the karhunen–loève expansion for systems with symmetry. *Physica D: Nonlinear Phenomena*, 142(1-2):1–19, 2000.

[231] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *J. Fluid Mech.*, 641:115–127, Dec. 2009.

[232] R. Rubinstein, A. M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[233] W. Rudin. *Real and Complex Analysis: Third Edition*. McGraw-Hill, 1987.

[234] R. D. Russell, D. M. Sloan, and M. R. Trummer. Some numerical aspects of computing inertial manifolds. *SIAM Journal on Scientific Computing*, 14(1):19–43, 1993.

[235] H. Sato. A Dai–Yuan-type Riemannian conjugate gradient method with the weak Wolfe conditions. *Comput. Optim. and Appl.*, 64(1):101–118, 2016.

[236] H. Sato and T. Iwai. A new, globally convergent Riemannian conjugate gradient method. *Optimization*, 64(4):1011–1031, 2015.

[237] H. Sato, H. Kasai, and B. Mishra. Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport. *SIAM J. Optim.*, 29(2):1444–1472, 2019.

[238] H. Sato and K. Sato. Riemannian trust-region methods for $\mathcal{H}_2$ optimal model reduction. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4648–4655. IEEE, 2015.

[239] H. Schaeffer, G. Tran, and R. Ward. Extracting sparse high-dimensional dynamics from limited data. *SIAM Journal on Applied Mathematics*, 78(6):3279–3295, 2018.

[240] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.*, 656:5–28, 2010.

[241] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.*, 656:5–28, 2010.

[242] P. J. Schmid and D. S. Henningson. *Stability and Transition in Shear Flows*, volume 142. Springer-Verlag New York, 2001.

[243] P. J. Schmid and J. L. Sesterhenn. Dynamic mode decomposition of numerical and experimental data. In *Sixty-First Annual Meeting of the APS Division of Fluid Dynamics*, San Antonio, Texas, USA, 2008.

[244] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[245] P. Sebastiani and H. P. Wynn. Maximum entropy sampling and optimal Bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157, 2000.

[246] L. Shaabani-Ardali, D. Sipp, and L. Lesshafft. Vortex pairing in jets as a global Floquet instability: modal and transient dynamics. *J. Fluid Mech.*, 862:951–989, 2019.

[247] M. Shamaiah, S. Banerjee, and H. Vikalo. Greedy sensor selection: Leveraging submodularity. In *49th IEEE Conference on Decision and Control*, pages 2572–2577. IEEE, 2010.

[248] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of applied statistics*, 14(2):165–170, 1987.

[249] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of time series analysis*, 3(4):253–264, 1982.

[250] L. Sirovich. Turbulence and the dynamics of coherent structures. i. coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[251] K. Skretting and K. Engan. Recursive least squares dictionary learning algorithm. *IEEE Transactions on signal processing*, 58(4):2121–2130, 2010.

[252] T. H. Summers, F. L. Cortesi, and J. Lygeros. On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, 3(1):91–101, 2015.

[253] T. H. Summers, F. L. Cortesi, and J. Lygeros. On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, 3(1):91–101, March 2016.

[254] W. Sun, G. Yang, B. Du, L. Zhang, and L. Zhang. A sparse and low-rank near-isometric linear embedding method for feature extraction in hyperspectral imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):4032–4046, 2017.

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

[256] S. Svoronos, G. Stephanopoulos, and R. Aris. Bilinear approximation of general non-linear dynamic systems with linear inputs. *International Journal of Control*, 31(1):109–126, 1980.

[257] N. Takeishi, Y. Kawahara, and T. Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pages 1130–1140, 2017.

[258] F. Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[259] S. Tao, D. Chen, and W. Zhao. Fast pruning for multi-output ls-svm and its application to chemical pattern classification. *Chemometrics and Intelligent Laboratory Systems*, 96:63–69, mar 2009.

[260] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[261] R. C. Thompson. Principal submatrices ix: Interlacing inequalities for singular values of submatrices. *Linear Algebra and its Applications*, 5(1):1–12, 1972.

[262] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[263] E. S. Titi. On approximate inertial manifolds to the Navier-Stokes equations. *Journal of mathematical analysis and applications*, 149(2):540–557, 1990.

[264] L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll. Hydrodynamic stability without eigenvalues. *Science*, 261:578–584, July 1993.

[265] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Simultaneous sparse approximation via greedy pursuit. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–721. IEEE, 2005.

[266] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *J. Comput. Dyn.*, 1(2):391–421, 2014.

[267] V. Tzoumas, A. Jadbabaie, and G. J. Pappas. Sensor placement for optimal Kalman filtering: Fundamental limits, submodularity, and algorithms. In *2016 American Control Conference*, pages 191–196. IEEE, 2016.

[268] O. Tzuk, S. R. Ujjwal, C. Fernandez-Oto, M. Seifan, and E. Meron. Period doubling as an indicator for ecosystem sensitivity to climate extremes. *Scientific reports*, 9(1):1–10, 2019.

[269] C. F. Van Loan. The ubiquitous kronecker product. *Journal of computational and applied mathematics*, 123(1-2):85–100, 2000.

[270] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[271] P. R. Vlachas, G. Arampatzis, C. Uhler, and P. Koumoutsakos. Learning the effective dynamics of complex multiscale systems. *arXiv preprint arXiv:2006.13431*, 2020.

[272] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213):20170844, 2018.

[273] M. J. Wainwright. *High-dimensional statistics*. Cambridge university press, 2019.

[274] W.-G. Wang and Y.-L. Jiang. $\mathcal{H}_2$ optimal model order reduction on the Stiefel manifold for the MIMO discrete system by the cross Gramian. *Math. and Comp. Modeling of Dyn. Sys.*, 24(6):610–625, 2018.

[275] Z. Wang, D. Xiao, F. Fang, R. Govindan, C. C. Pain, and Y. Guo. Model identification of reduced order fluid dynamics systems using deep learning. *International Journal for Numerical Methods in Fluids*, 86(4):255–268, 2018.

[276] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti. Visual interaction networks: Learning a physics simulator from video. In *Advances in Neural Information Processing Systems*, pages 4542–4550, 2017.

[277] H. Whitney. Differentiable manifolds. *Annals of Mathematics*, pages 645–680, 1936.

[278] H. Whitney. The self-intersections of a smooth n-manifold in $2n$-space. *Annals of Mathematics*, pages 220–246, 1944.

[279] M. O. Williams, M. S. Hemati, S. T. Dawson, I. G. Kevrekidis, and C. W. Rowley. Extending data-driven Koopman analysis to actuated systems. *IFAC-PapersOnLine*, 49(18):704–709, 2016. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016.

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

[281] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel-based method for data-driven Koopman spectral analysis. *J. Comput. Dyn.*, 2(2):247–265, 2015.

[282] P. Wolfe. Convergence conditions for ascent methods. *SIAM Rev.*, 11(2):226–235, 1969.

[283] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

[284] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2008.

[285] C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95 – 103, 1983.

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

[287] K.-L. Xu and Y.-L. Jiang. An unconstrained h_2 model order reduction optimisation algorithm based on the Stiefel manifold for bilinear systems. *Int. J. Control*, 92(4):950–959, 2019.

[288] Y. Xu and T. Zeng. Fast optimal $\mathcal{H}_2$ model reduction algorithms based on Grassmann manifold optimization. *International Journal of Numerical Analysis and Modeling*, 10(4):972–991, 2013.

[289] Y.-L. Xu and D.-R. Chen. Partially-linear least-squares regularized regression for system identification. *IEEE Transactions on Automatic Control*, 54(11):2637–41, nov 2009.

[290] W.-Y. Yan and J. Lam. An approximate approach to $\mathcal{H}_2$ optimal model reduction. *IEEE Trans. Automat. Contr.*, 44(7):1341–1358, 1999.

[291] P. Yang, Y.-L. Jiang, and K.-L. Xu. A trust-region method for h_2 model reduction of bilinear systems on the Stiefel manifold. *J. Franklin Inst.*, 356(4):2258–2273, 2019.

[292] H. C. Yee, N. D. Sandham, and M. J. Djomehri. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *Journal of Computational Physics*, 150(1):199–238, 1999.

[293] E. Yeung, S. Kundu, and N. Hodas. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. *arXiv preprint arXiv:1708.06850*, aug 2017.

[294] E. Yeung, S. Kundu, and N. Hodas. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. In *2019 American Control Conference (ACC)*, pages 4832–4839. IEEE, 2019.

[295] B. Yildirim, C. Chryssostomidis, and G. Karniadakis. Efficient sensor placement for ocean measurements using low-dimensional concepts. *Ocean Modelling*, 27(3-4):160–173, 2009.

[296] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[297] A. Zare, T. T. Georgiou, and M. R. Jovanović. Stochastic dynamical modeling of turbulent flows. *Ann. Rev. Contr. Robot. Aut. Sys.*, 3:195–219, 2020.

[298] T. Zeng and C. Lu. Two-sided Grassmann manifold algorithm for optimal model reduction. *Int. J. Numer. Meth. Eng.*, 104(10):928–943, 2015.

[299] H. Zhang, R. Ayoub, and S. Sundaram. Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms. *Automatica*, 78:202–210, 2017.

[300] K. Zhou, G. Salomon, and E. Wu. Balanced realization and model reduction for unstable systems. *International Journal of Robust and Nonlinear Control*, 9(3):183–198, mar 1999.