KERNEL METHODS FOR SYSTEM IDENTIFICATION AND FAULT DETECTION IN NONLINEAR SYSTEMS

By

ZACHARY DANIEL MORRISON

Bachelor of Science in Aerospace Engineering
Oklahoma State University
Stillwater, Oklahoma
2020

Bachelor of Science in Mathematics
Oklahoma State University
Stillwater, Oklahoma
2023

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2024

KERNEL METHODS FOR SYSTEM IDENTIFICATION AND FAULT DETECTION IN NONLINEAR SYSTEMS

Thesis Approved:

Dr. Rushikesh Kamalapurkar

Thesis Advisor

Dr. Lisa Mantini

Dr. He Bai

ACKNOWLEDGMENTS

I would like to express my gratitude towards Dr. Kamalapurkar for your outstanding guidance, patience, expertise, and deep understanding of the literature behind Koopman operator theory. You helped push me to gain a better understanding of functional analysis, and you taught me how to think as a scientist. For that I am grateful.

Thank you, Dr. Mantini, Dr. Lebl, Dr. Wright, and Dr. Mills of the Department of Mathematics. I would not be where I am today without the mathematics foundation provided to me here at Oklahoma State.

Thank you, Dr. Bai, for your comments and support.

Thank you to my labmates - especially Moad - I'm grateful for all the productive conversations regarding operator theory.

Lastly, I would like to thank Kylee for her unwavering love and support.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: ZACHARY DANIEL MORRISON

Date of Degree: MAY, 2024

Title of Study: KERNEL METHODS FOR SYSTEM IDENTIFICATION AND FAULT DETECTION IN NONLINEAR SYSTEMS

Major Field: MECHANICAL AND AEROSPACE ENGINEERING

Abstract: In recent years, data-driven methods for the analysis of nonlinear systems have flourished. Derived from machine learning techniques, these methods allow one to analyze, predict, and control the behavior of a nonlinear system without prior model knowledge. The only requirement are data taken from the nonlinear system of interest; moreover, data-driven models are particularly useful for complex nonlinear systems and have seen success in many branches of engineering, from modal decomposition of fluid flows to designing stabilizing controllers for nonlinear systems. In particular, this thesis will focus on the extension of two of these data-driven methods: dynamic mode decomposition (DMD) and kernelized principal component analysis (KPCA).

DMD, which relies on representing a nonlinear system as an infinite-dimensional linear operator, has seen success in predicting the behavior of both continuous-time and discrete-time nonlinear systems without prior model knowledge; however, the extension of DMD methods to discrete-time, controlled nonlinear systems is nontrivial. In this thesis, we develop a novel operator representation of discrete-time, control-affine nonlinear dynamical systems. The representation is learned using recorded snapshots of the system state resulting from arbitrary, potentially open-loop control inputs. We thereby extend the predictive capabilities of dynamic mode decomposition to discrete-time nonlinear systems that are affine in control.

KPCA is typically a data-driven dimensionality reduction technique that allows one to study a nonlinear system via a reduced-order model in a higher-dimensional space; however, KPCA can be used for fault detection in nonlinear systems without prior model knowledge. Reliable operation of automatic systems is heavily dependent on the ability to detect faults in the underlying dynamics. While traditional model-based methods have been widely used for fault detection, data-driven approaches have garnered increasing attention due to their ease of deployment and minimal need for expert knowledge. In the latter portion of this thesis, we develop a novel fault detection method using KPCA with the occupation kernel as the feature map. Occupation kernels result in feature maps that are tailored to the measured data, have inherent noise-robustness due to the use of integration, and can utilize irregularly sampled system trajectories of variable lengths for PCA.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation

In recent years, machine learning has flourished in applications to dynamical systems: applications ranging from dynamic mode decomposition (DMD) (e.g. [6, 12, 27, 30]), DMD-based feedback law design [35, 36], and fault detection [7, 19]. In this thesis, we will derive two new applications: a DMD method for discrete-time control-affine dynamical systems and a fault detection method which extends kernelized principal component analysis (KPCA) by using a feature map that encodes entire trajectories into a Hilbert space.

In the second chapter of this thesis, a novel representation of discrete-time control-affine nonlinear systems as infinite-dimensional linear operators over reproducing kernel Hilbert spaces (RKHSs) is introduced. This effort is inspired by the method first developed in [30], which introduced similar operator representations for *continuous-time* dynamical systems. The idea of representing a nonlinear system as an infinite-dimensional linear operator in Hilbert space was first put forth by B.O. Koopman in [10] and the resulting composition operator is aptly known as the Koopman operator. This higher-dimensional space is typically referred to as the *feature space* or *lifted space* and the Koopman operator acts as a composition operator on the lifted space. In recent years, dynamic mode decomposition (DMD) and other data-driven methods have seen a resurgence due to the abundance of data and increased the availability of computational power [13]. An example of the application of DMD can be seen in the fluid mechanics community, where modal decomposition of fluid flows is accomplished [32], [16]. In a more general sense, DMD is intimately connected to the Koopman operator

as DMD is one method used to approximate the Koopman operator associated with the dynamical system [16], [37].

The Koopman approach is amenable to spectral methods in linear operator theory in certain cases, e.g. see [5], but spectral convergence cannot be guaranteed in general; therefore, Koopman DMD methods are pseudo-spectral numerical methods. Despite this theoretical limitation, Koopman DMD and Liouville DMD methods in both continuous and discrete time have been shown to exhibit remarkable predictive accuracy over finite-time horizons [13]. Moreover, Koopman DMD allows one to study dynamical systems without direct knowledge of the dynamics, as Koopman DMD is strictly data driven and requires no knowledge of the dynamical system [13]. For measurements corrupted by noise or in the case of stochastic systems, robust approximations of the Koopman operator can be formulated [34]. The ultimate goal of DMD is to develop a data-driven model via an eigendecomposition of the Koopman operator, under the assumption that the full-state observable (the identity function) is in the span of the eigenfunctions [5]. The addition of control adds greater difficulty to data-driven methods like DMD, as the Koopman operator associated with the dynamical system depends upon the control input. Furthermore, in discrete time, the Koopman operator is generally not linear in its symbol, which makes separating the influence of the controller from the drift dynamics challenging. Despite the difficulty, there have been several successful methods for generalizing Koopman DMD for dynamical systems with control in results such as [6], [8, 12, 27, 28, 40].

In the third chapter of this thesis, we extend KPCA to make use of trajectories generated by a dynamical system as a fundamental unit of data. Principal component analysis (PCA) diagonalizes the covariance matrix associated with the fault-free training data (i.e. training data taken from the normal operating regime of a dynamical system). The idea of using the covariance matrix in PCA is that we can obtain *orthogonal*, uncorrelated variables (i.e. the eigenvectors of the covariance matrix) that allow us to describe the directions of maximum variance in our dataset. The dominant eigenvectors of the covariance matrix can be used

to effectively reduce the dimension of a dataset because typically the number of principal components required to describe the variance in the dataset is far less than the dimension of the dataset. A key limitation of PCA is that it fails to capture nonlinear relationships present in the data. Kernelized PCA remedies this limitation by lifting the data to a higher-dimensional feature space via a (nonlinear) feature map [33]. PCA is then applied in the feature space, resulting in nonlinear principal components [33]. Fault detection using KPCA / PCA relies on computation of a metric (T^2, SPE, etc.) that measures how well new data can be reconstructed using the principal components [3, 7, 15, 18, 19, 21, 38, 39].

The feature maps used for KPCA are typically the canonical feature maps associated with generic kernel functions such as the Gaussian radial basis function [7]. As such, the feature maps are largely independent of the system or the measured data. In the third chapter, a new PCA framework is developed where the feature maps are also derived from the training data. The idea, motivated by results such as [31], is to use trajectories generated by a dynamical system as a fundamental unit of data by embedding them in a reproducing kernel Hilbert space (RKHS) using the so-called occupation kernels. The resulting PCA method, called occupation kernel PCA (OKPCA), is expected to perform better owing to the use of feature maps that are adapted to the data. In addition the computations required to implement OKPCA rely exclusively on integrals of kernel functions evaluated along system trajectories. As a result, OKPCA is endowed with intrinsic robustness to zero-mean noise and can be implemented on data sets containing variable length trajectories that are irregularly sampled [31]. Fault detection then proceeds by reconstructing a given trajectory as a linear combination of eigenfunctions of a suitably defined kernelized covariance operator and computing a suitable analog of the reconstruction error used for KPCA by Hoffman [7].

1.2 Literature Review

Current state-of-the-art dynamic mode decomposition techniques for discrete-time dynamical systems with control can be summarized in the following papers. The method presented

in [27] yields a DMD routine to represent a general nonlinear system with control as a control-affine linear system. This idea is generalized in [12] with extended DMD (eDMD), providing greater predictive power. Furthermore, for a general discrete-time, nonlinear dynamical system with control, the authors in [12] utilize the shift operator to describe the time evolution of the control signal. Also, in discrete-time, separation of the control input from the state can be achieved via first order approximations [35]. For continuous-time dynamical systems, the Koopman canonical transform (see [36]) is used in [6] to leverage a formulation of the dynamical system in the lifted space as a control-affine, bilinear system, called the Koopman bilinear form (KBF). The KBF is then amenable to the design of feedback laws using techniques from optimal control.

The aforementioned methods demonstrate the ability to predict the response of both discrete-time [12] and continuous-time [6] dynamical systems to open-loop inputs. The algorithm developed in this thesis offers an advantage over the methods from [6, 8, 12, 27, 28, 35, 40], since in addition to a predictive model, it also estimates eigenfunctions, and consequently, a Koopman invariant subspace of the *closed-loop system*. A key contribution here is the extension of the method presented in [30] to the discrete-time case. The operator representation presented in [30] relies on linearity of differential and multiplication operators to separate the influence of the controlled and the uncontrolled part of the system dynamics. In discrete time, the differential operators need to be replaced by composition operators, and composition operators are typically not linear in their symbol. Herein lies the difficulty of extending continuous-time DMD results to discrete-time DMD results, as separation of the effect of the control input from the effect of the drift dynamics is nontrivial.

Fault detection methods for dynamical systems rely on the identification of anomalous behavior using measured data. Applications of fault detection range from healthcare [39]; manufacturing [29,38]; monitoring sensor behavior [18,21]; monitoring chemical processes [3,15]; identifying the onset of nonlinear behavior in dynamical systems [19]; and identifying traffic anomalies [26]. A multitude of approaches to fault detection have been studied over the past

few decades, such as data-driven, set-based, observer-based, and time-series analysis methods [17, 20, 22, 29]. Set-based methods accomplish fault detection by computing a forward reachable set and checking if the system state at the next time step is inside that set [17]. State estimation techniques such as the extended Kalman filter (EKF) and the Leunberger observer have been successfully implemented for fault detection in industrial processes [4, 22]. Set-based and observer-based methods are model based, whereas this thesis focuses on data-driven fault detection [29].

Data-driven fault detection methods, such as principal component analysis (PCA), kernelized principal component analysis (KPCA), and the Kahrunen-Loeve transform (KLT) [9], typically employ multivariate statistical procedures combined with an index, such as a reconstruction error, Hotteling's T^2, a squared prediction error (SPE), or a combination thereof, to detect anomalies [7, 17, 29]. The KLT utilizes the expansion of a random variable as a linear combination of eigenfunctions of the covariance operator for fault detection [20]. In finite dimensions and in the context of data driven methods or discrete sampling, the KLT is simply PCA.

1.3 Contributions

The contributions of this thesis are as follows:

1. The development of a pseudo-spectral method (dubbed DCLDMD) for the analysis of discrete-time dynamical systems which are affine in control. Given trajectory data that encodes the dynamical system's response to a series of (potentially arbitrary) open-loop control inputs, the DCLDMD algorithm allows one to accurately estimate the response of a dynamical system to a known feedback controller.

2. It is demonstrated that, heuristically, DCLDMD can estimate invariant subspaces of a discrete-time, control-affine dynamical system under the effect of a known feedback law - to the best of the author's knowledge, no current state-of-the-art techniques

in the literature can do this without prior knowledge of invariant subspaces of the uncontrolled dynamical system.

3. Furthermore, the predictive capabilities of DCLDMD outperform current techniques in the literature for discrete-time, control-affine dynamical systems as demonstrated in the experiment section of Chapter 2. This is due to the fact that DCLDMD develops a data-driven model in terms of eigenfunctions of the DCLDMD operator - current DMD methods for discrete-time dynamical systems do not perform a spectral decomposition of the composition operator associated to the dynamical system, and, as such, cannot develop a *nonlinear* data-driven model.

4. A fault detection technique is developed that provides a novel extension of the kernelized PCA method to allow for the use of trajectories as a fundamental unit of data. The resulting method - known as occupation kernel PCA - has greater flexibility with training data sets, while still maintaining the fault detection properties of KPCA.

CHAPTER II

KERNEL METHODS FOR SYSTEM IDENTIFICATION IN NONLINEAR SYSTEMS

In this chapter, we take an operator-theoretic approach to DMD with a novel operator definition that accounts for the effect of control and the discrete-time nature of the problem. The algorithm is referred to as discrete control Liouville DMD or DCLDMD for brevity. To accomplish DCLDMD, the discrete, nonlinear dynamical system is represented as a composition of two operators acting on a Hilbert space of functions. The first operator mimics the effects of a composition operator, which maps from a reproducing kernel Hilbert space (RKHS) to a vector-valued RKHS (vvRKHS). In order to account for the effect of control, we make use of a multiplication operator which maps functions in the vvRKHS back into the RKHS. In doing so, we obtain an approximate representation of the dynamical system as a composition of the aforementioned operators.

2.1 Background

In this section, we provide a brief overview of RKHSs and vvRKHSs and their role in DCLDMD.

Definition 2.1.1 *An RKHS $\tilde{H}$ over a set $X \subset \mathbb{R}^n$ is a Hilbert space of functions $f : X \to \mathbb{C}$ such that for all $x \in X$ the evaluation functional $E_x f \coloneqq f(x)$ is bounded. By the Riesz representation theorem, there exists a function $\tilde{K}_x \in \tilde{H}$ such that $f(x) = \langle f, \tilde{K}_x \rangle_{\tilde{H}}$ for all $f \in \tilde{H}$.*

The snapshots of a dynamical system are embedded into an RKHS via a kernel map $x \mapsto \tilde{K}(\cdot, x) \coloneqq \tilde{K}_x$. Moreover, the span of the set $\{\tilde{K}_x : x \in X\}$ is dense in $\tilde{H}$.

Proposition 2.1.1 *If $A \coloneqq \{\tilde{K}_x : x \in X\}$, then span $A = \tilde{H}$.*

Proof. To show that the span of the set $\{\tilde{K}_x : x \in X\}$ is dense in $\tilde{H}$ amounts to showing that $(A^\perp)^\perp = \tilde{H}$. Let $h \in A^\perp$, then $\langle h, \tilde{K}_x \rangle = h(x) = 0$. Hence $h \equiv 0$ on X. Thus $A^\perp = \{0\}$. ■

In order to account for the effect of control, we make use of a vvRKHS.

Definition 2.1.2 *Let $\mathcal{Y}$ be a Hilbert space, and let H be a Hilbert space of functions from a set X to $\mathcal{Y}$. The Hilbert space H is a* vvRKHS *if for every $\bar{u} \in \mathcal{Y}$ and $x \in X$, the functional $f \mapsto \langle f(x), \bar{u} \rangle_{\mathcal{Y}}$ is bounded.*

To each $x \in X$ and $\bar{u} \in \mathcal{Y}$, we can associate a linear operator over a vvRKHS given by $(x, \bar{u}) \mapsto K_{x,\bar{u}}$, following [30]. The function $K_{x,\bar{u}}$ is known as the kernel operator and the span of these functions constitutes a dense set in the respective vvRKHS [30, Proposition 1]. Given a function $f \in H$, the reproducing property of $K_{x,\bar{u}}$ implies $\langle f, K_{x,\bar{u}} \rangle_H = \langle f(x), \bar{u} \rangle_{\mathcal{Y}}$. For more discussion on vvRKHSs see [2].

2.2 Problem Statement

Consider a control-affine, discrete-time dynamical system of the form

$$x_{k+1} = F(x_k) + G(x_k)u_k, \tag{2.2.1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, $F : \mathbb{R}^n \to \mathbb{R}^n$ and $G : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are functions corresponding to the drift dynamics and the control effectiveness, respectively. We refer to the individual functions which comprise the columns of G by $G_j : \mathbb{R}^n \to \mathbb{R}^n$, for $1 \leq j \leq m$. Given a feedback law $\mu : \mathbb{R}^n \to \mathbb{R}^m$ and a set of data points $\{(x_k, x_{k+1}, u_k)\}_{k=1}^n$, where u_k are arbitrary (potentially open-loop) control inputs, the goal is to predict the response of the system in (2.2.1) to the feedback law μ.

For our purposes, the set X is selected to be a compact subset of $\mathbb{R}^n$, the set $\mathcal{Y}$ is selected to be $\mathbb{C}^{1\times(m+1)}$, $\tilde{H}$ denotes an RKHS of continuous functions from X to $\mathbb{C}$, and H denotes a vvRKHS of continuous functions from X to $\mathbb{C}^{1\times(m+1)}$.

2.3 Operator representation of controlled discrete-time systems

A linear operator can be associated with the dynamical system in (2.2.1), as a composition of two operators: a composition-like discrete Liouville operator and a multiplication operator. This operator representation is derived in this section.

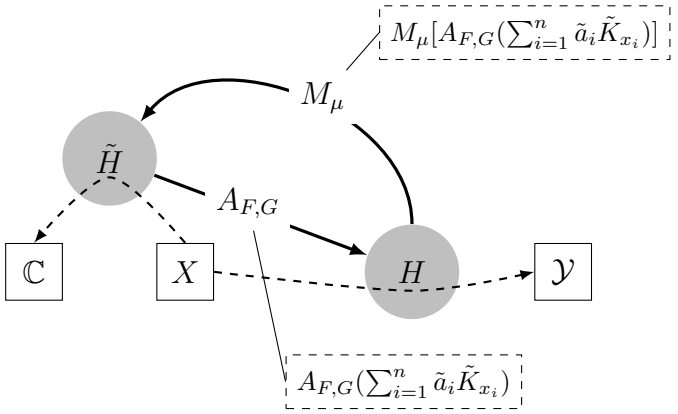

Figure 1: A schematic diagram of the construction presented in Section 2.3. The RKHSs are represented by filled circles. The squares at the endpoints of the dashed arrows passing through the circles indicate the domains and co-domains of the functions contained in the RKHSs. The thick arrows between the RKHSs indicate operators. Lastly, $\sum_{i=1}^n \tilde{a}_i \tilde{K}_{x_i}$ represents the projection of an observable onto the span $\alpha \subset \tilde{H}$ (see Algorithm 1 and Section 2.4 for more information).

2.3.1 A Composition-like Kernel Propagation Operator

The technical lemma below and the proposition that follows are needed for the kernel propagation operator to be well-defined.

Lemma 2.3.1 *The set* $\Omega \subset H$*, defined as* $\Omega \coloneqq \{K_{x,\bar{u}} : x \in X, u \in \mathbb{R}^m,$ *and* $\bar{u} \coloneqq \begin{pmatrix} 1 & u^\top \end{pmatrix} \in \mathcal{Y}\}$*, satisfies* $\Omega^\perp = \{0\}$.

Proof. Let $h \in \Omega^\perp$. The reproducing property of $K_{x,\bar{u}}$ implies that for all $u \in \mathbb{R}^m$ and $x \in X$, $\left\langle h(x), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = \langle h, K_{x,\bar{u}} \rangle_H$. Since $h \in \Omega^\perp$ and $K_{x,\bar{u}} \in \Omega$, we conclude

that $\langle h, K_{x,\bar{u}} \rangle_H = 0$. As a result, for each fixed $x \in X$ and for all $u \in \mathbb{R}^m$, we have $\left\langle h(x), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = 0$. That is, $\left\langle \Re\left(h(x)\right), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} + \mathrm{i} \left\langle \Im\left(h(x)\right), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = 0$, where $\Re\left(h(x)\right)$ denotes the real part of $h(x)$ and $\Im\left(h(x)\right)$ denotes the imaginary part of $h(x)$, respectively. Since the only such $h(x) \in \mathcal{Y}$ is the zero vector, we conclude that $h = 0$. ■

Proposition 2.3.1 *Let $L_z \in H$ be a function such that for all tuples (x, u, y) satisfying $y = F(x) + G(x)u$, we have*

$$\left\langle [L_z](x), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = \tilde{K}_z(y). \tag{2.3.1}$$

For all $z \in X$, the map $\tilde{K}_z \mapsto L_z$ is a well-defined operator.

Proof. For a given $z \in X$, suppose there are two functions, L_z^1 and L_z^2, each of which satisfy (2.3.1) given above. Then, for any tuple (x, y, u) which satisfies $y = F(x) + G(x)u$,

$$\left\langle [L_z^1](x), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = \left\langle [L_z^2](x), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}},$$

and therefore, $\left\langle [L_z^1](x) - [L_z^2](x), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = 0$.

Using the reproducing property,

$$\left\langle [L_z^1] - [L_z^2], K_{x,\bar{u}} \right\rangle_H = 0 \tag{2.3.2}$$

for all vectors in the set $\Omega \coloneqq \{K_{x,\bar{u}} : x \in X \text{ and } \bar{u} \in \mathcal{Y} \mid \bar{u} = \begin{pmatrix} 1 & u^\top \end{pmatrix}, u \in \mathbb{R}^m\}$.

As a result, $[L_z^1] - [L_z^2] \in \Omega^\perp$, where $\perp$ denotes the orthogonal complement of $\Omega \subset \mathcal{Y}$.

Since $\Omega^\perp = \{0\}$ according to lemma 2.3.1, we conclude that for all $z \in X$, $[L_z^1] = [L_z^2]$. That is, the operator $\tilde{K}_z \mapsto L_z$ is well defined on the set $\{\tilde{K}_z\}_{z \in X}$. ■

The operator can be canonically extended to $\operatorname{span}\{\tilde{K}_z\}_{z \in X}$ by insisting that for finite linear combinations of kernels we have

$$\sum_{i=1}^{N} a_i \left\langle [L_{z_i}](x), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = \sum_{i=1}^{N} a_i \tilde{K}_{z_i}(y). \tag{2.3.3}$$

Linearity of the operator then implies that it is also well-defined on $D \coloneqq \{h + \mathrm{i}g \in \tilde{H} \mid h, g \in \operatorname{span}\{\tilde{K}_z\}_{z \in X}\}$.

Definition 2.3.1 *Let $A_{F,G} : \mathcal{D}(A_{F,G}) \to H$ be the linear operator with domain $\mathcal{D}(A_{F,G}) \coloneqq D$ that maps, for each $z \in X$, the function $\tilde{K}_z$ to a function $[A_{F,G}\tilde{K}_z] \in H$ such that for all tuples (x, u, y) satisfying $y = F(x) + G(x)u$, we have*

$$\left\langle \left[A_{F,G}\tilde{K}_z\right](x), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = \tilde{K}_z(y). \tag{2.3.4}$$

A few remarks regarding definition 2.3.1 are in order. The kernel propagation operator $A_{F,G}$ is composition-like in the sense that if a linear kernel $\tilde{K}_z(x) = z^\top x$ is used, one could define $A_{F,G}$ explicitly as $A_{F,G}\tilde{K}_z = [\tilde{K}_z(F(\cdot)), \tilde{K}_z(G_1(\cdot)), \cdots, \tilde{K}_z(G_m(\cdot))]$. In that case, due to linearity of the kernel, $\left\langle \left[A_{F,G}\tilde{K}_z\right](x), \begin{pmatrix} 1 & u^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = \tilde{K}_z(F(x)) + \sum_{j=1}^{m} \tilde{K}_z(G_j(x))u_j = \tilde{K}_z(y)$. That is, similar to the Koopman operator for autonomous systems, the operator $A_{F,G}$, when composed with the inner product operation in a RKHS with a linear kernel, propagates the observable $\tilde{K}_z$ one step forward in time.

In the case of nonlinear kernels, an explicit expression for the operator $A_{F,G}$ cannot be derived. However, the implicit definition above, which achieves one-step propagation of the kernels by definition, is still useful for DMD.

Since the set D is dense in H, the kernel propagation operator $A_{F,G}$ is densely defined. As such, the adjoint $A^*_{F,G}$ exists and can be defined through its domain.

Definition 2.3.2 *The domain of the adjoint $A^*_{F,G}$ of $A_{F,G}$ is defined as $\mathcal{D}(A^*_{F,G}) \coloneqq \{f \in H : h \mapsto \langle A_{F,G}h, f\rangle_H$ is bounded on $\mathcal{D}(A_{F,G})\}$.*

Note that for all $x \in X$ and $\bar{u} \in \mathcal{Y}$, the kernel functions $K_{x,\bar{u}}$ of H are in the domain of the adjoint $A^*_{F,G}$. Indeed, if $A_{F,G}h \in H$, $\langle A_{F,G}h, K_{x,\bar{u}}\rangle$ is bounded by definition 2.1.2 and hence $K_{x,\bar{u}} \in \mathcal{D}(A^*_{F,G})$.

2.3.2 Multiplication Operators

Let $\nu : X \to \mathbb{R}^{1\times(m+1)}\mathcal{Y}$ be a continuous function. The multiplication operator with symbol ν is denoted as $M_\nu : \mathcal{D}(M_\nu) \to \tilde{H}$. For a function $h \in \mathcal{D}(M_\nu)$, we define the action of the

multiplication operator on h as

$$[M_\nu h](\cdot) = \langle h(\cdot), \nu(\cdot) \rangle_{\mathcal{Y}},$$

where the domain of the multiplication operator is given as

$$\mathcal{D}(M_\nu) \coloneqq \{h \in H \mid x \mapsto \langle h(x), \nu(x) \rangle_{\mathcal{Y}} \in \tilde{H}\}.$$

For completeness, we recall the interaction between multiplication operators and kernel operators from [30]. The interaction is used to calculate the finite-rank representation of the composition of the multiplication operator with the kernel propagation operator from Definition 2.3.1.

Proposition 2.3.2 *Suppose that $\nu : X \to \mathbb{R}(1 \times (m+1)$ corresponds to a densely defined multiplication operator $M_\nu : \mathcal{D}(M_\nu) \to \tilde{H}$ and $\tilde{K} : X \times X \to \mathbb{R}$ is the kernel function of the RKHS $\tilde{H}$. Then, for all $x \in X$, $\tilde{K}_x \in \mathcal{D}(M_\nu^*)$, where M_ν^* is the adjoint of M_ν, and $M_\nu^* \tilde{K}_x = K_{x,\nu(x)}$.*

The composition of the kernel propagation operator from Definition 2.3.1 and the multiplication operator can be used to define the discrete control Liouville operator.

2.3.3 The Discrete Control Liouville Operator

Taking the composition of $A_{F,G}$ and M_ν, for a known feedback law $\mu : \mathbb{R}^n \to \mathbb{R}^m$, the evolution of an observable along trajectories of the dynamical system can be described in terms of an infinite-dimensional linear operator.

Definition 2.3.3 *Let $\nu \coloneqq \begin{pmatrix} 1 & \mu^\top \end{pmatrix} \in H$. The discrete control Liouville operator corresponding to the closed-loop system*

$$x_{k+1} = F(x_k) + G(x_k)\mu(x_k)$$

is defined as the composition $M_\nu A_{F,G} : \mathcal{D}(M_\nu A_{F,G}) \to \tilde{H}$, where $\mathcal{D}(M_\nu A_{F,G}) = \mathcal{D}(A_{F,G})$.

The discrete control Liouville operator governs the flow of observables in $\mathcal{D}(M_\nu A_{F,G}) \subseteq \tilde{H}$ along trajectories of the discrete-time dynamical system as

$$[M_\nu A_{F,G} h](x_k) = \left\langle [A_{F,G} h](x_k), \begin{pmatrix} 1 & \mu(x_k)^\top \end{pmatrix} \right\rangle_{\mathcal{Y}} = h(x_{k+1}).$$

Furthermore, the composition $M_\nu A_{F,G}$ is a linear operator by linearity of the inner product and by definition 2.3.1.

2.4 Discrete-time control Liouville DMD

In order to represent the infinite-dimensional discrete control Liouville operator as a finite-dimensional operator, we select bases $\alpha = \left\{\tilde{K}_{x_i}\right\}_{i=1}^n \subset \tilde{H}$ and $\beta = \{K_{x_i,\bar{u}_i}\}_{i=1}^n \subset H$, where $\bar{u}_i \coloneqq \begin{pmatrix} 1 & u_i^\top \end{pmatrix} \in \mathcal{Y}$. DMD is then performed via an eigendecomposition of the finite-dimensional representation.

Given an observable $h \in \tilde{H}$, let $\tilde{h} \coloneqq P_\alpha h = \sum_{i=1}^n \tilde{a}_i \tilde{K}_{x_i}$ be the projection of h onto $\operatorname{span} \alpha$. One can recover a finite rank proxy of the discrete control Liovuille operator by observing its action restricted to $\operatorname{span} \alpha \subset \tilde{H}$ and projecting the output $M_\nu A_{F,G} \tilde{h}$ back onto $\operatorname{span} \alpha$. That is, recovering the finite-rank proxy amounts to writing $P_\alpha M_\nu A_{F,G} \tilde{h}$ as $\sum_{i=1}^n \tilde{b}_i \tilde{K}_{x_i}$ and finding a matrix that relates the coefficients $\{\tilde{a}_i\}_{i=1}^n$ and $\{\tilde{b}_i\}_{i=1}^n$. For brevity of notation, let $\tilde{a} \coloneqq \begin{pmatrix} \tilde{a}_1 & \dots & \tilde{a}_n \end{pmatrix}^\top$ and $\tilde{b} \coloneqq \begin{pmatrix} \tilde{b}_1 & \dots & \tilde{b}_n \end{pmatrix}^\top$. The coefficients can be computed by solving the linear system of equations (see [30] and [5])

$$\tilde{G} \begin{pmatrix} \tilde{b}_1 \\ \vdots \\ \tilde{b}_n \end{pmatrix} = \begin{pmatrix} \langle M_\nu P_\beta A_{F,G} \tilde{h}, \tilde{K}_{x_1} \rangle_{\tilde{H}} \\ \vdots \\ \langle M_\nu P_\beta A_{F,G} \tilde{h}, \tilde{K}_{x_n} \rangle_{\tilde{H}} \end{pmatrix}, \tag{2.4.1}$$

where $\tilde{G} = \{\tilde{K}(x_i, x_j)\}_{i,j=1}^n$ is the kernel gram matrix for α. Since the kernel functions in $\alpha \subset \tilde{H}$ are in the domain of the adjoint of the multiplication operator (see proposition 2.3.2),

for all j, $\langle M_\nu P_\beta A_{F,G}\tilde{h}, \tilde{K}_{x_j}\rangle_{\tilde{H}} = \langle A_{F,G}\tilde{h}, P_\beta M_\nu^* \tilde{K}_{x_j}\rangle_H$. Furthermore, by linearity of $A_{F,G}$,

$$\begin{aligned}\langle A_{F,G}\tilde{h}, P_\beta M_\nu^* \tilde{K}_{x_j}\rangle_H &= \sum_{i=1}^{n} \tilde{a}_i \langle A_{F,G}\tilde{K}_{x_i}, P_\beta M_\nu^* \tilde{K}_{x_j}\rangle_H \\ &= \sum_{i=1}^{n} \tilde{a}_i \langle A_{F,G}\tilde{K}_{x_i}, \sum_{k=1}^{n} w_{k,j} K_{x_k,\bar{u}_k}\rangle_H,\end{aligned}$$

where $\{w_{k,j}\}_{k=1}^n$ are weights in the projection of $M_\nu^*\tilde{K}_{x_j}$ onto span β and

$$w_j \coloneqq \begin{pmatrix} w_{1,j} & \dots & w_{n,j}\end{pmatrix}^\top.$$

Thus,

$$\begin{aligned}\langle A_{F,G}\tilde{h}, P_\beta M_\nu^* \tilde{K}_{x_j}\rangle_H &= \sum_{i,k=1}^{n} \tilde{a}_i w_{k,j}^* \langle A_{F,G}\tilde{K}_{x_i}, K_{x_k,\bar{u}_k}\rangle_H \\ &= \sum_{i=1}^{n}\sum_{k=1}^{n} \tilde{a}_i w_{k,j}^* \langle [A_{F,G}\tilde{K}_{x_i}](x_k), \begin{pmatrix} 1 & u_k{}^\top\end{pmatrix}\rangle_{\mathcal{Y}} = \tilde{a}^\top \tilde{I} w_j^*,\end{aligned}$$

where $\tilde{I} = \left(\left\langle [A_{F,G}\tilde{K}_{x_i}](x_k), \begin{pmatrix} 1 & u_k{}^\top\end{pmatrix}\right\rangle_{\mathcal{Y}}\right)_{i,k=1}^{n}$ is computed using the fact that

$$\left\langle [A_{F,G}\tilde{K}_{x_i}](x_k), \begin{pmatrix} 1 & u_k{}^\top\end{pmatrix}\right\rangle_{\mathcal{Y}} = \tilde{K}_{x_i}(x_{k+1}).$$

Note that w^* denotes the complex conjugate of w.

Since M_ν^* maps $\tilde{K}_{x_j}$ to $K_{x_j,\nu(x_j)}$, the coefficients w_j in the projection of $K_{x_j,\nu(x_j)}$ onto span $\beta \subset H$ are solutions of

$$G\begin{pmatrix} w_{1,j} \\ \vdots \\ w_{n,j}\end{pmatrix} = \begin{pmatrix} \langle K_{x_j,\nu(x_j)}, K_{x_1,\bar{u}_1}\rangle_H \\ \vdots \\ \langle K_{x_j,\nu(x_j)}, K_{x_n,\bar{u}_n}\rangle_H\end{pmatrix}, \tag{2.4.2}$$

where $G = \left(\left\langle K_{x_i,\bar{u}_i}, K_{x_j,\bar{u}_j}\right\rangle_H\right)_{i,j=1}^n$ and $\nu(x_j) = \begin{pmatrix} 1 & \mu(x_j)^\top\end{pmatrix}$. If a diagonal kernel operator $K_{x_i} \coloneqq \operatorname{diag}\begin{pmatrix} \tilde{K}_{x_1} & \dots & \tilde{K}_{x_{m+1}}\end{pmatrix}$ is used, with $\tilde{K}_{x_j} = \tilde{K}_{x_i}$ for $1 \le j \le m+1$, then the inner products in G can be computed as

$$\langle K_{x_i,\bar{u}_i}, K_{x_j,\bar{u}_j}\rangle_H = \langle K_{x_i,\bar{u}_i}(x_j), \begin{pmatrix} 1 & u_j^\top\end{pmatrix}\rangle_{\mathcal{Y}} = \begin{pmatrix} 1 & u_i^\top\end{pmatrix} \tilde{K}(x_j, x_i) \begin{pmatrix} 1 & u_j^\top\end{pmatrix}^\top. \tag{2.4.3}$$

Letting $I_j^\top$ denote the column vector on the right-hand side of (2.4.2), he jth row of I is given by

$$I_j = \left(\langle K_{x_j,\nu(x_j)}, K_{x_1,\bar{u}_1}\rangle_H, \ldots, \langle K_{x_j,\nu(x_j)}, K_{x_n,\bar{u}_n}\rangle_H\right).$$

The complete finite-rank representation of the DCLDMD operator is then recovered as $[M_\nu P_\beta A_{F,G}]_\alpha^\alpha = \tilde{G}^\dagger I G^\dagger \tilde{I}^\top$, where the subscript α denotes the restriction of the operator to the span α; the superscript α denotes projection of the output onto span α; and $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse.

2.4.1 Discrete Control Liouville Dynamic Mode Decomposition

DMD can be accomplished via an eigendecomposition of the finite-rank proxy of discrete control Liovuille operator. Let $\{v_i, \lambda_i\}_{i=1}^n$ be the eigenvalue-eigenvector pairs of the matrix $[M_\nu P_\beta A_{F,G}]_\alpha^\alpha$. Following [5], if v_j is an eigenvector of the matrix $[M_\nu P_\beta A_{F,G}]_\alpha^\alpha$, then the function $\varphi_j = \sum_{i=1}^n (v_j)_i \tilde{K}_{x_i}$ is an eigenfunction of the operator $P_\alpha M_\nu P_\beta A_{F,G}|_\alpha$. Here the subscript i in $(v_j)_i$ denotes the i-th component of the eigenvector v_j.

If φ_j is an eigenfunction of $P_\alpha M_\nu P_\beta A_{F,G}|_\alpha$ with eigenvalue λ_j, then

$$\varphi_j(x_{k+1}) = M_\nu A_{F,G} \varphi_j(x_k) = \lambda_j \varphi_j(x_k).$$

Hence, the eigenfunctions evolve linearly along the flow. The normalized eigenfunctions are defined as $\hat{\varphi}_j \coloneqq \frac{1}{\sqrt{v_j^{*\top} \tilde{G} v_j}} \sum_{i=1}^n (v_j)_i \tilde{K}_{x_i}$.

Assuming that the j-th component identity function, g_{id}, defined as $g_{\mathrm{id},j}(x) \coloneqq x_j$ is in $\mathcal{D}(M_\nu A_{F,G}) \subset \tilde{H}$, for each $j = 1, 2, \ldots, n$, we can describe the evolution of the full-state observable $g_{\mathrm{id}}(x) = x$ as a linear combination of eigenfunctions of $M_\nu A_{F,G}$. This approach yields a data-driven model of the closed-loop dynamical system as a linear combination of eigenfunctions of the operator $P_\alpha M_\nu P_\beta A_{F,G}|_\alpha$. That is, for a given $x_0 \in X$ we have a pointwise approximation of the flow of the closed-loop system

$$x_{k+1} = F(x_k) + G(x_k)\mu(x_k) \approx \sum_{i=1}^n \lambda_i^k \xi_i \hat{\varphi}_i(x_0). \tag{2.4.4}$$

We refer to the vectors ξ_i as the *Liouville Modes*, these are the coefficients required to represent the full-state observable as a linear combination of the eigenfunctions. We can calculate the modes by solving $g_{\text{id}}(x) = x = \sum_{i=1}^{n} \xi_i \varphi_i$ for ξ_i, which yields $\xi \coloneqq \begin{pmatrix} \xi_1 & \dots & \xi_n \end{pmatrix} = X(V^\top \tilde{G})^\dagger$, where V is the matrix of normalized eigenvectors of the finite-rank representation $[M_\nu P_\beta A_{F,G}]_\alpha^\alpha$ and $X \coloneqq \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix}$ is the data matrix. We refer to this method as the *direct reconstruction* of the flow.

We can also formulate an *indirect reconstruction* of the flow by considering the function

$$F_\mu \coloneqq x \mapsto \sum_{i=1}^{n} \lambda_i \xi_i \hat{\varphi}_i(x) \tag{2.4.5}$$

that approximates the closed loop dynamics under the feedback law μ as $x_{k+1} \approx F_\mu(x_k)$. The indirect method generally performs better for approximating the nonlinear dynamics; we hypothesize that the better performance is due to the fact we are estimating nonlinear dynamics using nonlinear functions, as the indirect reconstruction yields a nonlinear model of the flow, as opposed to the direct reconstruction, where the estimated model is linear. Due to its superior performance, we will use the indirect reconstruction in the numerical experiments in section 2.6. The DCLDMD algorithm is summarized in Algorithm 1.

2.5 Convergence Properties of DCLDMD

Discrete control Liouville DMD enjoys convergence guarantees on par with current state-of-the-art Koopman methods. That is, the sequence of finite-rank operators $P_\alpha^n M_\nu P_\beta^n A_{F,G} P_\alpha^n$, where P^n denotes the projection onto the n-dimensional span of α and β, respectively, converges to the operator $M_\nu A_{F,G}$ in the *strong operator topology* (SOT). Underlying this fact is the assumption that as $n \to \infty$, the Gram matrices $\tilde{G}$ and G do not become rank deficient.

Theorem 2.5.1 *If $A_{F,G} : \tilde{H} \to H$ and $M_\nu : \mathcal{D}(M_\nu) \to \tilde{H}$ are bounded, and $\alpha \coloneqq \{\tilde{K}_{x_n}\}_{n=1}^\infty \subset \tilde{H}$ and $\beta \coloneqq \{K_{x_n, \bar{u}_n}\}_{n=1}^\infty \subset H$ are two orthonormal sequences in $\tilde{H}$ and H, respectively, then for all $f \in \tilde{H}$, $\lim_{n\to\infty} \left\| P_\alpha^n M_\nu P_\beta^n A_{F,G} P_\alpha^n f - M_\nu A_{F,G} f \right\|_{\tilde{H}} = 0$.*

Proof. Suppose $f \in \tilde{H}$, then

$$\begin{aligned}
&\left\|P_\alpha^n M_\nu P_\beta^n A_{F,G} P_\alpha^n f - M_\nu A_{F,G} f\right\|_{\tilde{H}} = \\
&\qquad \left\|(P_\alpha^n - I) M_\nu P_\beta^n A_{F,G} P_\alpha^n f + M_\nu (P_\beta^n A_{F,G} P^n f - A_{F,G} f)\right\|_{\tilde{H}} \\
&\qquad \leq \left\|(P_\alpha^n - I)(M_\nu P_\beta^n A_{F,G} P_\alpha^n f - M_\nu A_{F,G} f)\right\|_{\tilde{H}} + \\
&\qquad \|(P_\alpha^n - I) M_\nu A_{F,G} f\|_{\tilde{H}} + \left\|M_\nu (P_\beta^n A_{F,G} P_\alpha^n f - A_{F,G} f)\right\|_{\tilde{H}} \\
&\qquad \leq \|(P_\alpha^n - I)\|_{op} \left\|(M_\nu P_\beta^n A_{F,G} P_\alpha^n f - M_\nu A_{F,G} f)\right\|_{\tilde{H}} + \\
&\qquad\qquad \|(P_\alpha^n - I) M_\nu A_{F,G} f\|_{\tilde{H}} + \left\|M_\nu (P_\beta^n A_{F,G} P_\alpha^n f - A_{F,G} f)\right\|_{\tilde{H}},
\end{aligned}$$

where $\|\cdot\|_{op}$ denotes the operator norm. Since M_ν is continuous and $\|(P_\alpha^n - I)\|_{op}$ is bounded (by Parseval's identity, see [24, Section 3.1.11]), and since $P_\beta^n A_{F,G} P_\alpha^n$ converges to $A_{F,G}$ in the SOT [24, Page 172]), the first and the third terms in the inequality above converge to 0 as $n \to \infty$. The fact that P_α^n converges to I in the SOT implies the convergence of the second term to zero. Therefore, the sequence of operators $P_\alpha^n M_\nu P_\beta^n A_{F,G} P_\alpha^n$ converges to $M_\nu A_{F,G}$ in the SOT. ■

Convergence in the SOT does not guarantee convergence of the spectrum, but by theorem 4 in [11], it does guarantee that there is a subsequence of eigenvalue-eigenfunction pairs of the finite-rank representation which converges to an eigenvalue-eigenfunction pair of the true discrete control Liouville operator.

2.6 Numerical Experiments

As a demonstration of the efficacy of the developed DCLDMD algorithm, we apply the method to the controlled Duffing oscillator and compare it with the linear predictor developed in [12].

Experiment 1: The controlled Duffing oscillator is a nonlinear dynamical system with

state-space form

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -\delta x_2 - \beta x_1 - \alpha x_1^3 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 + \sin(x_1) \end{pmatrix} u \tag{2.6.1}$$

where α, β, δ are coefficients in $\mathbb{R}$, $[x_1, x_2]^\top \in \mathbb{R}^2$ is the state, and $u \in \mathbb{R}$ is the control input. For the experiments the parameters are selected to be: $\delta = 0$, $\alpha = 1$, and $\beta = -1$.

We descretize (2.6.1) using a time step of 0.01 seconds to yield a discrete-time, control-affine dynamical system of the form $x_{k+1} = F(x_k) + G(x_k)u_k$. Using the data set consisting of tuples $\{(x_k, x_{k+1}, u_k)\}_{k=1}^n$ generated by the dynamical system, we aim to predict the response of the system starting from the initial condition $x_0 = [2, -2]^\top$ to two different feedback laws, $\mu(x_k) = -2x_{k,1} - x_{k,2}$ and $\bar{\mu}(x_k) = -2x_{k,1}^3 - x_{k,2}$ for a total of 5 seconds.

In the implementation of DCLDMD for the linear feedback law, μ, we generate 225 data points $\{(x_k, x_{k+1}, u_k)\}_{k=1}^{225}$ with initial conditions sampled from a 15×15 grid within the set $[-3, 3] \times [-3, 3] \subset \mathbb{R}^2$. The control inputs are sampled uniformly from the interval $[-2, 2] \subset \mathbb{R}$. For the case of the nonlinear feedback law, $\bar{\mu}$, we generate 1225 data points from initial conditions sampled from a 35×35 grid within the set $[-5, 5] \times [-5, 5] \subset \mathbb{R}^2$ and the control input are sampled uniformly from the interval $[-8, 8] \subset \mathbb{R}$.

In both implementations of DCLDMD, the Gaussian kernel $\tilde{K}(x, y) = \mathrm{e}^{\frac{-\|x-y\|_2^2}{\sigma}}$ is used for calculation of the Gram matrices associated with $\alpha \subset \tilde{H}$. The kernel width is set to $\sigma = 10$ and $\sigma = 20$ for the response of the system to μ and $\bar{\mu}$, respectively. For $\beta \subset H$, we associate to each pair $\{(x_k, u_k)\}_{k=1}^n$ a kernel $K_{x_k, \bar{u}_k} \coloneqq \begin{pmatrix} 1 & u_k^\top \end{pmatrix} K_{x_k} \in H$. Here we use the kernel operator $K_{x_i} \coloneqq \operatorname{diag}\begin{pmatrix} \tilde{K}_{x_1} & \cdots & \tilde{K}_{x_{m+1}} \end{pmatrix}$ where $\tilde{K}_{x_j}(y) = \mathrm{e}^{\frac{-\|x_j - y\|_2^2}{\sigma}}$ for $j = 1, \ldots, m+1$. Lastly, we select $\varepsilon = 10^{-6}$ for regularization of the Gram matrices in order to ensure invertibility of both $\tilde{G}$ and G in the finite-rank representation (see Algorithm 1).

A comparison between the true trajectories and the indirectly reconstructed trajectories corresponding to the feedback laws μ and $\bar{\mu}$ can be seen in figures 2 and 3, respectively.

Experiment 2: In this experiment, we compare the predictive capabilities of the indirect reconstruction via DCLDMD with the linear predictor derived in [12] and the bilinear

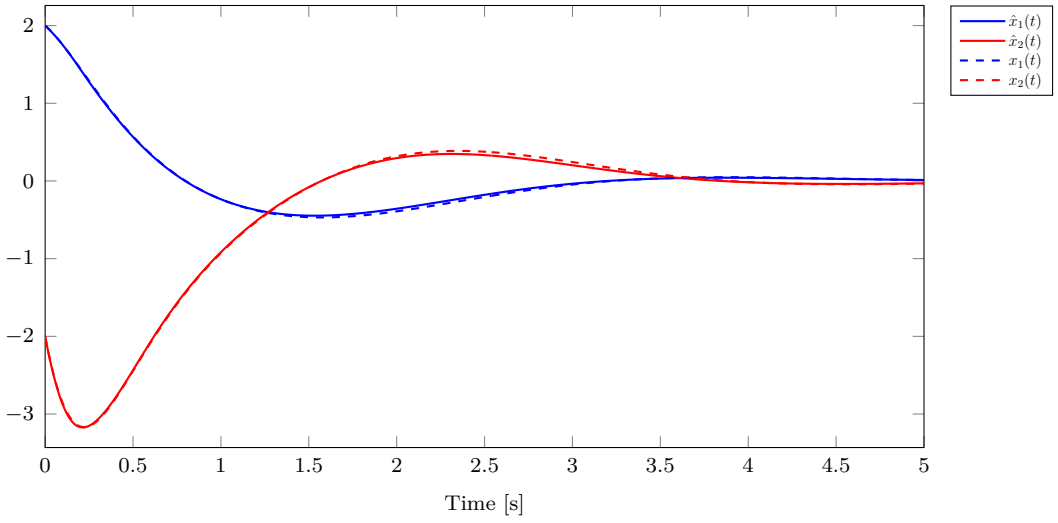

Figure 2: A comparison of indirectly reconstructed trajectories $\hat{x}_1(t)$ and $\hat{x}_2(t)$ with the true trajectories $x_1(t)$ and $x_2(t)$ of the Duffing oscillator resulting from the linear feedback law μ in experiment 1.

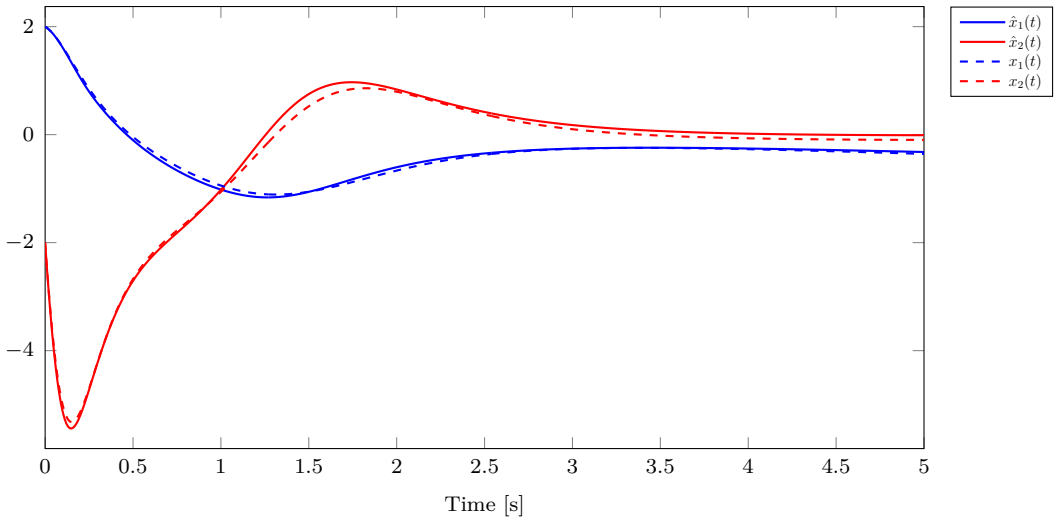

Figure 3: A comparison of indirectly reconstructed trajectories $\hat{x}_1(t)$ and $\hat{x}_2(t)$ with the true trajectories $x_1(t)$ and $x_2(t)$ of the Duffing oscillator resulting from the nonlinear feedback law $\bar{\mu}$ in experiment 1.

Algorithm 1 The DCLDMD algorithm

Input: Data points $\{(x_k, y_k, u_k)\}_{k=1}^n$ that satisfy $y_k = F(x_k) + G(x_k)u_k$, reproducing kernels $\tilde{K}_{x_j}$ and $K_{x_j,\bar{u}_j}$ for $\tilde{H}$ and H, respectively. A feedback law μ, kernel parameter σ, and a regularization parameter ϵ.

Output: $\{\hat{\varphi}_j, \lambda_j, \xi_j\}_{j=1}^n$

1: $\tilde{G} \leftarrow \{\tilde{K}(x_i, x_j)\}_{i,j=1}^n$

2: $\tilde{I} \leftarrow \{\tilde{K}(x_{k+1}, x_i)\}_{k,i=1}^n$

3: $G \leftarrow \{\langle K_{x_i,\bar{u}_i}, K_{x_j,\bar{u}_j}\rangle_H\}_{i,j=1}^n$ (see (2.4.3))

4: $I \leftarrow \{\langle K_{x_j,\nu(x_j)}, K_{x_i,\bar{u}_i}\rangle_H\}_{i,j=1}^n$ (see (2.4.3))

5: Compute $[M_\nu P_\beta A_{F,G}]_\alpha^\alpha = \tilde{G}^\dagger I G^\dagger \tilde{I}^\top$

6: Eigendecomposition: $\{\varphi_j, \lambda_j\}_{j=1}^n \leftarrow [M_\nu P_\beta A_{F,G}]_\alpha^\alpha$

7: Normalize the eigenfunctions: $\{\hat{\varphi}_j\}_{j=1}^n \leftarrow \hat{\varphi}_j = \frac{1}{\sqrt{v_j^{*\top}\tilde{G}v_j}} \sum_{i=1}^n (v_j)_i \tilde{K}_{x_i}$

8: Liouville modes: $\xi \leftarrow X(V^\top \tilde{G})^\dagger$

9: **return** $\{\hat{\varphi}_j, \lambda_j, \xi_j\}_{j=1}^n$

predictor developed in [25]. The linear predictor in [12] is of the form $z_{k+1} = Az_k + Bu_k$ with $x_k = Cz_k$ and z being the lifted state (see [12] for more details). The structure of the bilinear predictor presented in [25] is of the form $z_{k+1} = A + \sum_{i=1}^m u_i B_i z_k$, similarly, $x_k = Cz_k$ is the lifted state (see [25] for more details). For a given feedback law μ, we can estimate the response of the Duffing oscillator described by equation (2.6.1) to the feedback law μ by using the aforementioned linear and bilinear predictors.

For the comparison with the linear predictor, we generate 1000 data points and DCLDMD is implemented using the same kernels as in experiment 1, except the kernel widths are both set to $\sigma = 100$. In the case of the bilinear predictor, we simulate 200 trajectories for a total of 5 seconds. Here, DCLDMD is performed with the Gaussian RBF kernel with kernel parameter $\sigma = 10$. The regularization parameter is set to $\varepsilon = 10^{-6}$ for both simulations. For both the linear and bilinear predictor, extended DMD (eDMD) is performed with the Gaussian radial basis functions as in [12]. For the initial condition $x_0 = [2, -2]^\top$ and the

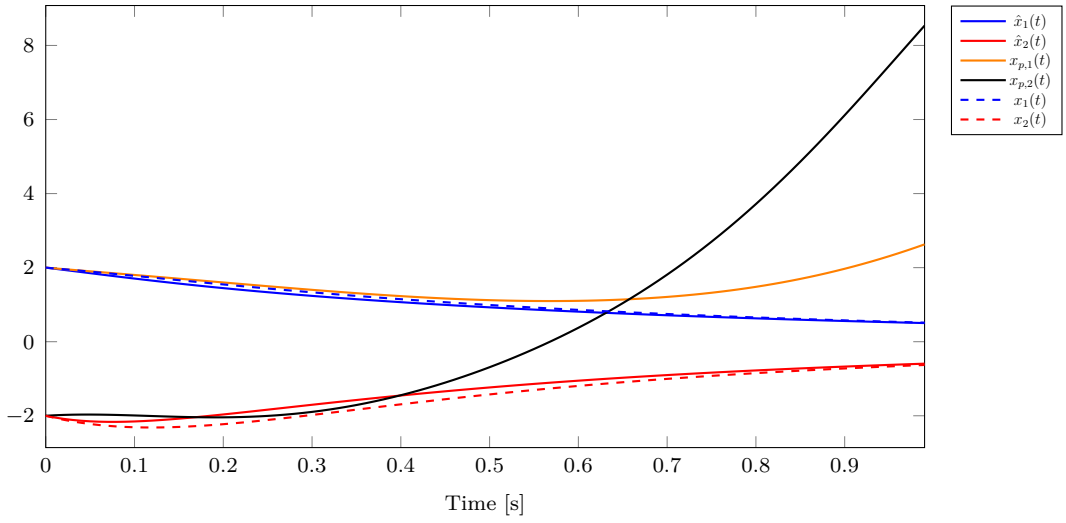

Figure 4: A comparison between the linear predictor developed in [12] and the indirect reconstruction via DCLDMD in experiment 2. Here, $\hat{x}_i(t)$, $x_{p,i}(t)$, and $x_i(t)$ represent the indirect reconstruction, the linear predictor, and the actual trajectories, respectively, where i is a subscript denoting an element of the state.

feedback law $\mu(x_k) = -2x_{k,1} - 2x_{k,2}$, we compare the predictions of the indirect DCLDMD method and the linear predictor with the true trajectories (see Figure 4). Furthermore, in Figure 5 we compare the predictions of the indirect DCLDMD method and the bilinear predictor with the true trajectories.

2.6.1 Discussion

The experiments demonstrate the efficacy of DCLDMD in an academic setting with the Duffing oscillator. The experiments are done with no prior model knowledge, besides the system being affine in control. The novelty of the representation can be seen in the separation of the control input and the state on the operator-theoretic level, while still preserving the nonlinearity of the dynamical system. This is opposed to the standard approach for discrete-time dynamical systems where the lifted state $z_k \in \mathbb{R}^N$ can be approximated as $z_{k+1} \approx Az_k + Bu_k$, with $A \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{1\times N}$ found using extended DMD. Unless the original nonlinear system admits an exact lifting, which is not generally the case, the

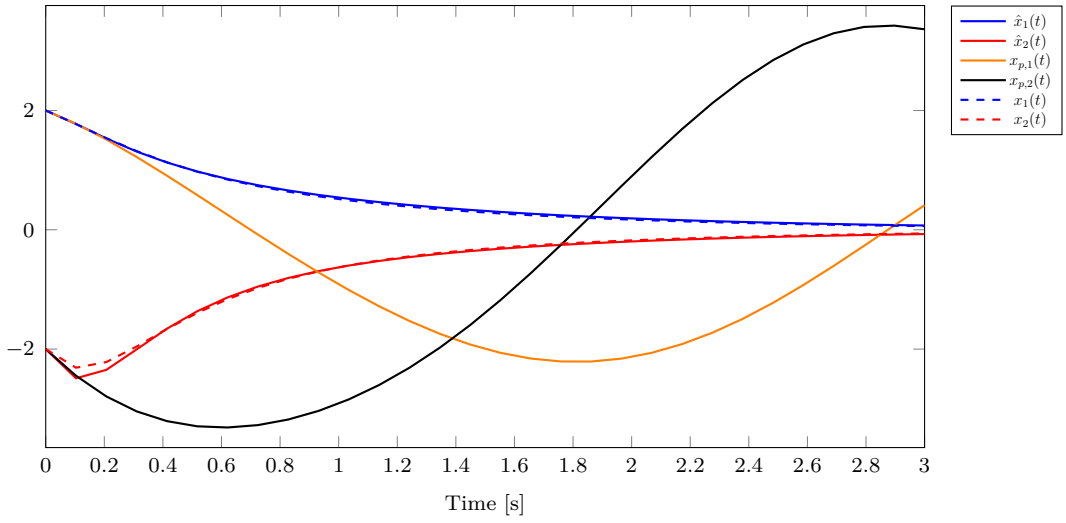

Figure 5: A comparison between the bilinear predictor developed in [25] and the indirect reconstruction via DCLDMD in experiment 2. Here, $\hat{x}_i(t)$, $x_{p,i}(t)$, and $x_i(t)$ represent the indirect reconstruction, the bilinear predictor, and the actual trajectories, respectively, where i is a subscript denoting an element of the state.

trajectories of the linear lifted systems are expected to diverge from the trajectories of the nonlinear system with increasing prediction horizons.

Furthermore, [12] mentions a formulation of bilinear predictors of the form $z_k = Az_k + (Bz_k)u_k$ by exploiting observables which are affine in control; however, [12] does not develop a bilinear predictor in the lifted space, only a formulation for the linear predictor. Therefore, we have to refer to the bilinear predictor developed in [25]. In [25] formulation of the *discrete* bilinear predictor is cogent as long as the discrete-time dynamical system is a discretization of a continous time dynamical system. For continuous dynamical systems, an exact formulation of a bilinear predictor requires strong assumptions, such as access to Koopman invariant subspaces a priori [6].

In experiment 2, specifically, in figure 4, we observe that as expected, the behavior of the linear predictor from [12] and the bilinear predictor from [25] diverge from the behavior of the nonlinear Duffing oscillator under the given feedback law, while the indirect reconstruction approach developed in this chapter accurately tracks the actual trajectory of the Duffing

oscillator. We postulate that the improved predictive capability can be attributed to the fact that the indirect predictive model is a *nonlinear* predictor, as opposed to the models from [12] and [25], which are linear and bilinear, respectively, albeit in a higher dimensional lifted state space.

A comparison between the average decomposition time and the average evaluation time for DCLDMD and the linear predictors in [12] computed over 100 trials is presented in Table 1. The *average decomposition time* is defined as the amount of time required to calculate the matrices A and B in [12] and the amount of time required to construct the indirect reconstruction function (see equation (2.4.5)) in DCLDMD; whereas the *average evaluation time* is the time required by each data driven model to reconstruct a trajectory for 1 second starting from the same initial condition.

Method	Avg. Decomp. time (s)	Avg. Eval. time (s)
DCLDMD	0.8736	0.0552
Linear Predictor	0.0652	0.00014

Table 1: A comparison between the average decomposition time and average evaluation time for the linear predictors developed in [12] and the DCLDMD algorithm.

In this sense, we observe that the linear predictors from [12] require less time to compute than DCLDMD, but the difference in average evaluation time for each method is on the scale of a 100th of a second and thus is marginal. The discrepancy in decomposition time is expected - due to the fact that the formulation of the linear predictors in [12] requires the Moore-Penrose pseudoinverse of only one $n \times m$ matrix (in the case of Experiment 2, $n = 103$ and $m = 1000$) and multiplication by a matrix of size $(n-1) \times m$, whereas DCLDMD requires the inverstion of two $m \times m$ matrices ($\tilde{G}$ and G) and a product of four $m \times m$ matrices (see Algorithm 1). In general, inverting an $n \times n$ matrix and multiplying $n \times n$ matrices has complexity $\mathcal{O}(n^3)$. It is worth noting that the indirect predictor formulated in Experiment 1 requires *far less data* in general (i.e. 225 data points) than what is required

to construct the linear predictors in [12]. Here, 1000 data points are needed to construct the linear predictors from [12].Note that we do not conduct a comparison of the decomposition time and evaluation time with the bilinear predictor from [25]. This is because the predictor in [25] relies on the discrete-time dynamical system being a discretization of a continous-time dynamical system; whereas, DCLDMD and the predictors in [12] are formulated explicity for discrete-time dynamical systems.

In both experiments, indirect reconstruction is used to estimate the flow. The indirect reconstruction explicitly depends upon the eigenfunctions of $P_\alpha M_\nu P_\beta A_{F,G}|_\alpha$. Whether or not we can always represent the full-state observable (i.e. the flow) in terms of the eigenfunctions is not entirely clear, but this is a standard assumption in the DMD literature. With this assumption in mind, DCLDMD is termed a heuristic approach for estimation of the dynamics. Regardless, the numerical experiments in section 2.6 demonstrate the capability of DCLDMD for prediction of the response of the control-affine system to given feedback laws.

CHAPTER III

KERNEL METHODS FOR FAULT DETECTION IN NONLINEAR SYSTEMS

In this chapter, a novel kernelized PCA method is developed. The idea, motivated by results such as [31], is to use trajectories generated by a dynamical system as a fundamental unit of data by embedding them in a reproducing kernel Hilbert space (RKHS) using the so-called occupation kernels. The resulting PCA method, called occupation kernel PCA (OKPCA), is expected to perform better owing to the use of feature maps that are adapted to the data. In addition the computations required to implement OKPCA rely exclusively on integrals of kernel functions evaluated along system trajectories. As a result, OKPCA is endowed with intrinsic robustness to zero-mean noise (since we can think of integration as a first-order filter) and can be implemented on data sets containing variable length trajectories that are irregularly sampled [31]. Fault detection then proceeds by reconstructing a given trajectory as a linear combination of eigenfunctions of a suitably defined kernelized covariance operator and computing a suitable analog of the reconstruction error used for KPCA by Hoffman [7].

3.1 Background

In this section, a brief overview of current PCA methods is provided for completeness.

3.1.1 Principal Component Analysis

Given a set of M *centered* observations $\{x_j \in \mathbb{R}^n\}_{j=1}^M \subset X \subseteq \mathbb{R}^n$, where "centered" indicates that $\sum_{j=1}^M x_j = 0$, the principal component analysis (PCA) procedure diagonalizes

the covariance matrix C defined by $C = \frac{1}{M}\sum_{j=1}^{M} x_j x_j^\top$ where C is at most rank M (if all observation vectors are linearly independent).

Here, it is worth noting that the matrix C is referred to as the covariance matrix due to the fact that the ith entry in the jth column of C is $\mathrm{Cov}(x_i, x_j)$, where

$$\mathrm{Cov}(x_i, x_j) = \frac{\sum_{k=1}^{M}(x_i^k - \bar{x})(x_j^k - \bar{x})}{M},$$

where x_i^k denotes the kth element of the vector x_i and $\bar{x}$ denotes the mean of the dataset $\{x_j\}_{j=1}^{M}$. In our case, we center the data beforehand yielding $\mathrm{Cov}(x_i, x_j) = \frac{1}{M}\sum_{k=1}^{M} x_i^k x_j^k = \frac{1}{M} x_i^\top x_j$, and we see that the entries along the diagonal are $\mathrm{Cov}(x_i, x_i) = \mathrm{Var}(x_i, x_i)$ for each $i = 1, 2, \ldots, M$.

Since C is a positive semi-definite matrix, it is diagonalizable and has nonnegative eigenvalues. The eigenvectors of C are referred to as the *principal components*, typically ordered in a decreasing sequence of the corresponding eigenvalues. Given a vector $v \in \mathbb{R}^n$ we note that $Cv = \frac{1}{M}\sum_{j=1}^{M}\langle x_j, v\rangle x_j$, where $\langle\cdot,\cdot\rangle$ indicates the standard dot product. In particular, if v is an eigenvector of C with eigenvalue λ, we have $Cv = \lambda v = \frac{1}{M}\sum_{j=1}^{M}\langle x_j, v\rangle x_j$, which implies that all eigenvectors of C lie in the span of $\{x_j\}_{j=1}^{M}$.

3.1.2 Kernelized Principal Component Analysis

Kernelized principal component analysis (KPCA) [7] extends the PCA procedure to produce *nonlinear* principal components. This is done by embedding the data into a reproducing kernel Hilbert space (RKHS) via a feature mapping $\Phi : X \subseteq \mathbb{R}^n \to H$.

Definition 3.1.1 *Let X be a nonempty set. A function $k : X \times X \to \mathbb{R}$ is called a* kernel function on X *if there exists a $\mathbb{R}$-Hilbert space H and a map $\Phi : X \to H$ such that for all $x, x' \in X$ we have $k(x, x') = \langle\Phi(x'), \Phi(x)\rangle_H$. We call Φ a feature map and H a feature space of k.*

In other words, the data point x is replaced by a element $\Phi(x)$ in the Hilbert space H and the dot product is replaced by an inner-product over the Hilbert space. It should be noted that

the choice of feature map is *not* unique, however, the Moore-Aronszajn theorem guarantees the existence of a unique RKHS corresponding to k and a canonical feature map that maps into that RKHS in the case where k is a positive semi-definite kernel. For completeness, we recall the definition of an RKHS.

Definition 3.1.2 *A RKHS, H, over a set X is a Hilbert space of real-valued functions over the set X such that for all $x \in X$ the evaluation functional, $E_x : H \to \mathbb{R}$, given as $E_x g := g(x)$ is bounded.*

The Riesz representation theorem guarantees, for all $x \in X$, the existence of a unique function $k_x \in H$ such that $\langle g, k_x \rangle_H = g(x)$, where $\langle \cdot, \cdot \rangle_H$ is the inner product for H [23, Chapter 1]. The function k_x is called the *reproducing kernel centered at* x, the function $k(x, y) = \langle k_y, k_x \rangle_H$ is called the *reproducing kernel of* H and the mapping $\Phi : X \to H$ given by $x \mapsto k(\cdot, x) = \Phi(x)$, is called the *canonical feature map*.

In this setting we can now define *nonlinear* principal components via analogous constructions. Given a feature map $\Phi : X \subseteq \mathbb{R}^n \to H$ and a set of data $\{x_j\}_{j=1}^M$ centered in H, i.e. $\sum_{j=1}^M \Phi(x_j) = 0$, the kernelized covariance operator $C : H \to H$ is defined as $C = \frac{1}{M} \sum_{j=1}^M [\Phi(x_j) \otimes \Phi(x_j)]$, where, the notation $[u \otimes v]$, for $u, v \in H$, denotes the rank one operator defined by $[u \otimes v]h = \langle h, v \rangle u$ for $h \in H$.

It is worth noting that C is a finite rank and positive semi-definite operator and thus diagonalizable. If v is an eigenfunction of C then automatically $v \in \text{span}\{\Phi(x_j) : j = 1, \ldots, M\}$ and $v = \sum_{j=1}^M \alpha_j \Phi(x_j)$ for $\alpha_j \in \mathbb{R}$. The coefficients α_i can be computed by solving a matrix equation, indeed for an eigenfunction $v \in H$, $\langle \Phi(x_k), Cv \rangle_H = \langle \Phi(x_k), \lambda v \rangle_H$, which, along with

$$\langle \Phi(x_k), \lambda v \rangle_H = \lambda \sum_{i=1}^M \langle \Phi(x_k), \alpha_i \Phi(x_i) \rangle,$$

implies by definition of $\otimes$ that

$$\langle \Phi(x_k), Cv \rangle_H = \sum_{i,j=1}^M \frac{\alpha_i \langle \Phi(x_j), \Phi(x_i) \rangle_H \langle \Phi(x_k), \Phi(x_j) \rangle_H}{M}. \tag{3.1.1}$$

If we define $\alpha = (\alpha_1, \ldots, \alpha_M)^\top$, $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle_H$ and $K = (k(x_i, x_j))_{i,j=1}^M$, equation (3.1.1) can be expressed in the matrix form

$$M\lambda K\alpha = K^2\alpha. \tag{3.1.2}$$

Since K is a positive semi-definite matrix, it is sufficient to solve the equation $K\alpha = \lambda M\alpha$ to recover all the solutions to (3.1.2). In other words, the vector of coefficients α is a normalized eigenvector of the matrix K.

Let $\alpha^{(1)}, \ldots, \alpha^{(N)}$, for $0 < N \leq M$, be a set of eigenvectors of K, corresponding to nonzero eigenvalues $0 < \lambda_1 \leq \ldots, \leq \lambda_N$, normalized such that for $k = 1, \ldots, N$, $\lambda_k \langle \alpha^{(k)}, \alpha^{(k)} \rangle_{\mathbb{R}^n} = 1$. The $k-$th eigenfunction $v^{(k)}$ of C can then be expressed as $v^{(k)} = \sum_{i=1}^M \alpha_i^{(k)} \Phi(x_i) \in H$.

Definition 3.1.3 *Given a test point $x \in X$, we call $\langle v^{(k)}, \Phi(x) \rangle_H$, where $v^{(k)}$ is an eigenfunction of C, a* nonlinear principal component of $\{x_j\}_{j=1}^M$ at x corresponding to Φ.

Remark 3.1.1 *If the data used for PCA are* uncentered *in H, they can be centered by replacing K with*

$$\tilde{K} = K - J_M K - K J_M + J_M K J_M,$$

where $(J_M)_{i,j} = \frac{1}{M}$.

3.2 Occupation Kernel PCA

In this section, we will appropriately modify KPCA to incorporate trajectories as a fundamental unit of data. To do so will require an embedding of trajectories into a RKHS. The embedding will be achieved by using the novel occupation kernels developed in [31], and the resulting technique will be called OKPCA.

Definition 3.2.1 *Let $X \subset \mathbb{R}^n$ be compact, H be a RKHS of real-valued continuous functions over X, and $\gamma \in C([0, T], X)$ be a trajectory, where $C([0, T], X)$ denotes the set of continuous functions from $[0, T]$ to X. The functional $g \mapsto \int_0^T g(\gamma(\tau)) \mathrm{d}\tau$ is bounded, and*

may be represented as $\int_0^T g(\gamma(\tau))\mathrm{d}\tau = \langle g, \Gamma_\gamma \rangle_H$, *for some* $\Gamma_\gamma \in H$ *by the Riesz representation theorem. The function* Γ_γ *is called the* occupation kernel corresponding to γ in H *[31].*

The occupation kernel corresponding to a trajectory can be shown to be the integral of a kernel function along the trajectory.

Proposition 3.2.1 *[31] Let* H *be a RKHS of real-valued continuous functions over a set* X *and let* $\gamma : [0, T] \to X$ *be a continuous trajectory as in Definition 3.2.1. The occupation kernel corresponding to* γ *in* H, Γ_γ, *may be expressed as*

$$\Gamma_\gamma(x) = \int_0^T K(x, \gamma(t))\mathrm{d}t. \tag{3.2.1}$$

Proof. Note that $\Gamma_\gamma(x) = \langle \Gamma_\gamma, K(\cdot, x) \rangle_H$, by the reproducing property of K. Consequently,

$$\begin{aligned}\Gamma_\gamma(x) &= \langle \Gamma_\gamma, K(\cdot, x) \rangle_H = \langle K(\cdot, x), \Gamma_\gamma \rangle_H \\ &= \int_0^T K(\gamma(t), x)\,\mathrm{d}t = \int_0^T K(x, \gamma(t))\,\mathrm{d}t,\end{aligned}$$

which establishes the result. ■

A kernelized covariance operator can now be defined for a set of trajectories.

Definition 3.2.2 *Let* H *be a Hilbert space,* $\Gamma = \{\gamma_i : [0, T] \to X\}_{i=1}^M$ *be a finite set of trajectories and* $\Phi : C([0, T], X) \to H$ *be a feature map taking trajectories into* H. *With* $\{\Phi(\gamma_j) : j = 1, \dots M\}$ *centered in* H, *i.e.* $\sum_{j=1}^M \Phi(\gamma_j) = 0$, *define the kernelized covariance operator as*

$$C_\Gamma = \frac{1}{M} \sum_{j=1}^M [\Phi(\gamma_j) \otimes \Phi(\gamma_j)].$$

Similar to the kernelized covariance operator C above, C_Γ is a positive semi-definite finite rank operator and as a result, admits eigenfunctions of the form $v^{(k)} = \sum_{i=1}^M \alpha_i^{(k)} \Phi(\gamma_i)$. The notion of nonlinear principal components then extends naturally to Hilbert spaces.

Definition 3.2.3 *Given a test trajectory* $\gamma : [0, T] \to X$ *and a corresponding feature map* $\Phi : C([0, T], X) \to H$, *we call* $\langle v^{(k)}, \Phi(\gamma) \rangle_H$, *where* $v^{(k)}$ *is an eigenfunction of* C_Γ, *a* nonlinear principal component of Γ at γ corresponding to Φ.

While the principal components can be defined with respect to any feature map, the occupation kernels themselves provide a feature map that is convenient for analysis and implementation. The convenience stems from the fact that if the occupation kernels are selected to be the feature maps, the coefficients $\alpha_i^{(k)}$ of the eigenfunction $v^{(k)}$ of C_Γ are given by normalized eigenvectors of the Gram matrix of occupation kernels.

Proposition 3.2.2 *The mapping $\Phi(\gamma) = \Gamma_\gamma$ is a feature map from $C([0,T], X)$ to H. The eigenfunctions of C_Γ under this feature map can be computed by solving $\tilde{K}_\Gamma \alpha = \lambda M \alpha$ where $\tilde{K}_\Gamma$ is the* centered *occupation kernel Gram matrix, given by*

$$\tilde{K}_\Gamma = K - J_M K - K J_M + J_M K J_M$$

where $(J_M)_{i,j} = \frac{1}{M}$ and $K = \left(\langle \Gamma_{\gamma_i}, \Gamma_{\gamma_j} \rangle_H\right)_{i,j=1}^M$ is the original occupation kernel Gram matrix. In particular, if $\alpha^{(1)}, \ldots, \alpha^{(N)}$, for $0 < N \leq M$, are eigenvectors of $\tilde{K}_\Gamma$, corresponding to nonzero eigenvalues $0 < \lambda_1 \leq \ldots, \leq \lambda_N$, normalized such that for $k = 1, \ldots, N$, $\lambda_k \langle \alpha^{(k)}, \alpha^{(k)} \rangle_{\mathbb{R}^n} = 1$, then the $k-$th eigenfunction $v^{(k)}$ of C_Γ can be expressed as

$$v^{(k)} = \sum_{i=1}^{M} \alpha_i^{(k)} \Gamma_{\gamma_i} \in H.$$

Proof. The proof of the above proposition proceeds analogously to what is done in KPCA. We need only note that $\langle \Phi(\gamma_i), \Phi(\gamma_j) \rangle_H = \langle \Gamma_{\gamma_i}, \Gamma_{\gamma_j} \rangle_H$ and that $\tilde{K}_\Gamma = (J_M - I) K (J_M - I)$ is positive semi-definite. ■

3.2.1 OKPCA for Fault Detection

Here we will outline an interesting application of OKPCA to detect faulty trajectories based on Hoffman's reconstruction error [7].

Definition 3.2.4 *Let γ be a test trajectory, $\Gamma = \{\gamma_j : j = 1, \ldots M\}$ be a collection of trajectories, $V = \{v^{(k)} : k = 1, \ldots N\}$ be a collection of eigenfunctions for C_Γ, and $\Phi_0 = \frac{1}{M} \sum_{j=1}^M \Phi(\gamma_j)$ be the* center *of Γ in H. Letting $\tilde{\Phi}(\gamma) = \Phi(\gamma) - \Phi_0$ we can define the*

reconstruction error for γ in H with respect to V *by*

$$R(\gamma) = \|\tilde{\Phi}(\gamma)\|_H^2 - \sum_{j=1}^{N} \langle \tilde{\Phi}(\gamma), v^{(j)} \rangle_H^2. \tag{3.2.2}$$

Remark 3.2.1 *If the feature maps are selected to be the occupation kernels, the reconstruction error can be computed using integrals of the kernel function along the trajectory. Indeed, using the feature map* $\Phi(\gamma) = \Gamma_\gamma$*, we get*

$$\begin{aligned}\|\tilde{\Phi}(\gamma)\|_H^2 &= \left\langle \Phi(\gamma) - \sum_{j=1}^{M} \frac{\Phi(\gamma_j)}{M}, \Phi(\gamma) - \sum_{j=1}^{M} \frac{\Phi(\gamma_j)}{M} \right\rangle_H \\ &= \langle \Gamma_\gamma, \Gamma_\gamma \rangle_H - \sum_{j=1}^{M} \frac{2\langle \Gamma_\gamma, \Gamma_{\gamma_j} \rangle_H}{M} + \sum_{i,j=1}^{M} \frac{\langle \Gamma_{\gamma_i}, \Gamma_{\gamma_j} \rangle_H}{M^2}\end{aligned}$$

and for a given k *we have*

$$\begin{aligned}&\langle \tilde{\Phi}(\gamma), v^{(k)} \rangle_H \\ &= \sum_{j=1}^{M} \alpha_j^{(k)} \left[\langle \Gamma_\gamma, \Gamma_{\gamma_j} \rangle_H - \frac{1}{M} \sum_{n=1}^{M} \langle \Gamma_{\gamma_n}, \Gamma_{\gamma_j} \rangle_H \right. \\ &\left. - \frac{1}{M} \sum_{\ell=1}^{M} \langle \Gamma_\gamma, \Gamma_{\gamma_\ell} \rangle_H + \frac{1}{M^2} \sum_{n,\ell=1}^{M} \langle \Gamma_{\gamma_n}, \Gamma_{\gamma_\ell} \rangle_H \right].\end{aligned}$$

The reconstruction error can then be computed using the fact that given two trajectories γ_i *and* γ_j*, the inner product of the corresponding occupation kernels is given by*

$$\langle \Gamma_{\gamma_i}, \Gamma_{\gamma_j} \rangle_H = \int_0^T \int_0^T k(\gamma_i(\tau), \gamma_j(t)) \, \mathrm{d}\tau \mathrm{d}t.$$

Remark 3.2.2 *Similar to KPCA, the OKPCA reconstruction error also has an interesting geometric interpretation. Note that the reconstruction error can be represented in the inner product form*

$$R(\gamma) = \left\langle \tilde{\Phi}(\gamma), \tilde{\Phi}(\gamma) - \sum_{j=1}^{N} \langle \tilde{\Phi}(\gamma), v^{(j)} \rangle_H v^{(j)} \right\rangle_H.$$

Hence, the reconstruction error is a measure of how well the projection of $\tilde{\Phi}(\gamma)$ *onto span*$\{v^{(j)} : j = 1, \ldots, N\}$ *recreates* $\tilde{\Phi}(\gamma)$*.*

Given a large enough set of normal trajectories, the reconstruction error can thus be used to detect faulty trajectories.

Definition 3.2.5 *Let* $\Gamma = \{\gamma_j : j = 1, \dots M\}$ *be a collection of trajectories, called* training data*. Let* $V = \{v^{(k)} : k = 1, \dots N\}$ *denote the* principal component vectors*, i.e., a collection of eigenfunctions for* C_Γ *corresponding to non-zero eigenvalues. Let* $R_V(\gamma)$ *be the reconstruction error for a test trajectory* γ *in* H *with respect to* V*. For a given threshold* $\varepsilon > 0$*, we will call a test trajectory* $\varepsilon-$faulty *if* $R_V(\gamma) > \varepsilon$*.*

Remark 3.2.3 *This definition of fault is dependent on* N*, the number of principal component vectors being used to compute the reconstruction error, the selected kernel, and the threshold* ε*. The threshold* ε *can be decided based on reconstruction errors evaluated at trajectories that are a part of the training data. For further remarks on the selection of the kernel and the number of principal component vectors, see the discussion section.*

3.3 Experiments

In the following, two numerical experiments are presented to illustrate the efficacy of the developed fault detection method. The first experiment is an academic one where the developed method is used to identify trajectories generated by a nonlinear system that is different from the one used to generate the training data.

In the second experiment, simulated trajectories of a quadrotor aircraft are used to train the algorithm. The trained algorithm is then used to identify trajectories generated by a faulty quadrotor, where the fault is introduced by changing control parameters.

3.3.1 Description and Results

Experiment 1: In this experiment, 100 fault detection trials are performed. In each trial, the training data comprises of 100 trajectories of the system

$$\dot{x}_1 = -x_1 + x_2 \sin\left(\frac{\pi x_1}{2}\right), \dot{x}_2 = -x_2 + x_1 \cos\left(\frac{\pi x_1}{2}\right)$$

initialized from randomly selected initial conditions on the unit circle. To test the developed OKPCA fault detection method, the reconstruction error is evaluated at 20 trajectories of the same system and 20 trajectories of the *faulty* system

$$\dot{x}_1 = -x_1 + 0.9x_2 \sin\left(\frac{\pi x_1}{5}\right), \dot{x}_2 = -x_2 + 0.8x_1 \cos\left(\frac{\pi x_2}{3}\right)$$

also starting from random initial conditions on the unit circle.

All trajectories are 2 seconds long and sampled every 0.01 seconds. The Gaussian radial basis function $k(x, y) = \mathrm{e}^{\frac{-\|x-y\|^2}{\mu}}$ is used as the kernel function with width parameter $\mu = 0.6$ and $N = 20$ eigenvectors are selected for the projection in (3.2.2). The detection threshold is set to be equal to 2 times the highest reconstruction error seen in the training data, that is, $\varepsilon = 2 \max_i \{R(\gamma_i)\}_{i=1}^M$. Normal test trajectories with reconstruction errors higher than the threshold are classified as false positives and faulty test trajectories with reconstruction errors smaller than the threshold are classified as false negative. To compare OKPCA and KPCA in a way that is independent of threshold selection, a *mixing percentage* is computed. The mixing percentage is defined as the percentage of the test trajectories that fall within the band defined by the smallest reconstruction error among faulty trajectories and the largest reconstruction error among normal trajectories. The performance of OKPCA and KPCA for this test is summarized in the third column of Table 2

Since the OKPCA method relies on integrals of trajectories, the data do not need to be equally spaced. To demonstrate the applicability of the OKPCA method to data sets with variable sampling rates, a sampling noise, uniformly distributed in the interval $[-0.004, 0.004]$ is added to each sampling instant of the training data and the test data (i.e., the sampling rate is uniformly distributed between 0.002s and 0.01s). The performance of OKPCA and KPCA for this test is summarized in the fourth column of Table 2.

As opposed to PCA, which is generally not robust to noise [14], occupation kernel PCA, owing to integration of the trajectories, is expected to have inherent robustness to zero-mean measurement noise and sampling noise. To test this hypothesis, the 100 trials are repeated

M	Method	No Noise (%)			Samp. Noise (%)			Meas. Noise (%)		
		FP	FN	MP	FP	FN	MP	FP	FN	MP
50	OKPCA	11.5	0.1	2.6	10.2	0.2	4	11.2	0.7	11.9
	KPCA	10.6	0.1	1.9	15	0.3	8.1	11.9	1.8	23.1
100	OKPCA	1.3	0.2	0.7	1.3	0.6	3.9	0.6	1.8	8.8
	KPCA	1.6	0.1	0.9	1.7	0.7	5.5	0.3	3.2	12.4
150	OKPCA	0.2	0.2	0	0.1	0.5	3.6	0.2	1.9	8.2
	KPCA	0.4	0.1	0.5	0.1	0.8	5.2	0.1	3.7	12.6

Table 2: A comparison of OKPCA with KPCA for the system and fault models in Experiment 1. The initialisms FP, FN, and MP denote the false positive rate, the false negative rate, and the mixing percentage, averaged over 100 trials, respectively.

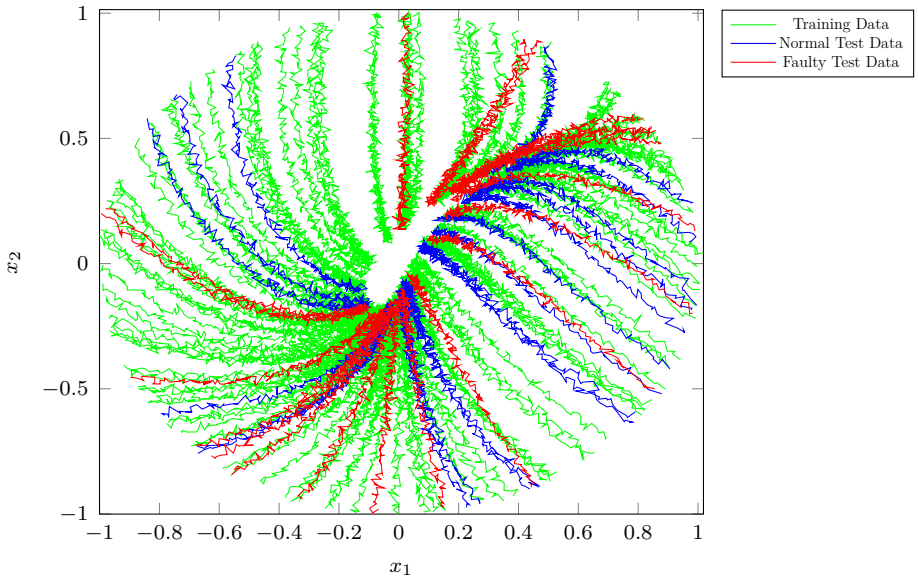

Figure 6: Noisy trajectories used as training data (green) and normal (blue) and faulty (red) test data to test degradation of performance in Experiment 1.

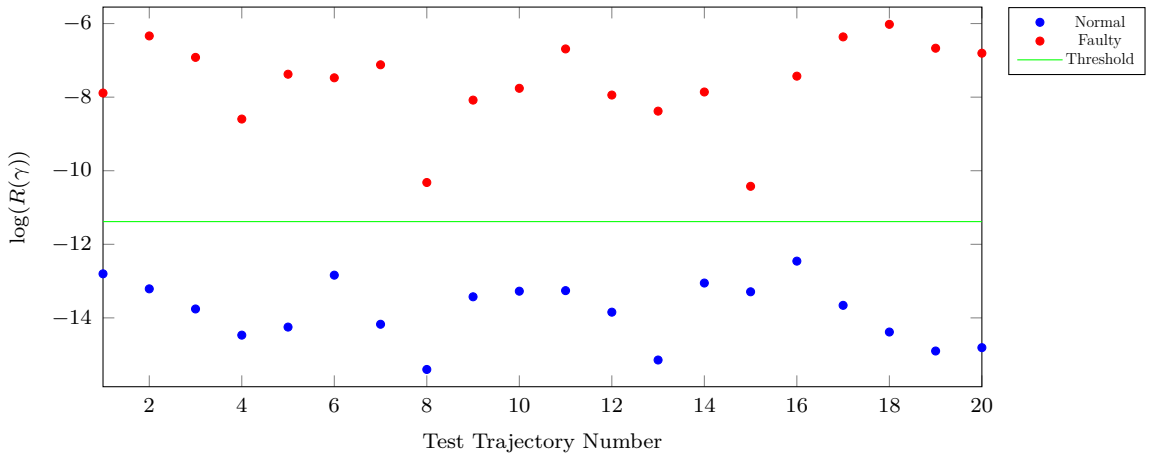

Figure 7: An example trial in Experiment 1 where the faulty trajectories and the normal trajectories are well-separated by the reconstruction error and no false negative or false positive results are generated.

with $M = 50$, 100, and 150 by adding Gaussian noise with standard deviation 0.01 to each measurement in the training data and the test data (see Fig. 6). For comparison, the KPCA fault detection method from [7] is applied to the same data set with 20 eigenvectors and $\mu = 5$. The performance of OKPCA and KPCA for this test is summarized in the last column of Table 2.

Fig. 7 illustrates the results of one of the *successful* (no false positives or false negatives) noisy trials where it can be seen that the faulty test trajectories have a higher reconstruction error than the normal test trajectories. Fig. 8 illustrates the results of one of the *unsuccessful* noisy trials where the decision boundary is not as clear as the successful trial.

Experiment 2: In the second experiment, the fault detection capabilities of OKPCA are evaluated using trajectories generated by a quadrotor. A quadrotor model under a known PID controller is simulated in MATLAB. A simplified model of the quadrotor in the vehicle frame is used by neglecting the Coriolis force and assuming the pitch (θ) and roll (ϕ) angles are small (see Equations $35 - 40$ in [1] for details). The model consists of 12 state variables that include position (x, y, z), velocity (u, v, w), Euler angles (ϕ, θ, ψ), and roll rates (p, q, r) of the quadrotor.

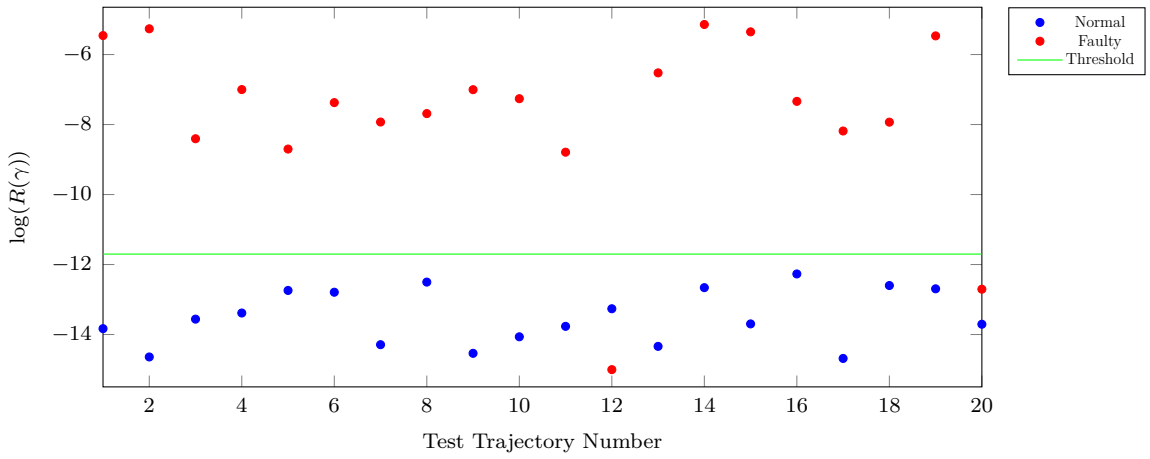

Figure 8: An example trial in Experiment 1 where a few of the faulty trajectories fall below the threshold, generating false negative results.

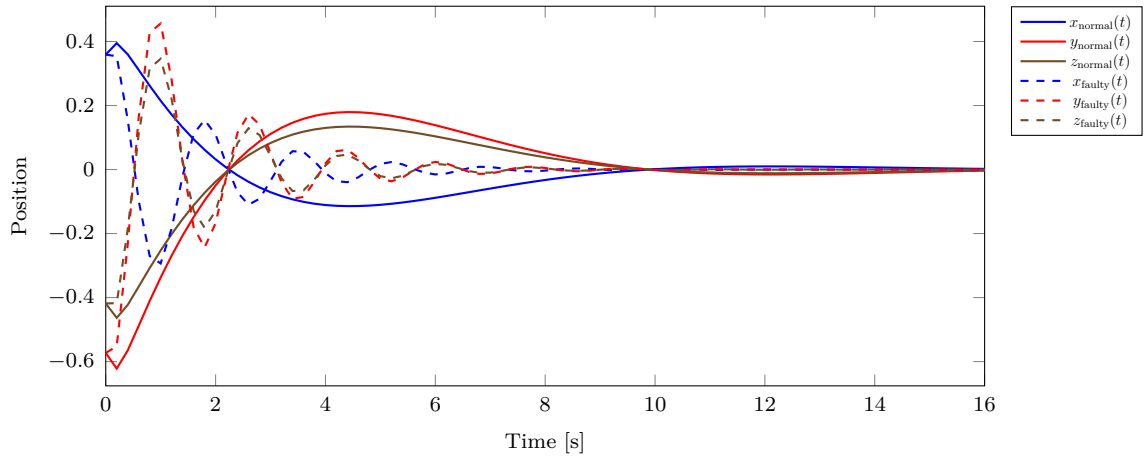

Figure 9: Example of a normal (solid) and a faulty (dotted) trajectory of the quadrotor in Experiment 2 under simulated major actuator fault.

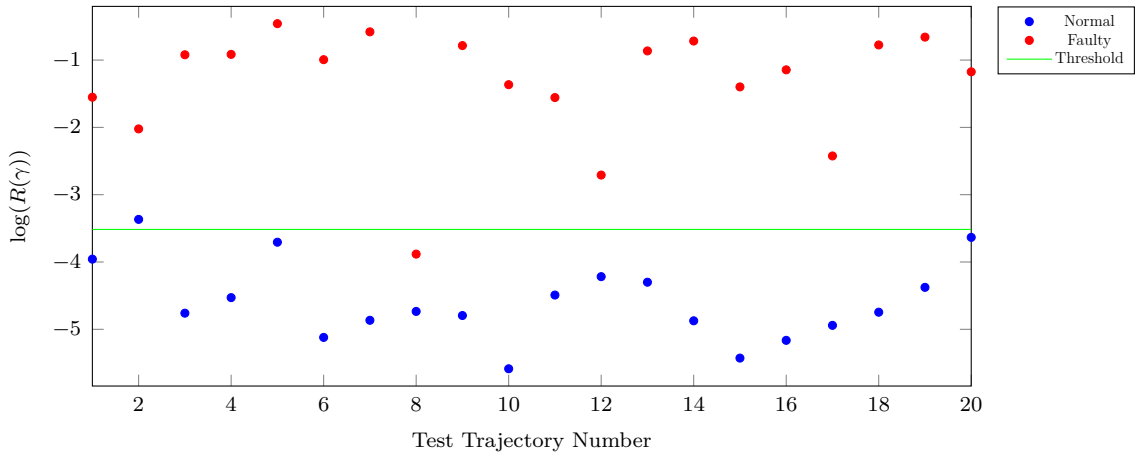

Figure 10: Reconstruction error comparison for major actuator faults in Experiment 2.

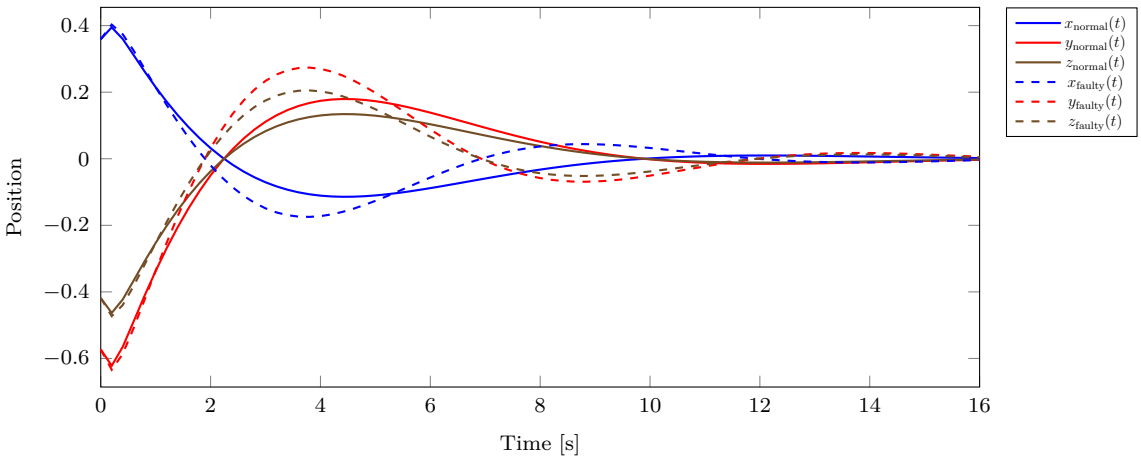

Figure 11: Example of a normal (solid) and a faulty (dotted) trajectory of the quadrotor in Experiment 2 under simulated minor actuator fault.

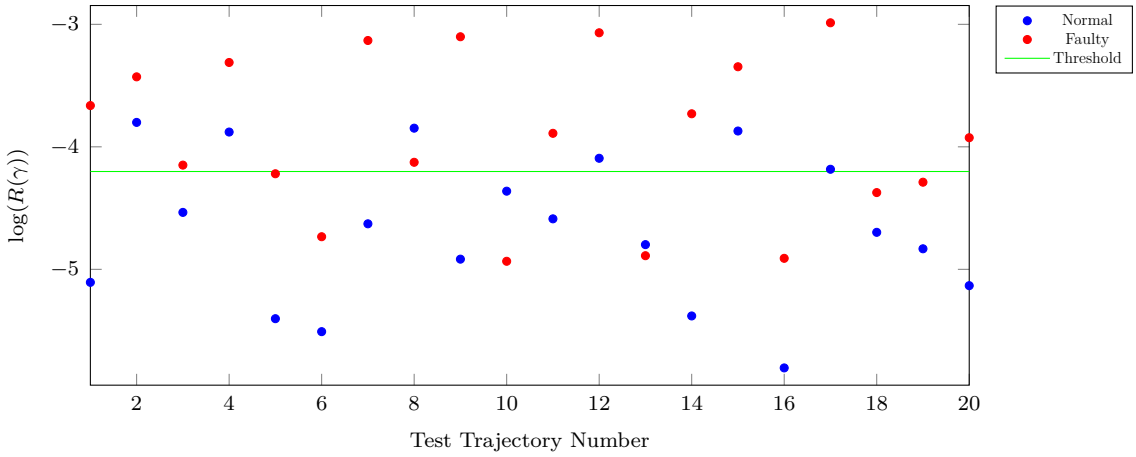

Figure 12: Reconstruction error comparison for minor actuator faults in Experiment 2.

The controller used in the simulation is from Sections 7, 7.2, and 7.3 in [1]. Given a desired setpoint, the controller regulates the quadrotor to the setpoint by manipulating the velocity, pitch, and roll using three separate proportional-integral-derivative (PID) controllers. The control gains are identical for each of the three PID controllers. For the training data the proportional gain K_P, integral gain K_I, and derivative gain K_D were selected to be $5, 2$, and 8, respectively. For examples of noise-free normal and faulty trajectories for the major and the minor faults, see Figs. 9 and 11, respectively.

The algorithm is trained on a data set consisting of 500 trajectories of randomly generated lengths, sampled at approximately 5 Hz. Irregular sampling rates and measurement noise are implemented similar to Experiment 1. Each trajectory is started from a random initial condition in the box with side length 2 centered at the origin in $\mathbb{R}^{12}$ and the quadrotor is commanded to fly to the origin. Actuator faults are simulated by altering the PID gains. To simulate major actuator faults, 20 trajectories are generated using $K_P = 15$, $K_D = 2$, and $K_I = 12$, and the minor actuator faults are simulated by generating another 20 trajectories using $K_P = 4$, $K_D = 7$, and $K_I = 3$.

The Gaussian radial basis function kernel with width parameter $\mu = 10$ is used for OKPCA and $N = 100$ eigenvectors are used for reconstruction. The reconstruction errors for the faulty trajectories are then compared with those corresponding to 20 newly generated

normal trajectories. Fig. 10 and Fig. 12 show the fault detection capabilities for trajectories generated with major and minor actuator faults, respectively. The fault detection threshold is set to be $\varepsilon_{major} = 2\max_i\{R(\gamma_i)\}_{i=1}^{M}$ for the major actuator tests and $\varepsilon_{minor} = \max_i\{R(\gamma_i)\}_{i=1}^{M}$ for the minor actuator tests.

3.3.2 Discussion

The experiments demonstrate the efficacy of OKPCA for data-driven fault detection applications. As noted in Experiment 1, in randomized trials, without any knowledge of the system model or the fault, OKPCA results in reconstruction errors that differentiate faulty trajectories from normal trajectories with less than 1% false positive and false negative rates, with moderate degradation in performance when the data and the sampling rates are corrupted with noise. In addition to the practical advantages of OKPCA over KPCA listed in the introduction, Table 2 also indicates that in most experiments, OKPCA outperforms KPCA in the mixing percentage metric. The false positive and false negative rates depend on the selected threshold, and as such are not suitable for use as a metric for comparison.

The results of Experiment 2 indicate that OKPCA can detect faulty trajectories irrespective of measurement noise and sampling noise. While major actuator faults are detectable with high confidence (Fig. 10), minor actuator faults are hard to detect (Fig. 12). Degradation of performance with decreasing severity of faults is expected in data-driven fault detection methods, especially in the presence of measurement noise.

Similar to Hoffman's observations in [7], too small values of the kernel width, μ, result in the kernel functions that are near zero everywhere, rendering PCA meaningless. Too large values of μ result in a near-zero reconstruction error for all trajectories, faulty and normal. In Experiment 1, a large range of values of μ, between 0.6 and 600, was found to yield similar performance. While large, the acceptable range of values of μ depends, in ways that are not well-understood, on density and number of trajectories in the training data. Selection of μ can be done using trial and error given a set of trajectories that are known to

be faulty. The number of eigenvectors, N, needs to be selected large enough to ensure that the reconstruction errors are near zero when evaluated at trajectories in the training data.

The results in Table 2 strongly indicate that larger data sets can result in fewer false positives when fault detection is performed using OKPCA. While the false negative rate is small, it shows no such trend. It should be noted that the errors in Table 2 are computed with the threshold in each trial selected as $\varepsilon = 2\max_i\{R(\gamma_i)\}_{i=1}^{M}$. The fact that the false positive rate drops to zero when a larger training data set is used implies that as the training data set gets larger the threshold ϵ could potentially be selected to be smaller. The authors hypothesize that with a more judicious selection of the threshold, the decreasing trend in false positive rates, observed in Table 2, can also be realized in the false negative rates, up to a limit, as the training data set gets larger. It should be noted, however, that OKPCA fault detection scales cubically in M, and as such, the use of large training data sets requires a significant amount of computational resources.

CHAPTER IV

CONCLUSION

In the second chapter, a novel operator representation of a control-affine nonlinear system is developed as a composition of a multiplication operator and a composition-like kernel propagation operator over an RKHS. The multiplication operator takes advantage of the affine nature of the system to capture the effect of control on the system behavior, while the kernel-propagation operator captures the effect of the system dynamics on the kernels of the underlying RKHS. The resulting DMD algorithm is entirely data driven and requires no model knowledge besides the dynamical system being affine in control. Furthermore, the DCLDMD formulation provides a novel way to separate the state from the control input on the operator-theoretic level. This separation leads to better prediction capabilities over existing methods, as evidenced by the results of Experiment 2. Moreover, since DCLDMD can be used to predict closed-loop trajectories of a nonlinear system under feedback laws, it could potentially be utilized for control synthesis, which is a topic for future research.

In the third chapter, the kernel PCA method is generalized to kernelized covariance operators on reproducing Kernel Hilbert spaces. The resulting OKPCA method generates principal components of a set of *trajectories* as opposed to a set of points. It is shown that when occupation kernels are used as feature maps, the computations involved reduce to computation of single and double integrals of kernel functions along the trajectories in the training data and the test data. The developed OKPCA method is applied to the data-driven fault detection problem to separate normal trajectories of a dynamical system from faulty ones, without any knowledge of the system dynamics. Two numerical experiments demonstrate the efficacy of the developed technique. The numerical experiments indicate that provided a

training data set of known normal trajectories and a test data set of known faulty trajectories is available, the parameters of the developed OKPCA fault detection method can be selected by trial and error from a wide range of acceptable values. Performance improvement with increasing amount of training data is also observed, albeit accompanied by a significant rise in computational costs. The numerical experiments also indicate an inherent robustness to noise. A theoretical analysis of noise-robustness is out of the scope of this thesis, and a part of future research.

REFERENCES

[1] Randal Beard, *Quadrotor dynamics and control rev 0.1*, All Faculty Publications 1325, Brigham Young University, 2008.

[2] Claudio Carmeli, Ernesto De Vito, Alessandro Toigo, and Veronica Umanitá., *Vector valued reproducing kernel Hilbert spaces and universality*, Anal. Appl. **08** (2010), no. 01, 19–61.

[3] Sang Wook Choi, Changkyu Lee, Jong-Min Lee, Jin Hyun Park, and In-Beum Lee, *Fault detection and identification of nonlinear processes based on kernel PCA*, Chemom. Intell. Lab. Syst. **75** (2005), no. 1, 55–67.

[4] Zohreh Fathi, W. Fred Ramirez, and Jozef Korbicz, *Analytical and knowledge-based redundancy for fault diagnosis in process plants*, AIChE J. **39** (1993), no. 1, 42–56.

[5] Efrain Gonzalez, Moad Abudia, Michael Jury, Rushikesh Kamalapurkar, and Joel A. Rosenfeld, *The kernel perspective on dynamic mode decompositions*, arXiv:2106.00106, 2021.

[6] Debdipta Goswami and Derek A. Paley, *Bilinearization, reachability, and optimal control of control-affine nonlinear systems: A Koopman spectral approach*, IEEE Trans. Autom. Control **67** (2022), no. 6, 2715–2728.

[7] Heiko Hoffmann, *Kernel PCA for novelty detection*, Pattern Recogn. **40** (2007), no. 3, 863–874.

[8] B. Huang, X. Ma, and U. Vaidya, *Feedback stabilization using Koopman operator*, Proc. IEEE Conf. Decis. Control, December 2018, pp. 6434–6439.

[9] R. Lynn Kirlin, *Eigenstructure, the Karhunen Loeve transform, and singular-value decomposition*, Covariance Analysis for Seismic Signal Processing (R. Lynn Kirlin and William J. Done, eds.), Society of Exploration Geophysicists, 1999, pp. 19–34.

[10] Bernard O Koopman, *Hamiltonian systems and transformation in Hilbert space*, Proc. Natl. Acad. Sci. U.S.A. **17** (1931), no. 5, 315.

[11] Milan Korda and Igor Mezić, *On convergence of extended dynamic mode decomposition to the koopman operator*, J. Nonlinear Sci. **28** (2017), no. 2, 687–710.

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

[13] J. Nathan Kutz, Steven L. Brunton, Bingni W. Brunton, and Joshua L. Proctor, *Dynanic mode decomposition - data-driven modeling of complex systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016.

[14] Cong-De Lu, Tai-Yi Zhang, Xing-Zhong Dy, and Can-Ping Li, *A robust kernel PCA algorithm*, Proc, Int. Conf. Mach. Learn. Cybern., vol. 5, 2004, pp. 3084–3087.

[15] Majdi Mansouri, Mohamed Nounou, Hazem Nounou, and Nazmul Karim, *Kernel PCA-based GLRT for nonlinear fault detection of chemical processes*, J. Loss Prev. Process Ind. **40** (2016), 334–347.

[16] Igor Mezić, *Analysis of fluid flows via spectral properties of the Koopman operator*, Annu. Rev. Fluid Mech. **45** (2013), 357–378.

[17] Bowen Mu, Xuejiao Yang, and Joseph K. Scott, *Comparison of advanced set-based fault detection methods with classical data-driven and observer-based methods for uncertain nonlinear processes*, Comput. Chem. Eng. **166** (2022), 107975.

[18] Mania Navi, Mohammad Reza Davoodi, and Nader Meskin, *Sensor fault detection and isolation of an industrial gas turbine using partial kernel PCA*, Automatica, vol. 48, Elsevier BV, 2015, pp. 1389–1396.

[19] Viet Ha Nguyen and Jean-Claude Golinval, *Fault detection based on kernel principal component analysis*, Eng. Struct. **32** (2010), no. 11, 3683–3691.

[20] Peter Fogh Odgaard and Jakob Stoustrup, *Gear-box fault detection using time-frequency based methods*, Annu. Rev. Control **40** (2015), 50–58.

[21] Hongyi Pan, Diaa Badawi, Ishaan Bassi, Sule Ozev, and Ahmet Enis Cetin, *Detecting anomaly in chemical sensors via L1-kernel-based principal component analysis*, IEEE Sens. Ltrs. **6** (2022), no. 10, 1–4.

[22] R.J. Patton and J. Chen, *Observer-based fault detection and isolation: Robustness and applications*, Control Eng. Pract. **5** (1997), no. 5, 671–682.

[23] Vern I. Paulsen and Mrinal Raghupathi, *An introduction to the theory of reproducing kernel Hilbert spaces*, vol. 152, Cambridge University Press, 2016.

[24] Gert K. Pedersen, *Analysis now*, Graduate Texts in Mathematics, vol. 118, Springer-Verlag, New York, 1989. MR 971256 (90f:46001)

[25] Sebastian Peitz, Samuel E. Otto, and Clarence W. Rowley, *Data-driven model predictive control using interpolated Koopman generators*, SIAM J. Appl. Dyn. Syst. **19** (2020), no. 3, 2162–2193.

[26] C. Piciarelli, C. Micheloni, and G.L. Foresti, *Trajectory-based anomalous event detection*, IEEE Trans. Circuits Syst. Video Technol. **18** (2008), no. 11, 1544–1554.

[27] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz, *Dynamic mode decomposition with control*, SIAM J. Appl. Dyn. Syst. **15** (2016), no. 1, 142–161.

[28] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz, *Generalizing Koopman theory to allow for inputs and control*, SIAM J. Appl. Dyn. Syst. **17** (2018), no. 1, 909–930.

[29] S. Joe Qin, *Data-driven fault detection and diagnosis for complex industrial processes*, IFAC Proc. Vols. **42** (2009), no. 8, 1115–1125.

[30] Joel A. Rosenfeld and Rushikesh Kamalapurkar, *Dynamic mode decomposition with control Liouville operators*, IFAC-PapersOnLine, vol. 54, July 2021, pp. 707–712.

[31] Joel A. Rosenfeld, Rushikesh Kamalapurkar, Benjamin Russo, and Taylor T. Johnson, *Occupation kernels and densely defined Liouville operators for system identification*, Proc. IEEE Conf. Decis. Control, December 2019, pp. 6455–6460.

[32] Peter J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, J. Fluid Mech. **656** (2010), 5–28.

[33] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller, *Nonlinear component analysis as a kernel eigenvalue problem*, Neural Comput. **10** (1998), no. 5, 1299–1319.

[34] Pranav Sharma, Bowen Huang, Venkatramana Ajjarapu, and Umesh Vaidya, *Data-driven identification and prediction of power system dynamics using linear operators*, Proc. IEEE Power Energy Soc. Gen. Meet., August 2019, pp. 1–5.

[35] Robin Strässer, Julian Berberich, and Frank Allgöwer, *Robust data-driven control for nonlinear systems using the Koopman operator*, arXiv.2304.03519, 2023.

[36] Amit Surana, *Koopman operator based observer synthesis for control-affine nonlinear systems*, Proc. IEEE Conf. Decis. Control, IEEE, 2016, pp. 6492–6499.

[37] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz, *On dynamic mode decomposition: Theory and applications*, J. Comput. Dyn **1** (2014), no. 2, 391–421.

[38] Huangang Wang and Ma Yao, *Fault detection of batch processes based on multivariate functional kernel principal component analysis*, Chemom. Intell. Lab. Syst. **149** (2015), 78–89.

[39] Shui-Hua Wang, Tian-Ming Zhan, Yi Chen, Yin Zhang, Ming Yang, Hui-Min Lu, Hai-Nan Wang, Bin Liu, and Preetha Phillips, *Multiple sclerosis detection based on biorthogonal wavelet transform, RBF kernel principal component analysis, and logistic regression*, IEEE Access **4** (2016), 7567–7576.

[40] Matthew O. Williams, Maziar S. Hemati, Scott T.M. Dawson, Ioannis G. Kevrekidis, and Clarence W. Rowley, *Extending data-driven koopman analysis to actuated systems*, IFAC-PapersOnLine **49** (2016), no. 18, 704–709.

VITA

Zachary Daniel Morrison

Candidate for the Degree of

Master of Science

Thesis: KERNEL METHODS FOR SYSTEM IDENTIFICATION AND FAULT DETECTION IN NONLINEAR SYSTEMS

Major Field: Mechanical and Aerospace Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Mechanical and Aerospace Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2024.

Completed the requirements for the Bachelor of Science in Mathematics at Oklahoma State University, Stillwater, Oklahoma in December, 2023.

Completed the requirements for the Bachelor of Science in Aerospace Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2020.

Professional Membership:

Tau Beta Pi