Eigenfunction construction for the Koopman operator

Application of subspace methods and the construction of derived bilinear models

M.T.P. van Laarhoven

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Eigenfunction construction for the Koopman operator

Application of subspace methods and the construction of derived bilinear models

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

M.T.P. van Laarhoven

August 31, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Eigenfunction construction for the Koopman operator

by

M.T.P. van Laarhoven

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: August 31, 2023

Supervisor(s):

Prof.dr.ir. M.H.G. Verhaegen

Reader(s):

Dr.ir. M. Jafarian

Prof.dr.ir. A.C. Schouten

Ir. J. Noom

Abstract

Over the past decade, Koopman operator methods received a surge in interest by enabling the application of linear techniques to nonlinear systems. Existing methods often rely on prior data dictionaries and closure of the operator on this dictionary.

We expand on the framework by Korda and Mezić (2020) to construct eigenfunctions directly from data by exploiting the eigenfunction partial differential equation (PDE), guaranteeing closure and eliminating the need for a prior data dictionary. The constructed models are extended to forced systems through the multi-step prediction error of a linear state-space model.

By identifying a relationship between Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) and Dynamic Mode Decomposition (DMD) applied to Hankel matrices, we simplify the original optimisation problem and significantly reducing the required model order. A detailed numerical investigation of both autonomous dynamics and forced dynamics follows. For autonomous dynamics we report a variance accounted for (VAF) up to 90% for a toy model and the Van der Pol oscillator, whilst the original work is unable to reconstruct the underlying dynamics on longer time scales. We were unable to reproduce accurate multi-step predictions under the influence of forcing.

We extend the constructed models by inclusion of monomial terms into the dynamics. This can be interpreted as a linear model with nonlinear output, approximated by a polynomial. Results on the Koopman generator and the inclusion of monomial terms suggest the construction of a bilinear model. A multi-step prediction is formulated, simplified and solved, expanding the predictive capabilities of the model. Whilst the inclusion of monomial terms improved the prediction of autonomous dynamics, the bilinear models failed to converge for the Duffing oscillator and Van der Pol oscillator.

We perform a further study on the constructed eigenfunctions by designing a new neural network architecture, aimed at learning Koopman eigenfunctions. The network architecture accurately recovers the autonomous dynamics of the system. The learned eigenfunctions suggest that the constructed eigenfunctions can be severely limited by the choice of the initial condition set Γ, opening the door for future research.

Table of Contents

Nomenclature

Glossary

DMD	Dynamic Mode Decomposition
eDMD	extended DMD
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
LPV	linear parameter varying
MILP	mixed-integer linear programming
ODE	ordinary differential equation
PDE	partial differential equation
ReLU	Rectified linear unit
SNR	signal to noise ratio
SVD	singular value decomposition
subDMD	subspace DMD
VAF	variance accounted for

Symbols

$A^\intercal$	Matrix transpose	$\mathbb{R}$	Real numbers
A^*	Hermitian conjugate or complex conjugate	$\mathbb{C}$	Complex numbers
		$\mathbb{Z}$	Set of integers
$A^\dagger$	Matrix pseudo-inverse	$\mathbb{N}$	Set of positive integers
$\mathcal{X}$	State space	N	Measurement sequence length
$\mathcal{U}$	Space of inputs		

n	State dimension
m	Output measurement dimension
p	Input dimension
N_g	Lifted state dimensions
N'	Derived state dimension
x	True state of a system
y	Output measurement
u	Forcing input to system
χ	State in eigenfunction coordinates
$x(t)$	State at continuous-time instance t
$x[k]$	State at discrete-time instance k
τ	Time step for discrete systems
F_t	Flow map of a dynamical system
f	Vector field of a dynamical system
h	Output measurement function
g	Lifting function
φ	Koopman eigenfunction
λ	(Koopman) eigenvalue
μ	Koopman generator eigenvalue
(A, B, C)	(Typically) linear state-space matrices
$\sigma(A)$	Spectrum of A
$\overline{A}$	A with the first d rows removed
$\underline{A}$	A with the last d rows removed
$\mathrm{col}(A)$	Column space of A
$\mathrm{span}(X)$	Linear span of elements in X
ς	Multi-index
$C^r(X)$	r-times continuously differentiable functions on X
$D^r(X)$	r-times continuously differentiable diffeomorphisms
$\|\cdot\|$	Norm
$\langle\cdot\|\cdot\rangle$	Inner product
$\mathcal{L}(X)$	Space of linear operators on X
$\mathcal{G}(X)$	Space of observation functions on X
$L_2(X)$	Set of square-integrable functions on X
$C^r(X)$	Set of r-times continuously differentiable functions

Chapter 1

Introduction

The problem of understanding the principles that generate a sequence of data is fundamental in science. In the context of physics, the emphasis is on revealing the conceptual laws responsible for generating this data. Within engineering, the focus shifts toward deriving numerical representations of these generating laws. Numerical methods that learn the dynamics of a given measurement sequence, enabling accurate predictions of these dynamics, are therefore vital tools in engineering.

In the domain of system and control, accurate models facilitate the design of control strategies to achieve some desired behaviour. Typically, such models are constructed using prior knowledge of the dynamics. However, this prior knowledge is not always available, necessitating the derivation of models from observed data. Effective methods for data-driven construction of models for linear systems have been developed, but doing so remains difficult for nonlinear systems. The prevalence of nonlinear systems in the real world, coupled with the large amount of data presently available, makes this an active area of research across numerous scientific domains [2].

In the last decade, data-driven analysis of nonlinear systems has grown into a popular field, following renewed interest in the Koopman operator. The Koopman operator lifts the dynamics of the system to dynamics of *observables*. These advances were initiated with the development of Dynamic Mode Decomposition (DMD) for the analysis of fluid dynamics [3], spawning a range of modifications. A few examples are: extended DMD (eDMD) by Williams et al. [4], eDMD with control by Korda and Mezić [5] and subspace DMD (subDMD) by Takeishi et al. [6]. Applications have emerged in the fields of neuroscience [7], robotics [8, 9], energy efficiency [10], quantum control [11], plasma physics [12] and molecular physics [13].

In the present work, we aim to construct models that can be employed for control of nonlinear systems. Existing methods either propose linear models [1, 5] or linear parameter varying (LPV) models [14]. Linear models have a limit on the nonlinear dynamics they can capture [15]. Work by Surana suggests that bilinear models are better at capturing the dynamics in lifted coordinates [16].

Most Koopman operator methods assume that the derived model is closed on the span of observables [17, 18]. This is a strong assumption to make, and can be avoided by using

eigenfunctions as coordinates, which are closed by definition. Deriving these directly from data is attempted by Kaiser et al. and Korda and Mezić [1, 18].

Korda and Mezić construct the eigenfunctions by exploiting the Koopman eigenfunction partial differential equation (PDE), extending any function on a surface of initial conditions to an eigenfunction of the Koopman operator. This work expands on this method, with developments focused around the underlying research question,

> Can we construct bilinear models with nonlinear output from measurement data using optimal construction of Koopman operator eigenfunction whilst maintaining a low computational complexity?

Chapter 2 builds up the terminology and Koopman operator theory, functioning as a literature review on existing Koopman operator methods. This chapter ends with a detailed summary of the method presented by Korda and Mezić and a problem statement, motivating the aforementioned research question. These are summarised in 6 subproblems.

Chapter 3 presents an alternative approach to solving the optimisation problem presented in the original method, whilst simultaneously reducing the constructed system order. The proposed approach shows a close relationship between DMD and Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [19]. The method is numerically tested and compared to the original results by Korda and Mezić for a Toy model, the Duffing oscillator and Van der Pol oscillator.

Chapter 4 extends the constructed models by approximating a nonlinear output with a polynomial output. Properties of the Koopman are exploited to include these into linear autonomous dynamics. This model is then applied to construct a bilinear model for the forced system. The chapter ends with a numerical investigation of these models.

We compare the constructed eigenfunctions with a eigenfunctions learned by a new neural network architecture in Chapter 5. The learned eigenfunctions point out a clear limitation of eigenfunction construction.

Finally, the work concludes with a discussion and outlook in Chapter 6 and a conclusion in Chapter 7.

Chapter 2

Theory and literature review

In this section, we aim to provide background theory on dynamical systems and the Koopman operator. We do this by introducing terminology for discussion of dynamical systems and for the identification of linear and nonlinear systems. This will follow with a definition for the Koopman operator and relevant properties. We end wtih a summary of the literature study on a range of numerical methods that are often used to construct a Koopman operator model from data. We conclude with a problem statement for the present work.

2-1 Identification of dynamical systems

Identification of dynamical systems is the problem of deriving the equations that generated the dynamics from data. In this section our objective is twofold: to introduce terminology of dynamical systems and define the problem of system identification.

Dynamical systems

Dynamical systems aim to mathematically desribe the evolution of a set of measurements over time. A dynamical system is typically defined by a set of states, written $\mathcal{X}$, and a map characterising the evolution over time, given by F_t. This is made explicit in the following definition.

Definition 2.1 (Dynamical system)**.** A dynamical system is a triplet $(F_t, \mathcal{X}, T)$, consisting of a state set $\mathcal{X}$, a time set T and a map $F_t : \mathcal{X} \to \mathcal{X}$ satisfying $F_0(x) = x$ and $F_{t_1}(F_{t_2}(x)) = F_{t_1+t_2}(x)$ for all $x \in \mathcal{X}$ and $t_1, t_2 \in T$. Furthermore, we call the systems

- *Continuous* if the flow map $F_t(x_0)$ of the dynamical system is the solution at time $t \geq 0$ given a differential equation with boundary condition $x(0) = x_0$. In terms of the vector field f, this becomes

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

- *Discrete* if the flow map $F_t(x_0)$ of the dynamical system is the solution at time $t \geq 0$ given a transition map f with initial condition $x[0] = x_0$

$$x[k+1] = f(x[k]).$$

Moreover, we call a system *forced* if the flow map $F_t(x_0, u(t))$ or $F_t(x_0, u[k])$ is dependent on a time-dependent forcing $u \in \mathcal{U}$, associated to the solution of the differential equation or transition map with boundary condition $x(0) = x_0$,

$$\dot{x}(t) = f(x(t), u(t)), \qquad x[k+1] = f(x[k], u[k])$$

In the present work only *time-independent* systems are considered, for which the vector field or transition map f is independent of the time variable $t \in T$. Describing the dynamical system through the map f is often more useful. The vector field often has a much simpler expression than the flow map F_t, and can often be derived from physics. We assume that this vector field f is Lipschitz continuous, required for the uniqueness of the map F_t. That is, for states $x, y \in \mathcal{X}$ we require

$$\|f(x) - f(y)\| \leq C\|x - y\|$$

for some $C > 0$. In particular, this often culminates in the considered state set $\mathcal{X}$ being bounded.

For forced systems, the map f is often decomposed into components for classification based on mathematical properties. The terminology for the present work is defined in Table 2-1. In the introduction of these models, the time dependence (t) of the state variable x is dropped for compactness. All definitions extend to discrete-time be replacing $\dot{x}$ with $x[k+1]$ and x with $x[k]$. It will be made clear in context whether a continuous-time or discrete-time model is considered.

Table 2-1: Model types considered in the present work for a state variable $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and $u \in \mathbb{R}^p$. We denote for the matrices $A \in \mathbb{R}^{n\times n}$, $B_0 \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{m\times n}$ and $B_i \in \mathbb{R}^{n\times n}$ for $i = 1, 2, \ldots, p$. Furthermore, we use the notation $f : \mathbb{R}^n \to \mathbb{R}^n$, $h : \mathbb{R}^n \to \mathbb{R}^m$ and $g_i : \mathbb{R}^n \to \mathbb{R}^n$ for $i = 1, 2, \ldots, p$ for the functions used in these models.

Name	Evolution map	Output map
Linear	$\dot{x} = Ax + B_0 u$	$y = Cx$
Bilinear	$\dot{x} = Ax + B_0 u + \sum_{i=1}^{k} u_i B_i x$	$y = Cx$
Linear in control	$\dot{x} = f(x) + B_0 u$	$y = h(x)$
Control-affine	$\dot{x} = f(x) + \sum_{i=1}^{p} g_i(x) u_i$	$y = h(x)$

In the present work we will discuss the systems introduced in Table 2-1, and aim to identify models of these systems from data. This data is often corrupted by some measurement noise due to uncertainties in the sensors and unmodeled high-frequency dynamics. We will consider measurements corrupted by white noise $\eta \sim \mathcal{N}(0, \sigma^2)$. The output map in Table 2-1 is then replaced by

$$y = Cx + \eta, \qquad \text{or} \qquad y = h(x) + \eta.$$

We will now turn to the system identification problem.

The system identification problem

The system identification problem boils down to an optimisation problem in terms of a set of measurements and a parameterised target model. The target model will be denoted M, with parameters given by θ, as an element of all, possibly constrained, parameters Θ. The target model aims to estimate the k'th future measurement from an initial given measurement and the inputs given until the k'th time step. The parameterisation of the model is often motivated by the goals that the model should achieve. For example, if linear control methods are to be employed, the model can parameterised as a transfer function or by the matrix entries of the linear model in Table 2-1.

Definition 2.2 (System identification problem)**.** Given measurement $y_{(i)}[k]$ and input sequences $u_{(i)}[k]$, with $k = 0, 1, 2, \ldots, N$, indexed by $i \in \mathcal{I}$. The identification problem aims to find the parameters of a parameterised target model $M : \Theta \times \mathbb{R}^m \times \mathcal{U} \times \mathbb{N} \to \mathbb{R}^m$ through the optimisation problem

$$\underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i \in \mathcal{I}} \sum_{k=1}^{N} \left\| y_{(i)}[k] - M(\theta, y_{(i)}[0], u_{(i)}, k) \right\|_2$$

In the present work we assume that we have control over the measurement sequences $y_{(i)}$ and input sequences $u_{(i)}$, making experiment design part of the identification problem.

Verification of the model on new data is a key component for understanding the validity of the model [20]. For this purpose we will consider the variance accounted for (VAF) as a performance measure for identified models, defined in terms of a validation sequence $y[k]$ and an estimated sequence $\hat{y}[k]$ of length N,

$$\text{VAF}(y, \hat{y}) = 100\% \times \left(1 - \frac{\sum_{k=0}^{N} (y[k] - \hat{y}[k])^2}{\sum_{k=0}^{N} (y[k])^2} \right). \tag{2-1}$$

For linear systems one typically performs correlation analysis to gauge the performance of model. If the model accurately captures the dynamics in a system, the residual $\epsilon = |\hat{y}[k] - y[k]|$ should be uncorrelated with itself. For systems with nonlinear components, it is often not feasible to achieve uncorrelated residuals [15], since the uncaptured dynamics have will culminate in residuals correlated to the state. Instead, a frequentist perspective for the residuals is often taken to analyse the performance of a nonlinear model in comparison to the best possible linear approximation.

The system identification process is the combination of model selection, experiment design and validation, which we summarise by the procedure presented in Figure 2-1.

Takens embedding theorem

A primary contributor to the complexity of system identification is that measurements y are results of the composition of the evolution functions f and output functions h of the system

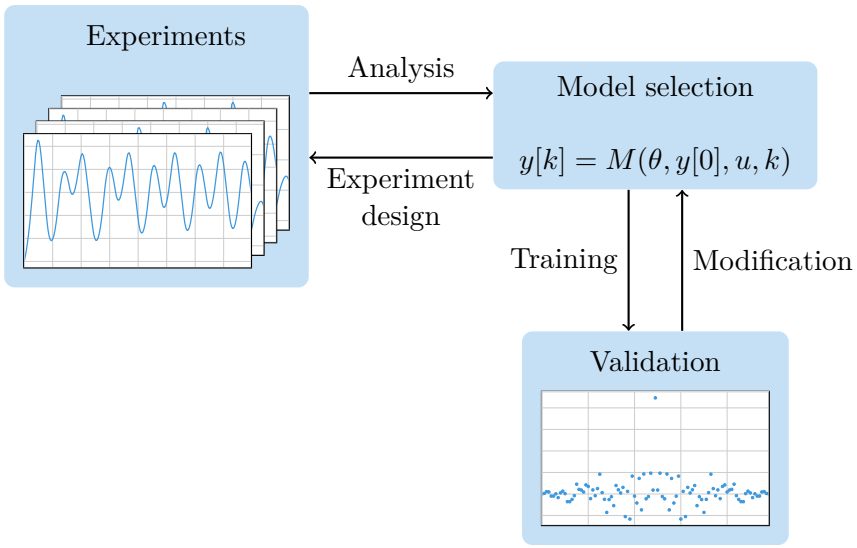

Figure 2-1: The system identification cycle, illustrating the use of data analysis for model selection and experiment design, and then validating the constructed models.

models defined in Table 2-1. Consequently, the state dimension n could be smaller than the measurement dimension m, limiting the information contained in the output measurements. In 1981, Takens proved the existence of a map between time-delayed measurements and the state dynamics of the system [21]. The importance of this embedding theorem was noted by the field of system identification as it moved toward nonlinear identification methods [22, 23]. In this section we will state a variation of the embedding theorem, and state the consequence of this theorem in the context of system identification.

This formulation is due to Stark [22], where we use f^p to denote p repeated applications of the evolution function in a discrete setting. We use $D^r(\mathcal{X})$ to denote the set of r-times differentiable diffeomorphic functions on $\mathcal{X}$ and $C^r(\mathcal{X}, \mathbb{R})$ to denote the r-times continuously differentiable maps from $\mathcal{X}$ into $\mathbb{R}$.

Theorem 2.1 (Takens embedding theorem [22])**.** *Let $\mathcal{X}$ be a compact n-dimensional manifold. If $d \geq 2n+1$, the set (f, h) for which the delay embedding map $\phi_{f,h} : \mathcal{X} \to \mathbb{R}^d$ is an embedding, defined by*

$$\phi_{f,h}(x) \coloneqq \begin{bmatrix} h(n) & h(f(n)) & h(f^2(n)) & \cdots & h(f^{d-1}(n)) \end{bmatrix},$$

is open and dense in $D^r(\mathcal{X}) \times C^r(\mathcal{X}, \mathbb{R})$ for $r \geq 1$.

Up to this point we kept $\mathcal{X}$ undefined for generality. In practice $\mathcal{X}$ is some subset of $\mathbb{R}^n$, the conditions of compactness and $\mathcal{X}$ to be a manifold result in some restrictions on the dynamics of the system. First, for finite-dimensional metric spaces with the induced topology, compactness of a set is equivalent to being closed and bounded. It follows that the dynamics f cannot diverge to remain properly defined. Second, $\mathcal{X}$ is a manifold as a subspace of $\mathbb{R}^n$ when the function representing the manifold does not have any singular points, this condition is almost always met [24].

The density in $D^r(\mathcal{X}) \times C^r(\mathcal{X}, \mathbb{R})$ means that $\phi_{f,h}$ is not necessarily an embedding for all pairs f, h. However, an embedding that is arbitrarily close to $\phi_{f,h}$ can be constructed such that

for $\epsilon > 0$, there exist $f' \in D^r(\mathcal{X})$ and $h' \in C^r(\mathcal{X}, \mathbb{R})$ such that $\|f - f'\|_\infty < \epsilon$, $\|h - h'\|_\infty < \epsilon$ and $\phi_{f',h'}$ is an embedding.

A function ϕ is an embedding if it maps $\mathcal{X}$ diffeomorphically onto its image. On a compact manifold this is equivalent to both ϕ and its derivative being injective on $\mathcal{X}$. Hence, one can define $f' \coloneqq \phi \circ f \circ \phi^{-1}$ as the evolution map in time-delay coordinates. Because ϕ is an embedding, this is well-defined with dynamics that are equivalent to the original system.

The work of Stark extends Theorem 2.1 to forced systems, both with known and unknown input signal dynamics. In particular, this yields an embedding of the form

$$\phi_{f,h,u}(x) \coloneqq \begin{bmatrix} h(x[k]) & h(x[k-1]) & h(x[k-2]) & \cdots & h(x[k-d+1]) \end{bmatrix}. \tag{2-2}$$

For specific inputs, the above delay embedding does not satisfy the density result in Theorem 2.1. However, it does for a dense subset of the possible inputs $u \in \mathcal{U}$. This yields equivalent dynammics in delay coordinates,

$$z[k+1] = \phi_{f,h,u}\left(f(\phi^{-1}_{f,h,u}(z[k]), u[k])\right).$$

We summarise this in Corollary 2.1

Corrollary 2.1 (Takens embedding for input-output systems)**.** Given a discrete-time input-output system

$$\begin{aligned} x[k+1] &= f(x[k], u[k]), \\ y[k] &= h(x[k]). \end{aligned}$$

This yields an associated system in time-delay coordinates for almost all $u[k] \in \mathcal{U}$,

$$\begin{aligned} z[k+1] &= f'(z[k], u[k]) \\ z[k] &= \phi(x[k]) \end{aligned}$$

Where ϕ satisfies the conditions is defined as in Equation (2-2).

In the context of system identification this implies that if we have a some convergent algorithm estimating the dynamic with time-delay coordinates, that this limit will have dynamics equivalent to the underlying system. We work out the delay embedding dynamics for a linear system in the following example.

Example 2-1.1 (Linear system)**.** Consider a linear system of the form

$$\begin{aligned} x[k+1] &= Ax[k] + Bu, \\ y[k] &= Cx[k]. \end{aligned}$$

The embedding ϕ becomes

$$\phi(x[k]) = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{d-1} \end{bmatrix} x[k] + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{d-2}B & CA^{d-3}B & \cdots & CB \end{bmatrix}}_{\mathcal{K}_B} \begin{bmatrix} u[k] \\ u[k+1] \\ \vdots \\ u[k+d-2] \end{bmatrix}.$$

Then, to find ϕ^{-1} we aim to find a solution for $x[k]$ in

$$z[k] - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{d-2}B & CA^{d-3}B & \cdots & CB \end{bmatrix} \begin{bmatrix} u[k] \\ u[k+1] \\ \vdots \\ u[k+d-2] \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{d-1} \end{bmatrix}}_{\mathcal{O}_d} x[k].$$

This has a solution for $x[k]$ if the matrix $\mathcal{O}_d$ has a left-inverse. This left-inverse exists as the pseudo-inverse $\mathcal{O}_d^\dagger$ if $\mathcal{O}_d$ has full column rank, which is the case if the system is observable and $d \geq n$.

For our example, take $d = n$ for $n = 3$. Then,

$$x[k+1] = A\mathcal{O}_d^\dagger z[k] - A\mathcal{O}_d^\dagger \begin{bmatrix} 0 & 0 & 0 \\ CB & 0 & 0 \\ CAB & CB & 0 \end{bmatrix} \begin{bmatrix} u[k] \\ u[k+1] \\ u[k+2] \end{bmatrix} + \begin{bmatrix} B & 0 & 0 \end{bmatrix} \begin{bmatrix} u[k] \\ u[k+1] \\ u[k+2] \end{bmatrix}.$$

We define $A' = \mathcal{O}_d A \mathcal{O}_d^\dagger$, to find

$$z[k+1] = A'z[k] - A' \begin{bmatrix} 0 & 0 & 0 \\ CB & 0 & 0 \\ CAB & CB & 0 \end{bmatrix} \begin{bmatrix} u[k] \\ u[k+1] \\ u[k+2] \end{bmatrix} + \begin{bmatrix} CB & 0 & 0 \\ CAB & CB & 0 \\ CA^2B & CAB & CB \end{bmatrix} \begin{bmatrix} u[k] \\ u[k+1] \\ u[k+2] \end{bmatrix}.$$

We conclude that a linear system yields a linear system when viewed in time-delay coordinates.

2-2 Example systems

In this work we will consider three systems for which we illustrate the presented concepts and test the developed numerical methods. Each of these models serves their own purpose and poses specific difficulties for the developed methods. The vector fields are presented in Figure 2-2.

Toy model

This Toy model was originally proposed by Tu et al., serving an illustrative role in the context of Koopman operator methods [17]. The vector field of this model is given by,

$$\begin{aligned} \dot{x}_1 &= \nu x_1 + u \\ \dot{x}_2 &= \sigma(x_2 - x_1^2). \end{aligned}$$

This model is relatively simple, and has dynamics reminiscent to those of linear systems. The vector field is presented in Figure 2-2, illustrating the attractive single point for $\nu = -0.4$ and $\sigma = -0.5$.

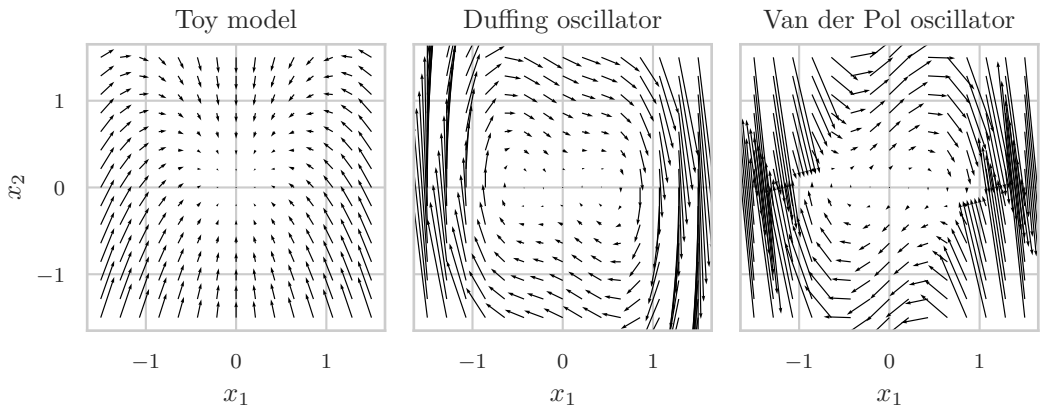

Figure 2-2: Vector field of the Toy model, Duffing oscillator and Van der Pol oscillator.

Duffing oscillator

The Duffing oscillator can be interpreted as a mass-damper system with a nonlinear spring [24]. The model is often used as an example of a symple system that expresses chaos. This is the format presented by Korda and Mezić [1].

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -0.5x_2 - x_1(4x_1^2 - 1) + u\end{aligned}$$

This vector field has three fixed points located at $(x_1, x_2) = (-0.5, 0)$, $(x_1, x_2) = (0.5, 0)$ and at $(0, 0)$. The fixed point at $(x_1, x_2) = (0, 0)$ is unstable, while the others are stable. The two stable fixed points introduce interesting dynamics into the system, since the attractive regions of the two fixed points cause the dynamics of close initial conditions to diverge towards the two different fixed points.

Van der Pol oscillator

The Van der Pol oscillator is a nonlinear system originally investigated in the context of electronics [25]. We use the format presented by Korda and Mezić [1].

$$\begin{aligned}\dot{x}_1 &= 2x_2\\ \dot{x}_2 &= -0.8x_1 + 2x_2 - 10x_1^2x_2 + u\end{aligned}$$

This model has a single unstable fixed point at the origin. This model poses two difficulties in the context of identification: For one, the system has a periodic orbit, introducing difficulties into identification problems, as any error in the initial state estimate at $t = 0$ will not disappear over time. Secondly, the vector field for states $\|x\| > 1$ is much larger than for states $\|x\| < 1$, putting requirements on the step size for both regions.

2-3 Koopman Operator

In this section we aim to define the Koopman operator. For this, we first define a few key terms from functional analysis, providing a mathematical framework for the Koopman operator.

Linear Operators

Linear operators aim to generalise the properties of matrices to infinite-dimensional Banach spaces. Banach spaces are closed spaces with a distance measure, the metric. We set the stage by giving a few background definitions [26].

Definition 2.3 (Linear Operator [26])**.** A linear operator A from Banach Spaces M to N is a pair $(A, D(A))$, with the domain $D(A)$ a subspace of M and $A : D(A) \to N$ a bounded linear operator. That is, for $x, y \in M$ and $\alpha, \beta \in \mathbb{C}$, we have

$$A(\alpha x + \beta y) = \alpha Ax + \beta Ay.$$

Furthermore, we have $\|Ax\| \leq c\|x\|$ for some $c \geq 0$ and all $x \in D(A)$. The set of linear operators on M to N is denoted $\mathcal{L}(M, N)$.

One of the main features of matrices is the information contained in eigenvalues and eigenvectors. One can extend their definition to the realm of infinite-dimensional operators. However, the determinant does not lend itself for the extension to infinite dimensions. Instead, we consider that for a matrix A, an eigenvalue λ makes $A - \lambda I$ not invertible. This is the definition that is generalised for linear operators.

Definition 2.4 (Resolvent and spectrum [26])**.** The resolvent set $\rho(A)$ of a linear operator A on X is the set of λ for which $\lambda I - A$ has a bounded two-sided inverse,

$$\rho(A) \coloneqq \{\lambda \in \mathbb{C} : \exists U \in \mathcal{L}(X) \text{ with } U(\lambda I - A)x = x = (\lambda I - A)Ux\}.$$

The spectrum is the complement of the resolvent

$$\sigma(A) \coloneqq \mathbb{C} \backslash \rho(A).$$

The spectrum of linear operators is significantly more intricate than those of matrices. A range of classification methods exist for λ in the context of dynamical systems. However, this is beyond the scope of the present work.

The Koopman operator really is an element of a family of operators, called a C_0-Semigroup. A C_0-Semigroup is effectively a generalisation of the exponential function. The properties of the exponential function extend themselves directly to a definition for the C_0-Semigroup.

Definition 2.5 (C_0-Semigroup [26])**.** A family $S = \{S_t\}_{t \geq 0}$ of bounded operators on M is a C_0-semigroup if

1. $S_0 = I$

2. $S_t S_s = S_{t+s}$ for all $t, s \geq 0$
3. $\lim_{t \downarrow 0} \|S_t x - x\| = 0$ for all $x \in M$

The generator of the C_0-semigroup is the operator L defined

$$D(L) \coloneqq \{x \in M : \lim_{t \downarrow 0} \frac{1}{t}(S_t x - x) \in M\}$$
$$Lx \coloneqq \lim_{t \downarrow 0} \frac{1}{t}(S_t x - x)$$

The relationship between the definition of the C_0-Semigroup and the exponential function will become more clear once the Koopman operator is introduced.

Spectral decomposition

A core result in linear systems is the spectral decomposition theorem, allowing a matrix to be defined in terms of the action on its eigenvectors. This decomposition theorem can be extended to linear operators, with a few modifications. For a normal $n \times n$ matrix A with eigenvalues λ_i, the spectral decomposition reads

$$A = \sum_{i=1}^{n} \lambda_i P_{\lambda_i},$$

with P_{λ_i} the projection matrix onto the span of the eigenvector associated to λ_i [26].

For normal linear operators a similar result can be stated in terms of the action on x

$$Ax = \sum_{i \geq 0} \underbrace{\lambda_i P_{\lambda_i} x}_{\text{atomic}} + \underbrace{\int \lambda \, \mathrm{d}\left(E_c(\lambda) x\right)}_{\text{continuous}}.$$

The integral term emerges due to the infinite-dimensionality of the operator. A detailed discussion of the spectral properties of linear operators is beyond the scope of the present work.

The Koopman Operator

The core idea of the Koopman operator is to take a nonlinear, continuous finite-dimensional system and trade in the nonlinearity of the system for infinite-dimensional linearity.

Definition 2.6 (Koopman operator on $\mathcal{G}(\mathcal{X})$)**.** Given an autonomous continuous-time dynamical system $(F_t, \mathcal{X}, T)$, the associated Koopman operator $K_t : \mathcal{G}(\mathcal{X}) \to \mathcal{G}(\mathcal{X})$ is the C_0-semigroup defined by

$$\mathcal{K}_t g(x) \coloneqq g(F_t(x)).$$

Effectively, the Koopman operator takes an input function g and propagates it forward in time using the flow map F_t. The flow map typically is the solution of an associated differential

equation $\dot{x} = f(x(t))$, given an initial condition. Hence, the benefit of the Koopman operator is not directly clear. The benefits of the operator lie in the linearity of the operator,

$$\mathcal{K}_t(\alpha g(x) + \beta h(x)) = \alpha \mathcal{K}_t g(x) + \beta \mathcal{K}_t h(x).$$

By projecting the infinite-dimensional operator to a finite-dimensional subspace, the operator becomes a matrix. This allows for numerical methods to be implemented for nonlinear systems on this finite-dimensional projection. For example, the linearity of the operator allows for spectral analysis of the system. And, by extension, imposes structure in the identification for linear models of nonlinear systems. The time propagation is summarised in the following schematic,

$$\begin{array}{ccc} x(t) & \xrightarrow{F_s} & x(t+s) \\ g \downarrow & & \downarrow g \\ g(x(t)) & \xrightarrow[K_s]{} & g(x(t+s)) \end{array}$$

Instead of first propagating the state in time, and then taking applying the observation functino, the observation function is applied, and then propagated in time afterwards. Ul-

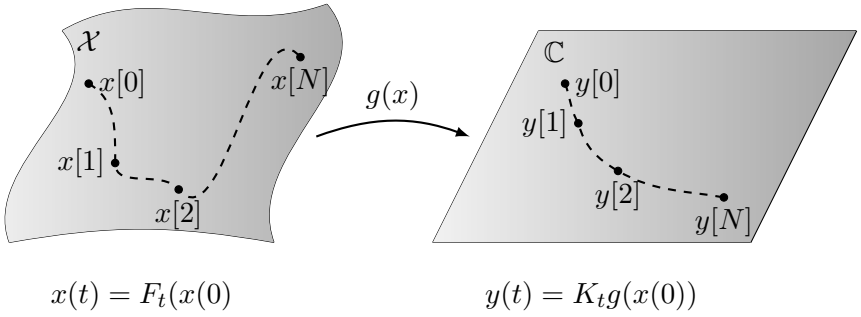

Figure 2-3: The Koopman operator lifts the nonlinear dynamics in a state space $\mathcal{X}$ to an infinite-dimensional $\mathcal{G}(\mathcal{X})$, with linear dynamics. The pointwise values in $\mathbb{C}$ propagate linearly in time.

timately the Koopman operator transforms the dynamics of a finite-dimensional nonlinear system with $x \in \mathcal{X}$ to an infinite-dimensional linear system with $\chi \in \mathcal{G}(\mathcal{X})$. Hence, we transform the nonlinear model into a linear model,

$$\dot{x} = f(x) \qquad \leftrightarrow \qquad \chi[k+1] = \mathcal{K}\chi[k].$$

Where the dynamics at a state x can be analysed by the map $x \mapsto \chi(x)$.

Properties

The primary reason for the investigation of the Koopman operator is the discussion of the associated spectral properties. We will show that these have a close analogue to linear systems

through the generator of the Koopman operator. An eigenfunction is any function $\varphi \in \mathcal{G}(\mathcal{X})$ satisfying

$$\mathcal{K}_\tau \varphi = \lambda^\tau \varphi.$$

These eigenvalues satisfy an important algebraic property. Note that the time dependence of the eigenvalue is solely determined by the exponent τ, we will motivate this later in this section.

Proposition 2.1 (Koopman operator eigenvalues)**.** Suppose $\varphi_1, \varphi_2 \in \mathcal{G}(\mathcal{X})$ are nonzero eigenfunctions of the Koopman operator with eigenvalues λ_1 and λ_2 respectively. Suppose $\varphi_1^n, \varphi_2^m \in \mathcal{G}(\mathcal{X})$, then, $\varphi_1^n \varphi_2^m$ is also an eigenfunction with eigenvalue $\lambda_1^n \lambda_2^m$ for $n, m \in \mathbb{Z}$.

In addition, λ_1^* is also an eigenvalue with eigenfunction φ_1^*.

Proof. Let $\varphi_1, \varphi_2 \in \mathcal{G}(\mathcal{X})$ be nonzero eigenfunctions of the Koopman operator $\mathcal{K}_\tau$, with eigenvalues λ_1 and λ_2 respectively. Let $n, m \in \mathbb{Z}$ such that $\varphi_1^n, \varphi_2^m \in \mathcal{G}(\mathcal{X})$. Then,

$$\mathcal{K}_\tau \varphi_1^n \varphi_2^m = (\varphi_1^n \varphi_2^m) \circ F_\tau = (\varphi_1 \circ F_\tau)^n (\varphi_2 \circ F_\tau)^m = \lambda_1^n \lambda_2^m \varphi_1^n \varphi_2^m.$$

We conclude that $\varphi_1^n \varphi_2^m$ is also an eigenfunction with eigenvalue $\lambda_1^n \lambda_2^m$.

The second property can be derived from

$$\mathcal{K}_\tau \varphi_1^* = \varphi_1^* \circ F_\tau = (\varphi_1 \circ F_\tau)^* = \lambda_1^* \varphi_1^*.$$

From which we conclude that φ_1^* is an eigenfunction with eigenvalue λ_1^*. □

The above proof works for $n, m \in \mathbb{R}$ as well. However, in the context of complex-valued functions an appropriate branch cut should be chosen. In our work we only need this property for integer multiples. The assumption that $\varphi_1^n \varphi_2^m \in \mathcal{G}(\mathcal{X})$ is required to ensure that the eigenfunction actually exists in the function space. For example, continuous functions do not necessarily have a continuous inverse, relating to the $n = -1$ in Proposition 2.1.

The generator of the Koopman C_0-Semigroup is, by definition,

$$Lg \coloneqq \lim_{\tau \to 0} \frac{g \circ F_\tau - g}{\tau} = \lim_{\tau \to 0} \frac{g(x(t+\tau)) - g(x(t))}{\tau} = \frac{\mathrm{d}}{\mathrm{d}t} g(x(t)). \tag{2-3}$$

Here we consider $x(t)$ as a given trajectory, satisfying the differential equation $\dot{x} = f(x)$. We can compute the derivative further,

$$\frac{\mathrm{d}}{\mathrm{d}t} g(x(t)) = \nabla g \cdot \dot{x}(t) = \nabla g \cdot f(x). \tag{2-4}$$

It is often noted that this is the Lie derivative of g along the vector field f [16, 27]. This lends itself to continuous-time eigenfunctions of the generator,

$$L\varphi(x) = \frac{\mathrm{d}}{\mathrm{d}t} \varphi = \mu \varphi(x) \tag{2-5}$$

In particular, any eigenfunction φ of the Koopman operator is also an eigenfunction of the generator,

$$L\varphi = \lim_{\tau \to 0} \frac{K_\tau \varphi - \varphi}{\tau} = \lim_{\tau \to 0} \frac{\lambda^\tau - 1}{\tau} \varphi = \log(\lambda) \varphi.$$

For an eigenfunction φ we conclude that discrete-time eigenvalues λ^τ and continuous-time eigenvalues μ of a system are related by $\lambda = e^\mu$. This is the same relationship satisfied by eigenvalues of a discretisation of a continuous-time linear system. We often work with both μ and λ, throughout this work the notation will be kept consistent, with μ referring to continuous-time eigenvalues and λ referring to discrete-time eigenvalues respectively.

In particular, we emphasise that Equation (2-5) leads to a useful insight. If a boundary condition for the eigenfunction at time $t = 0$ can be given, the differential equation can be solved by propagating the initial condition forward in time. That is, given $\varphi(x_0)$ at $t = 0$, we have that

$$e^{\mu t}\varphi(x_0) = \lambda^t \varphi(x_0)$$

is a solution of the differential equation.

The linear properties of the Koopman operator allow for a spectral decomposition of the operator. That is, given a set of eigenfunctions

$$\Phi = \{\varphi \in \mathcal{G}(\mathcal{X}) : \mathcal{K}_\tau \varphi = \lambda^\tau \varphi\},$$

the action of the Koopman operator on an observation function g can be computed by projection onto the eigenfunctions as

$$\mathcal{K}_\tau g = \sum_{k\geq 1} \lambda^\tau \varphi_i P_{\varphi_i} g = \sum_{k\geq 1} \lambda^\tau \varphi_i \langle \varphi_i | g \rangle. \tag{2-6}$$

Or, equivalently in continuous-time,

$$Lg = \sum_{k\geq 1} \mu \varphi_i P_{\varphi_i} g = \sum_{k\geq 1} \mu \varphi_i \langle \varphi_i | g \rangle$$

Examples

To illustrate the properties of the Koopman operator we investigate the eigenfunctions of two systems. In particular, we investigate the Toy model and Van der Pol oscillator, introduced in Section 2-2. For these systems, the partial differential equation (PDE) in Equation (2-5) can be solved by the method of characteristics [28]. It is rarely mentioned by authors that this is possible. This raises the question whether insights can be recovered from comparing the results of numerical methods for the Koopman operator with these analytical expressions.

Example 2-3.1 (Toy model)**.** We solve the problem for the generalised form of the Toy model presented in Section 2-2. Given the differential equations,

$$\begin{aligned} \dot{x}_1 &= \nu x_1, \\ \dot{x}_2 &= \sigma(x_2 - x_1^2), \end{aligned}$$

We can solve the eigenfunction PDE in Equation (2-5) analytically,

$$\begin{aligned} \mu\varphi &= \langle \nabla\varphi | f \rangle \\ &= \nu x_1 \frac{\partial \varphi}{\partial x_1} + \sigma(x_2 - x_1^2)\frac{\partial \varphi}{\partial x_2} \end{aligned}$$

This equation can be solved by the method of characteristics. This yields the following set of ordinary differential equations (ODEs),

$$\begin{aligned} \frac{\mathrm{d}x_2}{\mathrm{d}x_1} &= \frac{\sigma(x_2 - x_1^2)}{\nu x_1}, & x_2(x_1) &= c_1 x_1^{\sigma/\nu} + \frac{\sigma}{\sigma - 2\nu} x_1^2, \\ \frac{\mathrm{d}\varphi}{\mathrm{d}x_1} &= \frac{\mu\varphi}{\nu x_1}, & \varphi(x_1) &= c_2 x_1^{\mu/\nu}. \end{aligned}$$

Along the curve $x_2(x_1)$ we have $\varphi(x_1) = c_2 x_1^{\mu/\nu}$. Now, c_2 can be a function of c_1, so that

$$c_2 = h(c_1) = h\left(\frac{x_2 - \frac{\sigma}{\sigma - 2\nu} x_1^2}{x_1^{\sigma/\nu}}\right).$$

This yields the general solution for φ, given an arbitrary eigenvalue μ,

$$\varphi_\mu = x_1^{\mu/\nu} h\left(\frac{x_2 - \frac{\sigma}{\sigma - 2\nu} x_1^2}{x_1^{\sigma/\nu}}\right)$$

Two simple eigenfunctions can be derived from this expression.

$$\begin{aligned} \varphi_1 &= x_1, & &\text{with} \quad \mu_1 = \nu \text{ and } h(s) = 1, \\ \varphi_2 &= x_2 - \frac{\sigma}{\sigma - 2\nu} x_1^2, & &\text{with} \quad \mu_2 = \sigma \text{ and } h(s) = s. \end{aligned}$$

These eigenfunctions are desirable in the approximation of the underlying dynamics, since on domains with $|x_1| < 1$, the original states x_1 and x_2 are approximated relatively well by these two eigenfunctions.

Proposition 2.1 raised the question whether there exists a finite set of eigenfunctions that can be used to construct all other eigenfunctions. The previous example serves as a counterexample to this question. In fact, the entire construction put no requirements on μ and h, making the set of eigenvalues and eigenfunctions uncountably large.

Example 2-3.2 (Van der Pol oscillator)**.** The general form of the Van der Pol oscillator is typically given in terms of a single parameter ν [25],

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \nu(1 - x_1^2)x_2 - x_1 \end{aligned}$$

Applying the method of characteristics, we find the following set of ODEs,

$$\begin{aligned} \frac{\mathrm{d}x_2}{\mathrm{d}x_1} &= \nu(1 - x_1^2) - \frac{x_1}{x_2}, & x_2(x_1) &= -\frac{\nu}{4} x_1^3 + \frac{\nu}{2} x_1 + \frac{c_1}{x_1}, \\ \frac{\mathrm{d}\varphi}{\mathrm{d}x_1} &= \frac{\mu\varphi}{x_2}, & \varphi(x_1) &= c_2 e^{\mu x_1}. \end{aligned}$$

Solving for the constant c_2 along the curve of x_2 in the above expression yields the following expression for the eigenfunctions of the Van der Pol oscillator

$$\varphi(x_1, x_2) = e^{\mu x_1} h\left(x_1 x_2 + \frac{\nu}{4} x_1^4 - \frac{\nu}{2} x_1^2\right).$$

It is not directly clear what functions h yield desirable properties for the reconstruction of the underlying dynamics.

2-4 Numerical methods for the Koopman operator

The ability of the Koopman operator to capture nonlinearities with linear models leads to the developments of numerical methods for the identification of the Koopman operator. In this section we give a brief overview of methods building on the development of Dynamic Mode Decomposition (DMD) by Williams et al. [4].

Finite-dimensional projection

In the context of numerical methods for the Koopman operator, we aim to approximate the infinitedimensional operator $\mathcal{K}$, acting on $\mathcal{G}(\mathcal{X})$ by a finite-dimensional approximation K_τ, acting on a subspace of $\mathcal{G}(\mathcal{X})$. This boils down to truncating the infinite series expansion in Equation (2-6) to a finite set of eigenfunctions based on the choice of the subspace considered.

The original developments on DMD take the space of observation functions $g(x) = x$ as the considered subspace, and extended DMD (eDMD) expands on this by augmenting the space with a prior dictionary of observation functions.

The projection of the Koopman operator on this subspace might not be closed however. This is clearly illustrated through an example by Kaiser et al. [18]. We consider the continuous-time Toy model presented in Section 2-2. For illustrative purposes we consider the generator projected onto the span of observables given by span $(\{x_1, x_2, x_1^2 x_1 x_2 x_2^2\})$. Decomposing onto these coordinates yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} \nu & 0 & 0 & 0 & 0 \\ 0 & \sigma & -\sigma & 0 & 0 \\ 0 & 0 & 2\nu & 0 & 0 \\ 0 & 0 & 0 & \nu+\sigma & -\sigma \\ 0 & 0 & 0 & 0 & 2\sigma \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2\sigma x_1^2 x_2 \end{bmatrix}$$

The residual term of $-2\sigma x_1^2 x_2$ gives perturbations in the generator and associated Koopman operator. As a consequence, the associated eigenfunctions are perturbed and do not behave linearly in the underlying system. If the estimate is to be used for prediction, it is therefore paramount to choose the observable coordinates such that linearity for the eigenfunctions is recovered. Another way of framing this problem, is to require the choice of basis functions to span a Koopman-invariant subspace.

Dynamic Mode Decomposition

In this section we aim to introduce DMD and illustrate the relation to the Koopman operator. We define a matrix of state measurements X, and its one step ahead measurement X^+. DMD then aims to identify a matrix $\hat{A}$ satisfying $X^+ = \hat{A}X$ by solving the least squares problem $\min_{\hat{A}} \|X^+ - \hat{A}X\|_2$. The solution of this optimisation problem can be computed directly by performing a singular value decomposition (SVD) on

$$X = U\Sigma V^*,$$

to obtain a solution

$$\hat{A} = U^* X^+ V \Sigma^{-1}.$$

The eigenvalues λ_i and eigenvectors w_i of $\hat{A}$ can be used to construct DMD-modes through

$$\hat{\phi} = X^+ V \Sigma^{-1} w.$$

Note that the original definition used to define the modes Uw. However, then the eigenvectors lie in the range of X instead of X^+ and are therefore not true eigenvectors of A. If the data was generated by linear dynamics, $x[k+1] = Ax[k]$ then these modes are the eigenvalue-eigenvector pairs of the projection of A onto the range of X^+ [17]. The algorithm is presented in Algorithm 1. A range of variations on Algorithm 1 exist for better insight or to gain

Algorithm 1: Dynamic Mode Decomposition

Input : Measurement data matrix pair (X, X^+)
Output: Evolution map $\hat{A}$ and a set W of DMD modes (λ, ϕ)

$(U, S, V) \leftarrow$ `SVD(`X`)`
$\hat{A} = U^* X^+ V S^{-1}$
`// Extract eigenvalue-eigenvector pairs and compute associated DMD mode`

for λ, w_λ **in** `eig(`$\hat{A}$`)` **do**
  $(\lambda_j, \phi_j) \leftarrow (\lambda, X^+ V S^{-1} w_\lambda)$
end
return (λ_j, ϕ_j), $\hat{A}$

a computational advantage if the measurement series is sequential. The key observation connecting DMD and the Koopman operator is that we can write the state in terms of the dynamic modes ϕ_j with amplitudes v_j,

$$x[k] = \sum_{j=1}^{n} \lambda_j^{k-1} v_j \phi_j.$$

Numerical methods

The effectiveness of DMD for the analysis of fluid dynamics problems spawned a broad range of modifications. In Table 2-2 an overview for these methods is provided. These modification range from the ability to deal with stochastic perturbations [29], adaptations for systems with specific mathematical properties [30–32], specific techniques for construction of models [6, 33] to models adapted to the control of nonlinear systems [5, 14].

We emphasised works focussing on the construction of models that are invariant on the considered subspace. One can achieve this by explicitly constructing models that are restricted to the space of eigenfunctions. These eigenfunctions can be constructed through deep learning [35, 36], or by modifying a prior subspace [18, 34]. The work by Korda and Mezić poses a method to exploit the Koopman PDE, constructing eigenfunctions directly from data in such a way that their span optimally recovers the underlying state measurements [1].

Table 2-2: Discussed methods in this section, satisfying objective properties based on discussion: 1. "✓" discussed in detail, 2. "‡" discussed briefly, 3. "-" not discussed. Invariance implies whether the constructed subspace is invariant to the Koopman operator. Properties of the method are characterised by 1. "+" if it holds , "=" if it holds under additional assumptions and "-" if does not hold. Lastly, the methods are characterised by design goal, "A" for analysis, "T" for theory, "P" for prediction and "C" for control. If an error analysis was performed, the associated performance metric is presented. Note that this performance metric cannot be compared between methods due to different error definitions.

Method	PDE's	Forced	Output	Stochastics	Full spectrum	Invariance	Neural Network	High Complexity	Prior info used	Goal	Performance (%)
ncDMD, fbDMD, tlsDMD [29]	✓	-	-	✓	-	-	-	-	-	A	10
subDMD [6]	✓	-	-	✓	-	-	-	-	-	A	5
Measure-preserving DMD [30]	✓	-	-	✓	✓	-	-	+	=	A	15
piDMD [31]	✓	-	‡	✓	-	-	-	+	=	A	-
Exploiting symmetry [32]	-	-	-	-	-	-	-	=	+	T	-
Pruning of eigenfunctions [34]	✓	-	‡	‡	-	‡	-	+	-	A	1
Deep learning of eigenfunctions [35]	✓	-	‡	-	✓	✓	+	+	-	A/P	10
Eigenflows [36]	-	-	‡	-	-	✓	+	+	+	P	1
SSD [33]	-	-	-	‡	-	✓	-	-	-	P	20
eDMD with control [14]	-	✓	-	-	-	-	-	-	-	A	-
eDMD for MPC [5]	✓	✓	✓	‡	-	-	-	-	-	C/P	25
KRONIC [18]	-	✓	-	✓	-	✓	-	=	-	C/P	-
Eigenfunction construction [1]	-	✓	‡	‡	-	✓	-	=	-	C/P	4

2-5 Eigenfunction construction

The differential equations of the Koopman eigenfunctions can be exploited to construct eigenfunctions using interpolation, instead of using a function dictionary. The work in this section was originally developed by Korda and Mezić [1]. We summarise their developments and indicate points for research.

Theory

The core idea is to exploit the time-independence of the differential equation in Equation (2-5). In this section we aim to construct an optimisation problem that uses the simplicity of this differential equation by considering a set of initial conditions $\Gamma \subseteq \mathcal{X}$. For simplicity, we first consider an arbitrary time-independent nonlinear system on $\mathcal{X} = \mathbb{R}^n$ of the form

$$\dot{x} = f(x) \tag{2-7}$$

Furthermore, we assume that the entire state x is available as measurements. On the set of initial conditions Γ, we assume that complex-valued continuous functions $g \in C(\Gamma)$, that, by definition, propagates linearly through Equation (2-5). Thus, given $g_0 = g(x_0)$ at $t = 0$, the dynamics are propagated linearly through time,

$$\psi(t) = e^{\mu t} g_0 = e^{\mu t} \varphi(x(0))$$

at $t > 0$. The function ψ is the time evolution of an eigenfunction φ over time, with associated eigenvalue μ. This eigenvalue μ is then an eigenvalue of the Koopman generator. The expression can be interpreted as the propagation of a function g on Γ propagated along trajectories of the system, illustrated in Figure 2-4. It remains to relate $x(t)$ back to $\psi(t)$. That is, find the eigenfunction φ satisfying the equality $\varphi(x(t)) = \psi(t)$.

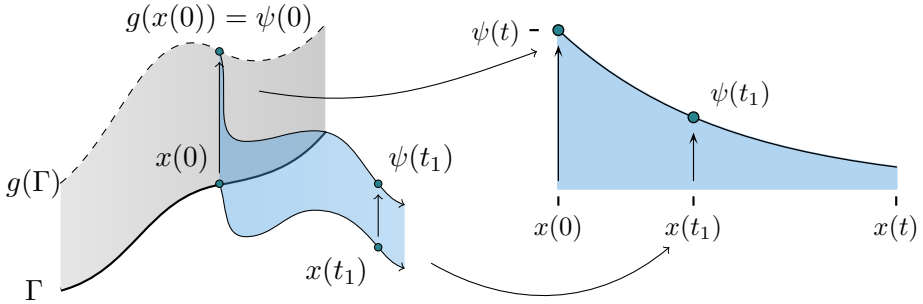

Figure 2-4: The trajectory traced by the function $\psi(t)$, starting from an initial condition $x(0) \in \Gamma$ can be unfolded along the trajectory $x(t)$ to obtain a linear evolution starting from an arbitrary function g on Γ.

Before we solve the problem of recovering φ from ψ, we introduce notation to formally define φ. We define a function $\nu(x)$ denoting the minimum time it takes to reach a given state x from the initial set Γ along the flow of the system, $\nu : \mathcal{X} \to T$. That is,

$$\nu(x) = \inf_{t \in T} \{t : F_{-t}(x) \in \Gamma\}$$

Using this notation we can define the eigenfunction φ on the state space $\mathcal{X}$, given that the trajectory originated from Γ,

$$\varphi(x) = \lambda^{\nu(x)} g(F_{-\nu(x)}(x)). \tag{2-8}$$

The eigenfunction effectively transports x back along the flow of the system to the set Γ, where it is lifted, and then propagated forward in time again. This is illustrated in Figure 2-5.

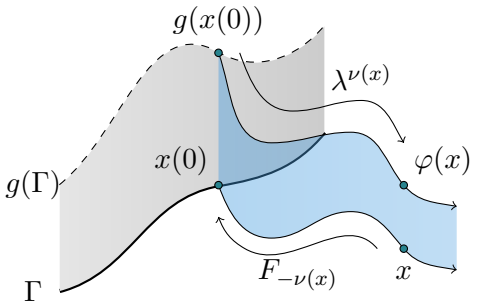

Figure 2-5: To construct an eigenfunction we translate x back along the flow F_t of the system, until it reaches the set Γ, where the state is lifted and propagated forward in time to find $\varphi(x)$.

We assumed to have control over our experiments, and hence we can perform experiments such that we obtain a number of trajectories originating from a set of initial conditions Γ. Given a trajectory $x_{(i)}$ with index i, originating at $x_{(i)}[0] \in \Gamma$, each subsequent measurement $x_{(i)}[k]$ is is related back to $x_{(i)}[0]$. Hence given a discrete-time Koopman eigenvalue λ and $g(x_{(i)}[0])$ we have

$$\varphi(x_{(i)}[k]) = \lambda^{k\tau} g(x_{(i)}[0]), \tag{2-9}$$

where τ is the time interval between measurements.

Theorem 2.2. *Equation* (2-8) *constitutes an Koopman generator eigenfunction with eigenvalue* $\mu = \log \lambda$ *of an autonomous system,*

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

Proof. We first compute the derivative along a trajectory $x(t)$. Then,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\varphi &= g(F_{-\nu(x)}(x))\frac{\mathrm{d}}{\mathrm{d}t}\lambda^{\nu(x)} + \lambda^{\nu(x)}\frac{\mathrm{d}}{\mathrm{d}t}g(F_{-\nu(x)}(x)) \\ &= \log(\lambda)\,\lambda^{\nu(x)} g(F_{-\nu(x)}(x)) + 0 \\ &= \log(\lambda)\,\varphi.\end{aligned}$$

We used that $\frac{\mathrm{d}}{\mathrm{d}t}\nu(x) = \frac{\mathrm{d}}{\mathrm{d}t}t = 1$ along a trajectory of the system. Similarly, $F_{-\nu(x)}(x) = x(0)$ remains constant along a trajectory of the system. Hence we can conclude that φ is an eigenfunction of the autonomous system. □

Given an arbitrary input-output system,

$$\begin{aligned}\dot{x} &= f(x,u) \\ y &= h(x).\end{aligned}$$

Korda and Mezić aim to derive a seperate model for each output variable y, with linear propagation of the eigenvalues. The assumption is that the system in lifted coordinates is linear, and that the measurement variable is a linear combination of the eigenfunctions,

$$y = c_1\varphi_1(x) + c_2\varphi_2(x) + \cdots + c_{N_g}\varphi_{N_g}(x).$$

Casting this into a standard linear model, this becomes

$$\begin{aligned} \chi[k+1] &= \begin{bmatrix} \lambda_1^\tau & 0 & & \\ 0 & \lambda_2^\tau & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & \lambda_{N_g}^\tau \end{bmatrix} \chi[k] + Bu[k] \\ \chi[k] &= \begin{bmatrix} \varphi_1(x[k]) & \varphi_2(x[k]) & \cdots & \varphi_{N_g}(x[k]) \end{bmatrix}^\mathsf{T} \\ y[k] &= \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}}_{C} \chi[k]. \end{aligned} \tag{2-10}$$

The eigenfunctions φ_j are propagated through time using their eigenvalues, yielding a diagonal evolution matrix. The observation matrix C has a 1 in every entry, since the coefficients c_j for a linear combination can be absorbed into the functions φ_j.

Note that if this method is applied to estimate the original state variable x, one estimates n models to ccompute the reconstruct the entire state variable.

Data equation

We presented a method to construct an eigenfunction, but have left the choice of the eigenvalues λ_j and initial conditions $g(x(0))$ open. In this section we aim to derive these eigenvalues and initial conditions from measurement data. We assume the availability of two datasets, one with, and one without forcing. We first consider the dataset without forcing, for which the linear model in Equation (2-10) lends itself to a data equation,

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[N] \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} \begin{bmatrix} g_1(x[0]) \\ g_2(x[0]) \\ g_3(x[0]) \\ \vdots \\ g_{N_g}(x[0]) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1^\tau & \lambda_2^\tau & \cdots & \lambda_{N_g}^\tau \\ \lambda_1^{2\tau} & \lambda_2^{2\tau} & \cdots & \lambda_{N_g}^{2\tau} \\ \vdots & \vdots & \ddots & \\ \lambda_1^{N\tau} & \lambda_2^{N\tau} & & \lambda_{N_g}^{N\tau} \end{bmatrix}}_{\Lambda} \begin{bmatrix} g_1(x[0]) \\ g_2(x[0]) \\ g_3(x[0]) \\ \vdots \\ g_{N_g}(x[0]) \end{bmatrix}.$$

The matrix Λ is a Vandermonde matrix, typically arising in signal analysis problems [19]. This data equation is used by Korda and Mezić to construct an optimisation problem for the eigenvalues λ_j and function values g_j for individual measurement variables y. To find a parameterisation for g, we find a range of initial conditions on Γ by augmenting the data with additional trajectories. These trajectories are included into the optimisation problem by

horizontal stacking of the data equation

$$\min_{\lambda_j, g_j} \left\| \begin{bmatrix} y_{(1)}[0] & y_{(2)}[0] & \cdots & y_{(q)}[0] \\ y_{(1)}[1] & y_{(2)}[1] & \cdots & y_{(q)}[1] \\ \vdots & \vdots & \ddots & \vdots \\ y_{(1)}[N] & y_{(2)}[N] & \cdots & y_{(q)}[N] \end{bmatrix} - \Lambda \begin{bmatrix} g_1(x_{(1)}[0]) & g_1(x_{(2)}[0]) & \cdots & g_1(x_{(q)}[0]) \\ g_2(x_{(1)}[0]) & g_2(x_{(2)}[0]) & \cdots & g_2(x_{(q)}[0]) \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_g}(x_{(1)}[0]) & g_{N_g}(x_{(2)}[0]) & \cdots & g_{N_g}(x_{(q)}[0]) \end{bmatrix} \right\|. \tag{2-11}$$

This optimisation problem is not convex, and thus requires specialised optimisation methods to prevent finding a local minimum. Korda and Mezić solve the optimisation problem using a descent method, initialised by the eigenvalues found by DMD to find a solution.

Interpolation

The data equation leads to estimates for g on the surface Γ, together with eigenvalues λ. Combining these with the known evolution presented in Equation (2-9), we can formulate an optimisation problem for the eigenfunctions φ. For each eigenvalue λ_i^τ, the optimisation problem becomes

$$\min_{\varphi} \sum_{j=1}^{q} \sum_{k=0}^{N} \|\varphi_i(x_{(j)}[k]) - \lambda_i^{k\tau} g_i(x_{(j)}[0]).$$

Each mode is used to propagate the function g on Γ over the state space, along the training trajectories. The optimisation problem is a *functional* optimisation problem, having a function as a solution. One can solve this by using interpolation to construct φ for the entire state space $\mathcal{X}$, through the use of basis functions, or by other methods such as sparse optimisation [37].

Implementing control

The autonomous dynamics can now be augmented by implementing a forcing input into the system, using a dataset where a forcing was present. By assuming that the forcing is additive, the knowledge of the evolution matrix A can be used to compute another optimisation problem. The measurement y_i at time instance k can be written in terms of the initial state and the time evolution.

$$y[k] = CA^k \chi[0] + \sum_{j=0}^{k-1} CA^{k-j-1} Bu[j].$$

The measurements y are known and the lifted state $\chi[0]$ can be computed using the estimated eigenfunctions $\varphi_j(x[0])$. The control input gain B is the only unknown in this expression, since the matrices A and C were computed in the previous subsection. We rewrite this expression in vectorised form,

$$\min_{B} \sum_{k=1}^{N} \left\| y[k] - CA^k \chi[0] + \sum_{j=0}^{k-1} \left(u^\intercal[j] \otimes (CA^j) \right) \mathrm{vec}(B) \right\|. \tag{2-12}$$

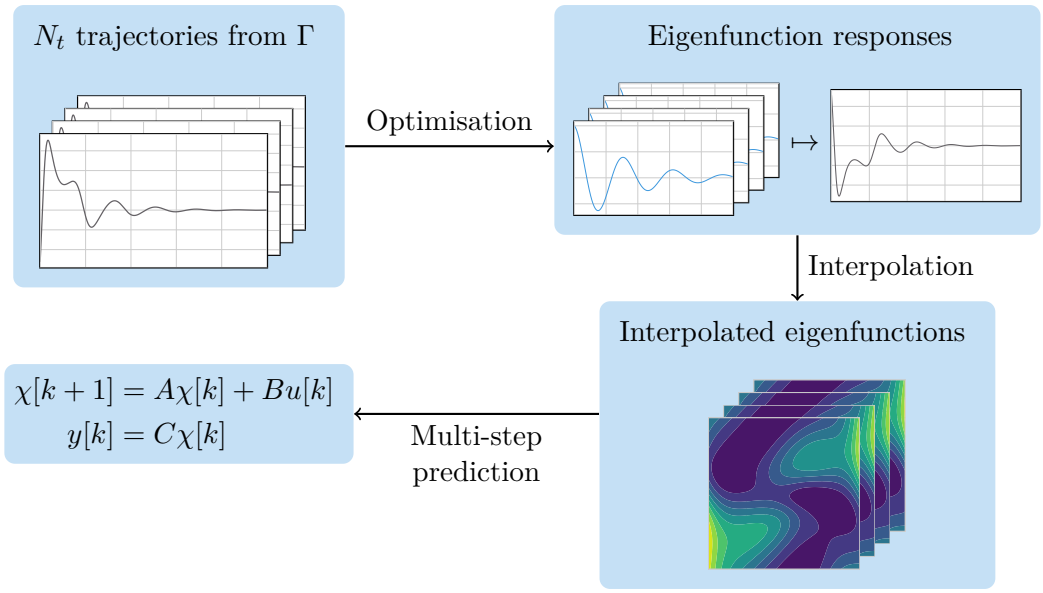

Figure 2-6: Diagram summarising the procedure by Korda and Mezić for the construction of a linear model through eigenfunction construction.

This is a least squares optimisation problem. To denote the solution, we define the matrices

$$\theta = \begin{bmatrix} y[1] - CA\chi[0] \\ y[2] - CA^2\chi[0] \\ \vdots \\ y[N] - CA^N\chi[0] \end{bmatrix}, \quad \text{and} \quad \Theta = \begin{bmatrix} u^\intercal[0] \otimes C \\ u^\intercal[0] \otimes (CA) + u^\intercal[1] \otimes C \\ \vdots \\ \sum_{j=0}^{k-1} \left(u^\intercal[j] \otimes (CA^j)\right) \end{bmatrix}.$$

Then B can be recovered by computing

$$B = \text{vec}^{-1}(\Theta^\dagger \theta)$$

Summary

The entire procedure by Korda and Mezić is summarised in Figure 2-6. By exploiting a set of sequences originating from a known subset of initial conditions $\Gamma \subseteq \mathcal{X}$, one can formulate an optimisation problem mirorring the eigenfunction PDE in Equation (2-5). The relationship of the eigenfunction dynamics to locations in the state-space result in a simple optimisation problem, allowing the eigenfunctions to reduce to interpolated values.

2-6 Koopman eigenfunction evolution for forced systems

Before concluding this chapter, we investigate the effect of an external forcing signal on the Koopman operator estimates. The Koopman operator is only defined for autonomous systems, since it assumes that the flow map F_τ of the dynamical system is only dependent on the state of the system. In this section we aim to extend the definition of the Koopman operator to

deal with a control input. There are two ways to approach this, one through a modification of the flow map F_t, and the other through the generator, applying Equation (2-4) to move the vector field f into continuous-time dynamics of the observation functions.

The flow map F_τ can be modified to introduce a time-independent forcing $u \in \mathcal{U}$ into the dynamics over a time interval τ, $x(t+\tau) = F_\tau(x(t))$. In this case, the Koopman operator K_τ becomes dependent on the given input u. That is,

$$\mathcal{K}_\tau(u)g(x) \coloneqq g \circ F_\tau(x, u).$$

This constitutes a 2-parameter group of operators, being a C_0-Semigroup in the coordinate τ. There is less structure in this problem, and consequently there appears to be limited work on such operator groups. For our purposes, this leads to linear parameter varying (LPV) models of the form

$$\chi[k+1] = \mathcal{K}_\tau(u)\chi[k],$$

with scheduling variable u. Unfortunately this poses little structure to decompose $\mathcal{K}_\tau(u)$ into parts without prior knowledge on the underlying system.

Instead we consider the generator definition, which allows us to apply Equation (2-4) to construct a Koopman model based on knowledge of the underlying vector field. As an example, we consider the systems discussed in Table 2-1. For systems that are linear in control, we have

$$Lg = \langle \nabla g | f + Bu \rangle = (\nabla g)^\intercal f + (\nabla g)^\intercal Bu$$

For systems that are affine in control, we have

$$Lg = \langle \nabla g | f + \eta u \rangle = (\nabla g)^\intercal f + (\nabla g)^\intercal \eta u$$

Following the differential equation in Equation (2-3) we need to integrate the forced generator definition over time. Given a constant input signal u, this is highly dependent on the underlying system. Instead, we can use a first-order Euler integration method to approximate the discrete-time evolution. We use the notation $\varphi[k]$ to denote the evolution of a *single* eigenfunction over time. Then,

$$\varphi[k+1] \approx \underbrace{\varphi[k] + \tau \frac{\mathrm{d}}{\mathrm{d}t}\varphi[k]}_{\varphi[k] \circ F_\tau} + \tau(\nabla\varphi[k])^\intercal \eta u[k] \approx \lambda^\tau \varphi[k] + (\nabla\varphi[k])^\intercal \eta u[k].$$

The terms $\varphi, \nabla\varphi$ and ν are functions on the state space $\mathcal{X}$, or equivalently, an element of an infinite-dimensional vector space. Under the assumption that the gradient lies in the span of eigenfunctions, $(\nabla\varphi)^\intercal \eta \in \mathrm{span}(\varphi_i)$, we can write this as a bilinear system in infinite-dimensional coordinates,

$$\varphi[k+1] \approx \lambda^\tau \varphi[k] + \tau B\varphi[k]u[k].$$

In this expression, the operator B acts on $\varphi[k]$ so that $B\varphi[k] = \tau(\nabla\varphi)^\intercal \eta$.

2-7 Conclusion and Problem statement

Interpretable black-box linear models for nonlinear systems

Koopman operator methods are typically applied to construct linear models of nonlinear systems and bear a close resemblance to linear methods for system identification. The estimates for the finite-dimensional projection of the Koopman operator constructed through methods such as DMD or eigenfunction construction do not tell much about the system on their own. However, the eigenfunctions associated to the eigenvectors make these models interpretable, giving information about periodic and decaying modes in terms of locations in the state space.

Problem Statement

Using the identified works, we were able to identify a range of open problems in the field. In this section we aim to summarise the problems that currently limit the applicability of Koopman operator methods for prediction and control. The core problems we were able to identify can be listed,

1. Koopman operator methods often apply only to autonomous systems.
2. Koopman operator methods often rely on the availability of full state measurements, there is limited investigation into the use of time-delay coordinates.
3. The choice of function dictionary is critical in the convergence properties of the Koopman operator. There is limited work on data-driven methods for function selection.

Korda and Mezić solve this problem through optimal selection of eigenfunctions. Although their work appears promising with direct applicability to the control of systems, it suffers from a few key problems.

- The problem of partial state measurements is not discussed.
- Isolated models are constructed for each measurement variable separately, yielding a large underlying state dimension.
- The method relies on a nonconvex optimisation problem for the selection of eigenfunctions
- Their method is only tested on two systems, the limitations of the method are not clear.

We derive the following research questions, building on the work from Korda and Mezić.

1. Can we avoid the nonconvex optimisation problem by exploiting the structure in the optimisation problem?
2. Instead of constructing isolated models, can we directly construct a combined linear model?

3. Can we improve the estimation performance of systems where the output map is not contained in the linear span of the constructed eigenfunctions.
4. Can we construct bilinear models using eigenfunction construction to improve predictive performance?
5. Can we extend the optimal eigenfunction construction to systems with partial state measurements, while maintaining a low-order model?
6. What are the limitations to the construction of optimal eigenfunctions in terms of robustness and system classes?

In the present work we aim to answer the above questions, summarised in the main research question.

> Can we construct bilinear models with nonlinear output from measurement data using optimal construction of Koopman operator eigenfunctions whilst maintaining a low computational complexity?

Chapter 3

Subspace methods for eigenfunction construction

In this section we aim to answer two research questions presented in Section 2-7. In particular, we investigate

1. Can we avoid the nonconvex optimisation problem by exploiting the structure in the optimisation problem?

2. Instead of constructing isolated models, can we directly construct a combined linear model?

We will show that both questions can be solved together, by identifying Dynamic Mode Decomposition (DMD) as a subspace method, drawing inspiration from another method, Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [19]. In the context of subspace methods, it is straightforward to generalise the optimisation problem to multivariable measurements, yielding a model of reduced order when compared to the work by Korda and Mezić.

The work in this section is structured by first deriving the main results, showing that the application of DMD to Hankel matrices and ESPRIT yield equivalent modes. Although the result is not difficult, this relationship is to our knowledge not recognised in existing works.

The application of DMD to the construction is investigated numerically. First, we investigate the effect of applying DMD to multiple trajectories numerically under white measurement noise, motivated by recognising that nonlinear systems might express nonglobal dynamics. Then, using techniques from linear subspace identification we define the hyperparameters used for the application of DMD to eigenfunction construction. These results are then compared to the results by Korda and Mezić.

3-1 Application of subspace methods to eigenfunction construction

Subspace methods are ubiquitous in system identification, and are typically applied to optimisation problems similar to the problem in Equation (2-11). In particular, we first aim to show that the optimisation problem can be solved in two steps. First, by performing a singular value decomposition to find the eigenvalues λ, and then finding the initial conditions $g(x_{(q)}[0])$. We will show that the DMD modes are equivalent to ESPRIT modes when applied to Hankel matrices. We will show that these methods estimate the correct eigenvalues under assumption of the given model in noise-free data. When introducing ESPRIT, we directly generalise to multivariable measurement data, laying the groundwork for the extension of Koopman eigenfunction construction to reduced order models.

Before introducing ESPRIT we will introduce notation for the top and bottom submatrices of a given matrix.

Definition 3.1 (Leading lower/upper principal submatrix)**.** Given a matrix $A \in \mathbb{C}^{n\times m}$, we define

- the leading upper principal submatrix $A_{(s)}$ as the matrix obtained by removing the last s rows from the matrix A.
- the leading lower principal submatrix $A^{(s)}$ as the matrix obtained by removing the first s rows from the matrix A.

The leading upper principal submatrix $A_{(2)}$ and lower principal submatrix $A^{(1)}$ then become

$$A = \begin{bmatrix} \overbrace{\begin{matrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m-1} & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m-1} & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \end{matrix}}^{A_{(2)}} \\ \underbrace{\begin{matrix} a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,m-1} & a_{n-1,m} \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m-1} & a_{n,m} \end{matrix}}_{A^{(1)}} \end{bmatrix}.$$

For the purpose of notational simplicity, we will additionally use the notation

$$\overline{A} = A^{(l)}, \qquad \text{and} \qquad \underline{A} = A_{(l)}$$

The parameter l will be clearly defined whenever we use this notation.

Equivalence of ESPRIT and DMD

ESPRIT was a method developed by Roy et al. in 1986 to estimates parameters in noisy sinusoidal signals [19]. This work was developed in a series of works in signal analysis, spawned by subspace ideas of Pisarenko [38]. ESPRIT assumes a sinusoidal signal model, written as

$$x[k] = \sum_{j=1}^{d} s_j e^{ik\omega_j} + \eta[k], \qquad \eta \sim \mathcal{N}(0, \sigma^2) \tag{3-1}$$

Here $s_j \in \mathbb{R}$ and $\omega_j \in (-\pi, \pi)$ are the parameters to be identified and η is additive white noise. Vectorising the data sequence, we obtain a data equation of the form

$$\underbrace{\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N] \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{i\omega_1} & e^{i\omega_2} & \cdots & e^{i\omega_d} \\ \vdots & \vdots & \ddots & \vdots \\ e^{Ni\omega_1} & e^{Ni\omega_2} & \cdots & e^{Ni\omega_d} \end{bmatrix}}_{\Lambda} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_d \end{bmatrix} + \begin{bmatrix} \eta[0] \\ \eta[1] \\ \vdots \\ \eta[N] \end{bmatrix}. \tag{3-2}$$

The structure in the Vandermonde matrix Λ is the main actor, allowing a time shift of the data sequence by multiplying by a diagonal matrix $D = \text{diag}(e^{i\omega_1}, e^{i\omega_2}, \dots e^{i\omega_d})$,

$$\overline{\Lambda} = \underline{\Lambda} D$$

Although the original work by Roy et al. was derived for single-state measurements, it is readily extended to state-space models with multiple measurements. Data generated by a general linear model of the form in Table 2-1 with $x \in \mathbb{R}^n$ yields that

$$\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N] \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^N \end{bmatrix}}_{\mathcal{O}} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_d \end{bmatrix} + \begin{bmatrix} \eta[0] \\ \eta[1] \\ \vdots \\ \eta[N] \end{bmatrix}. \tag{3-3}$$

satisfies a a similar shift property with $D = A$, albeit with a shift of n rows. We will call a variation of this property the subspace invariance property of order l.

Definition 3.2 (Subspace invariance of order l)**.** Let $k > d$. Suppose we have matrices $\Lambda \in \mathbb{C}^{N \times d}$ and $M \in \mathbb{C}^{d \times k}$, and a $d \times d$ invertible matrix D. Then the matrix product AM satisfies the subspace invariance property of order l if,

$$\Lambda^{(l)} M = \Lambda_{(l)} D M.$$

The name stems from a geometric perspective, where the subspace spanned by the columns of ΛM is invariant under removal of the first or last l rows.

Let $y \in \text{col}(\Lambda_{(l)} D) = \text{col}(\Lambda^{(l)})$. By definition, there exists x such that $\Lambda^{(l)} x = y$. But then, $y = \Lambda_{(l)}(Dx)$, from which $y \in \text{col}(\Lambda_{(l)})$ follows. Furthermore, we have that $\text{rank}(\Lambda_{(l)} D) = \text{rank}(\Lambda_{(l)})$ due to the invertibility of D. Hence, the column space of both matrices have equal dimension. We conclude

$$\text{col}(\Lambda^{(l)} M) = \text{col}(\Lambda_{(l)} M),$$

ESPRIT aims to exploit this property in the crosscorrelation matrices of the data sequence vectors X,

$$R_{\underline{X}\underline{X}^*} = \mathbb{E}[\underline{X}\,\underline{X}^*] = \Lambda S \Lambda^* + \sigma^2 I$$
$$R_{\underline{X}\overline{X}^*} = \mathbb{E}[\underline{X}\,\overline{X}^*] = \Lambda S D^* \Lambda^* + \sigma^2 Z.$$

Where we define Z as the cross-correlation between the noise one time-step ahead.

Given a vector of data X, as defined in Equation (3-2), we define the *Hankel matrix* with parameter s as

$$H = \begin{bmatrix} x[0] & x[1] & \cdots & x[N-s] \\ x[1] & x[2] & \cdots & x[N-s+1] \\ \vdots & \vdots & \ddots & \vdots \\ x[s-1] & x[s] & \cdots & x[N] \end{bmatrix} \tag{3-4}$$

If a sequence X satisfies the shift invariance property of order l, then so does the associated Hankel matrix.

The first-order time shift of the Hankel matrix is closely related to the crosscorrelation when x is one-dimensional. This relationship reads,

$$R_{\underline{X}\underline{X}^*} = \underline{H}\underline{H}^*, \qquad R_{\underline{X}\overline{X}^*} = \underline{H}\overline{H}^*.$$

ESPRIT estimates the eigenvalues λ of the shift matrix D by computing the generalised eigenvalues of the matrix crosscorrelation pair. In the noise-free case we can show that these generalised eigenvalues are the eigenvalues of the shift matrix D,

$$\left(R_{\underline{X}\overline{X}^*} - \lambda R_{\underline{X}\underline{X}^*}\right) x = \Lambda S \left(D^* - \lambda I\right) \Lambda^* x.$$

The underlying eigenvalues are then recovered by Hermitian conjugation. Under the influence of noise, the generalised eigenvalues are biased. The work by Roy et al. is able to compensate for this bias to recover the original eigenvalues [19]. Roy et al. already noted that the Hankel matrices could be used directly, so that the correlation matrices do not need to be computed. In fact, it is often preferably to act directly on data instead [39]. The next result shows that the Hankel matrices can be applied directly to derive the eigenvalues of the shift matrix D. Furthermore, we show that this is equivalent to the application of DMD to the Hankel matrix of measurements.

In Proposition 3.1 we consider a noise-free signal model. Under the influence of Gaussian measurement noise, the truncation of the singular value decomposition (SVD) to the modes with highest variation causes the correct eigenvalues to be estimated in the limit of infinite measurements, $N \to \infty$ [40].

Proposition 3.1 (ESPRIT equivalence to DMD modes)**.** Given a SVD for a data Hankel matrix $\underline{H} = USV^*$ satisfying the subspace invariance property in Definition 3.2 of order l, with $\overline{H} = \overline{\Lambda} M$, we have that

$$\sigma(D) \subseteq \sigma(U^* \overline{H} V S^\dagger)$$

Proof. We first show that for $\lambda \in \sigma(D)$ we have that $\lambda \in \sigma(\underline{H}^\dagger \overline{H})$. For this purpose, consider the generalised eigenvalues of the pair $(\overline{H}, \underline{H})$, defined as $\overline{H}x = \lambda \underline{H}x$. Then

$$0 = \left(\overline{H} - \lambda \underline{H}\right) x = \overline{A} \left(D - \lambda I\right) M x.$$

Thus any λ satisfying $(D - \lambda I)x = 0$, will also be a generalised eigenvalue of the pair $(\overline{H}, \underline{H})$. Then, by definition, we have that λ is an eigenvalue of $\underline{H}^\dagger \overline{H}$.

Using

$$\overline{H} = UU^*\overline{H}VV^*,$$

we can write for the Hankel matrix product

$$\overline{H}\underline{H}^\dagger = UU^*\overline{H}VS^\dagger U^*$$

Using the cyclic property for square matrices of the spectrum, $\sigma(AB) = \sigma(BA)$, we can conclude

$$\sigma(\underline{H}^\dagger\overline{H}) = \sigma(\overline{H}\underline{H}^\dagger) = \sigma(U^*\overline{H}VS^\dagger)$$

□

Multiple trajectories

The previous method was constructed for a Hankel matrix H of a single measurement sequence. However, for nonlinear systems the dynamics in different regions of the state space $\mathcal{X}$ might differ significantly. These can be included by horizontal stacking of the Hankel matrices without affecting the shift invariance. Denoting data sequences X_i, such as in Equation (3-3) with Hankel matrices H_i,

$$\begin{bmatrix}\overline{H}_1 & \overline{H}_2 & \cdots & \overline{H}_q\end{bmatrix} = \underline{\mathcal{O}}A\begin{bmatrix}M_1 & M_2 & \cdots & M_q\end{bmatrix}.$$

Where we consider a shift invariance of order n. As a consequence, we can apply the result in Proposition 3.1 to horizontally stacked Hankel matrices of different trajectories in order to extract eigenvalues from different trajectories.

Application to eigenfunction construction and reduced order models

ESPRIT will estimate the correct eigenvalues of the Koopman operator, under the assumption that the data can be explained by a model of the form in Equation (2-7). Application using a single measurement variable to the optimisation problem in Equation (2-11) is therefore unambiguous, by applying ESPRIT with a single shift operation. However, we can reduce the model order in this way as well. We will now assume a model of the form

$$\begin{aligned}\chi[k+1] &= A\chi[k]\\ y[k] &= C\chi[k]\\ \chi[k] &= \begin{bmatrix}\varphi_1(x[k]) & \varphi_2(x[k]) & \cdots & \varphi_{N_g}(x[k])\end{bmatrix}^\mathsf{T}.\end{aligned} \tag{3-5}$$

Where $A = \mathrm{diag}(\lambda_1^\tau, \lambda_2^\tau, \ldots, \lambda_{N_g}^\tau)$. The largest difference with the original model considered in Equation (2-7) is the introduction of the linear output map $C \in \mathbb{C}^{N_g \times m}$. This condenses the m different models needed for the original method, to a single model. Consequently, we reduce the model order from mN_g to N_g.

The data equation for the condensed model becomes

$$\begin{bmatrix}y[0]\\ y[1]\\ \vdots\\ y[N]\end{bmatrix} = \underbrace{\begin{bmatrix}C\\ CA\\ \vdots\\ CA^N\end{bmatrix}}_{O}\begin{bmatrix}g_1(x[0])\\ g_2(x[0])\\ \vdots\\ g_{N_g}(x[0])\end{bmatrix}.$$

Note that this matrix format again lends itself to a shift invariance property of order m,

$$O^{(m)}G = O_{(m)}AG$$

Therefore, the eigenvalues of the Koopman operator in the matrix A can be computed using with the proposed generalised ESPRIT in Proposition 3.1.

Linear output map

To find the linear output map C, we seperate the measurement model into individual measurements. Writing for the linear output map

$$C = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1N_g} \\ c_{21} & c_{22} & \cdots & c_{2N_g} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mN_g} \end{bmatrix}$$

we find for individual state measurements that

$$\begin{bmatrix} y_i[0] \\ y_i[1] \\ \vdots \\ y_i[N] \end{bmatrix} = \begin{bmatrix} C_i \\ C_iA \\ \vdots \\ C_iA^N \end{bmatrix} \begin{bmatrix} g_1(x[0]) \\ g_2(x[0]) \\ \vdots \\ g_{N_g}(x[0]) \end{bmatrix} = \Lambda \begin{bmatrix} c_{i1}g_1(x[0]) \\ c_{i2}g_2(x[0]) \\ \vdots \\ c_{iN_g}g_{N_g}(x[0]) \end{bmatrix} = \Lambda \begin{bmatrix} g'_{i1}(x[0]) \\ g'_{i2}(x[0]) \\ \vdots \\ g'_{iN_g}(x[0]) \end{bmatrix}$$

Thus for each measurement i of the output y, a scaled version of $g_i(x[0])$ can be found by taking the pseudo-inverse of the Vandermonde matrix Λ and computing $\mathcal{O}_i^\dagger Y$. We choose for all j,

$$g_j(x[0]) = \max_{i=1,2,\ldots,m} |g'_{ij}(x[0])|.$$

The elements c_{ij} may then be computed

$$c_{ij} = \frac{g'_{ij}(x[0])}{g_j(x[0])}.$$

Complex conjugate pairs

In our systems, we have real-valued measurements $y[k] \in \mathbb{R}^m$ for all k. Consequently, any eigenvalue $\lambda \in \sigma(A)$ has a complex conjugate $\lambda^* \in \sigma(A)$. This follows from every matrix in Proposition 3.1 being a real-valued matrix, since the measurement data in H is real.

Considering the Koopman operator eigenvalue properties in Proposition 2.1, we know that if $\lambda_i = \lambda_j^*$ that the associated eigenfunctions satisfy $\varphi_i = \varphi_j^*$. Consequently,

$$g_i(x[0]) = g_j^*(x[0]) \tag{3-6}$$

Thus, to find the eigenfunctions, the interpolation problem only needs to be solved for the functions associated to the eigenvalues with $\mathrm{Im}(\lambda) \geq 0$, reducing the number of interpolation problems that need to be solved.

3-2 Numerical study

To assess the performance of subspace methods in the application of eigenfunction construction we will outline a numerical comparison of ESPRIT in the presence of white noise and in the application to eigenfunction construction. First, we perform a quantitative comparison for eigenvalue estimation accuracy to compare the performance under the condition of using multiple trajectories. The proposed method is then applied to the eigenfunction construction problem and compared to the original results by Korda and Mezić on uncontrolled systems. Lastly we compare the reduced order models in order to assess the predictive performance under influence of a square-wave forcing.

Quantitative study on accuracy under noise

To better understand the effect of the Hankel matrix construction on the performance of subspace methods under measurement noise we first perform a short numerical study. In this section we aim to estimate the eigenvalues of the matrix D for sequences following a signal model of the form in Equation (3-1).

Typically, to normalise the noise variance with respect to the signal magnitude we use the signal to noise ratio (SNR). The SNR is defined in terms of the noise variance σ and mean-squared signal magnitude $\mathbb{E}[x[k]^2]$ on a logarithmic scale,

$$\text{SNR} = 10 \log_{10} \left(\frac{\mathbb{E}[x[k]^2]}{\sigma} \right). \tag{3-7}$$

Thus in the limit SNR $\to 0$, we have that the measured signal consists solely of noise. Similarly, for SNR $\gg 1$, the noise on the signal is almost negligable. Typical values for real-world measurement data ranges range from 10 to 100 dB [40].

Measurement noise on the data results in a limit on the minimum eigenvalue that can be identified. Epps and Techet propose the following bound on the singular values s_i, for the mode to be noise-free [40, 41]. This bound is expressed in terms of the dimensions of the $n \times m$ data matrix H on which the SVD is performed, assuming a noise model of the form $H + \eta$, with $\eta \sim \mathcal{N}(0, \sigma^2)$. Yielding,

$$s_i > \sigma \sqrt{nm}. \tag{3-8}$$

The introduction of measurement noise on the data matrix introduces a small bias on the expected value of the estimated singular values, with order of magnitude being the ratio between the noise variance and the size of the singular value [41],

$$\mathbb{E}[\tilde{s}_i] = s_i + \mathcal{O}\left(\frac{\sigma^2}{s_i} \right).$$

Performance of an estimation method is measured through the relative error between the nominal eigenvalue λ and the estimated eigenvalue $\hat{\lambda}$,

$$\varepsilon = \left\| \frac{\lambda - \hat{\lambda}}{\lambda} \right\|. \tag{3-9}$$

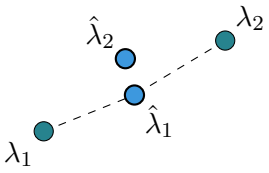

Figure 3-1: For a set of true eigenvalues λ_1 and λ_2 and their unordered estimates $\hat{\lambda}_1$ and $\hat{\lambda}_2$ it is unclear how they should be paired.

To assess the performance of the proposed method under SNR we compare DMD applied to Hankel matrices generated from a single trajectory, and applied to a horizontal stack of Hankel matrices. For SNR's between 5 dB and 100 dB, we generate 24 random eigenvalues pairs. These eigenvalues are distributed uniformly along a circle of radius $r \sim \mathcal{N}(1, 0.1)$. For each set of eigenvalues, we generate 40 trajectories of length 50.

On this data the proposed methods are applied to generate eigenvalue estimates $\hat{\lambda}$, for which their relative error is computed. This poses another problem however, since we now have two unordered sets of eigenvalues. These sets do not have the same elements, nor do all elements correspond to a correct eigenvalue estimate. This is illustrated Figure 3-1.

Eigenvalue pairing

The problem of eigenvalue pairing can be reduced to a formal problem. Given two sets A and B, we want to find for each $x \in A$ a the closest complex number $y \in B$ such that $|x - y| < \epsilon$. This number may not exist, if the method failed to identify the eigenvalue due to noise in the measurement data.

This problem can be casted into a mixed-integer linear programming (MILP) problem. Suppose A and B are of size n_A and n_B respectively. Then we index the sets A and B with $i = 1, 2, \ldots, n_A$ and $j = 1, 2, \ldots, n_B$ respectively. Then we can introduce a binary variable $z_{i,j}$, denoting whether we pair $x_i \in A$ and $y_j \in B$. The goal then becomes to maximise the used number of elements under the conditions that each element of A and B is used at most once, whilst satisfying the condition that $|x_i - y_j| < \epsilon$. The full optimisation problem is then,

$$\begin{aligned} \min_{z_{i,j} \in \{0,1\}} \quad & \sum_{i,j} z_{i,j} |x_i - y_i| - z_{i,j} & & \\ \text{s.t.} \quad & \sum_{i} z_{i,j} \leq 1, & & j = 1, 2, \ldots, n_B \\ & \sum_{j} z_{i,j} \leq 1, & & i = 1, 2, \ldots, n_A \\ & z_{i,j} |x_i - y_i| < \epsilon & & \forall i, j \end{aligned} \tag{3-10}$$

We solve this optimisation problem using `scipy`'s MILP solver, which uses an implementation Huangfu and Hall [42]. The maximum absolute eigenvalue error allowed is set at $\epsilon = 0.1$ in our work.

Hyperparameter selection

There are three hyperparameters that are left open to choice at this point. Namely, the time delay τ, the Hankel matrix parameter s and the number of eigenvalues N_g.

There is a range of research on selection of the time-step τ and Hankel matrix parameter s for subspace methods. Typically, the time-step is taken at 10 times the bandwidth of the system [20]. However, for the interpolation problem in Equation (3-6), we want τ smaller for a more accurate representation of the lifting function. To solve this, we propose a two-step approach, selecting a time step τ based on literature for the subspace identification in Proposition 3.1, and using another time step $\tau' < \tau$ to solve the interpolation problem in Equation (3-6). The eigenvalues of time step τ' are related to the eigenvalues of τ through

$$\lambda^{\tau'} = (\lambda^{\tau})^{\frac{\tau'}{\tau}} . \tag{3-11}$$

The Hankel matrix parameter s is typically chosen by looking at the singular values in the data. For this purpose, we plot the singular values in S of the SVD of $\underline{H} = USV^*$ for s' greater than the expected model order, selecting the order such that the *elbow* point is contained in the data [20, 43].

Like the previous section, the noise floor of the singular values limit the smallest singular value that feasibly describes the data. Therefore the limit in Equation (3-8) determines an expected lower bound, and can be used as prior knowledge in the process of the model order selection s [41].

Quantitative comparison in application to Eigenfunction construction

We compare the performance of the proposed method to results generated by the original work by Korda and Mezić. We report on the variance accounted for (VAF) of the unforced dynamics for the models presented in Section 2-2.

All models were derived from 50 random trajectories of length 2500 with a time-step of 0.01, originating from a circle of radius 9 for the Toy model, a radius of 1 for the Duffing oscillator and a radius 0.02 for the Van der Pol oscillator. The differential equations of these systems, presented in Section 2-2 are solved with a dynamic Runge-Kutta integration, using Scipy's initial-value-problem solver with default settings. These circles define the sets Γ proposed in Section 2-5. A subset of these trajectories are presented in Figure 3-2.

To gauge the sensitivity of the proposed method to measurement noise, the trajectories are perturbed by gaussian noise with an SNR of 20 dB.

To identify the proposed models, the trajectories were resampled at the time step presented in the next section. The derived eigenvalues were then converted to the smaller sampling time for interpolation using Equation (3-11). Similarly, to compute the VAF, we use the autonomous dynamics of 25 trajectories of length 2500, originating from a uniform distribution on the aforementioned circles.

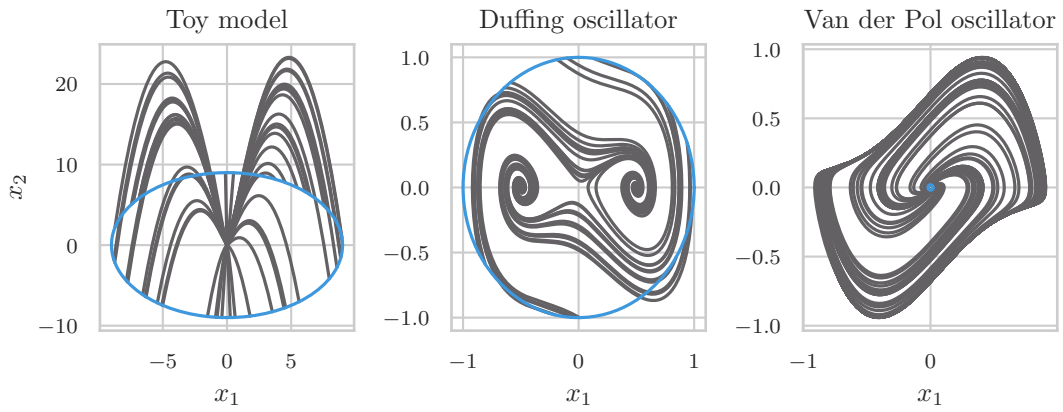

Figure 3-2: A subset of the training trajectories used for construction of Koopman eigenfunctions for the discussed models. The set Γ of initial conditions is presented in blue.

3-3 Results

Estimated eigenvalue accuracy

The relative error in the estimated DMD modes are presented in Figure 3-3 and Table 3-1 for a range of SNRs. The distribution plot in Figure 3-3 presents primarily the order of magnitude of the relative error in the eigenvalue magnitude and the associated variation. We find that the eigenvalues found by DMD when applying only a single trajectory for estimation is slightly more accurate and more concentrated around the mean order of magnitude. However, the

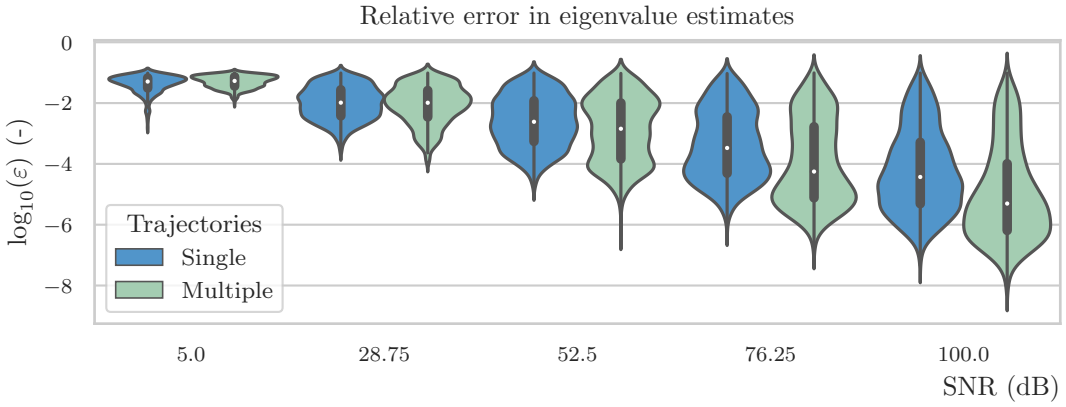

Figure 3-3: Distribution in the magnitude of the relative error in estimated eigenvalues for DMD applied to hankel matrices for the estimation of eigenvalues in terms of the SNR. Both the use of a single trajectory and 40 trajectories is investigated.

results in Table 3-1 paint a different picture. Whenever only a single trajectory is included for the identification of eigenvalues the method fails to accurately identify about 20% of these modes, whilst DMD increases in accuracy as the SNR decreases toward 0.

Table 3-1: Mean order of magnitude in the relative error with standard deviation, and the accuracy in estimated eigenvalues for DMD applied to Hankel matrices for the estimation of eigenvalues in terms of the SNR. The results are presented with the use of a single trajectory and for 40 trajectories.

SNR (dB)		5	28.75	52.5	76.25	100
Relative error $\log_{10}\varepsilon$ (-)	Single	-1.4 ± 0.3	-2.0 ± 0.5	-2.6 ± 0.8	-3.4 ± 1.1	-4.2 ± 1.3
	Multiple	-1.3 ± 0.2	-2.1 ± 0.6	-2.9 ± 1.1	-3.9 ± 1.4	-5.0 ± 1.5
Accuracy (%)	Single	28	63	82	89	97
	Multiple	25	67	84	92	95

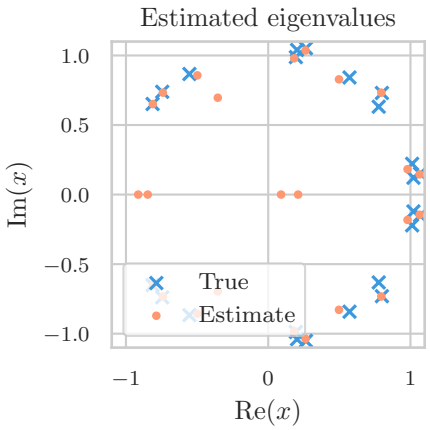

Figure 3-4: Eigenvalue estimates produced by DMD under a SNR of 28.75 dB estimated with 40 trajectories of length 50. The eigenvalues are uniformly distributed across a circle with a gaussian distributed radius around 1.

In Figure 3-4 we present the eigenvalues estimated by DMD by using 40 trajectories of length 50 at a SNR of 28.75 dB. Comparing these with the results in Table 3-1 we find that the accuracy is primary lost whenever eigenvalues are positioned close together. Under the noise variation and trajectory length these eigenvalues become indistinguishable for the presented method. These result in incorrectly estimated eigenvalues.

Application of subspace methods to eigenfunction construction

We first estimate the time step required to capture the dynamics in the system. The step responses of the system are presented in Figure 3-5. For linear systems, one can approximate the bandwidth of the system by looking at the step-response, and approximating the dynamics by a second-order model. The 3 dB response time then gives an estimate for this bandwidth.

To select the model order, we consider the magnitude of singular values in the Hankel data matrix. These magnitudes are presented for both noise-corrupted and noise-free data in Figure 3-6. The results are presented together with the theoretical bound presented in Equation (3-8). We indeed find that the theoretical bound is on a similar order of magnitude to the noise floor.

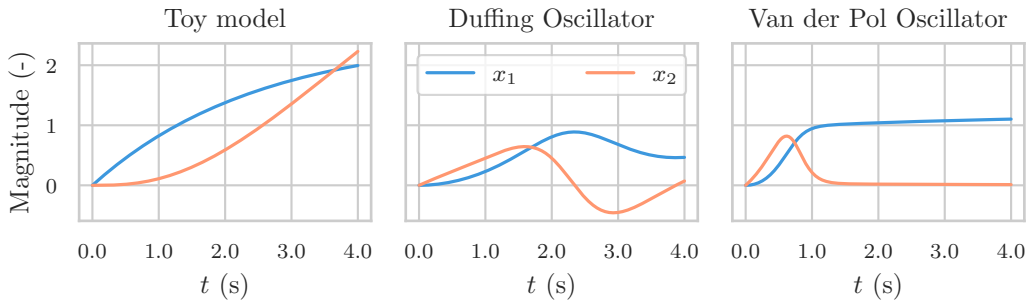

Figure 3-5: Step response over time to a unit input signal of the Toy model, the damped Duffing oscillator and the Van der Pol oscillator.

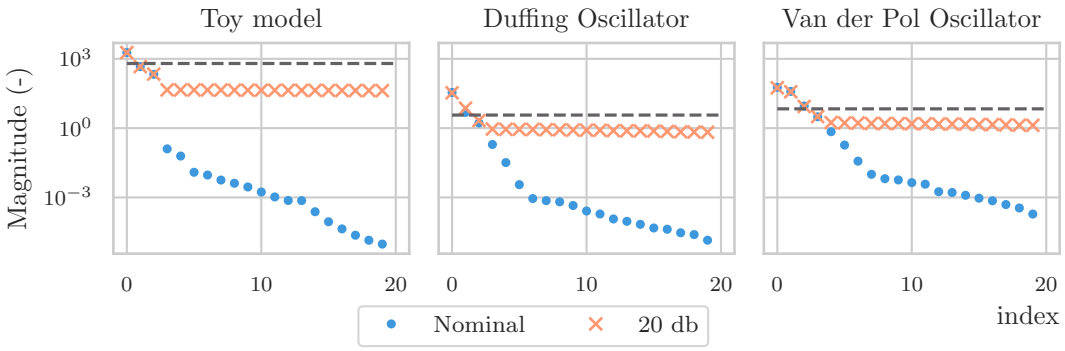

Figure 3-6: The magnitude of the singular values of the Hankel data matrix up to $s = 20$ for both noise-free data and with noise-corrupted data with $\mathrm{SNR} = 20$. The dashed line is the theoretical bound on modes that are not corrupted by noise. The singular values are presented for the Toy model, the Duffing oscillator and the Van der Pol oscillator.

For linear systems a gap in the singular value magnitude typically emerges, similar to the gap in singular values for the Toy model in Figure 3-6. However, for the more complex nonlinear systems, there does not appear to be such a gap. The autonomous dynamics in the toy model can be reconstructed with three eigenfunctions, as shown in Section 2-3. However, the eigenfunctions for the Duffing oscillator and Van der Pol oscillator do not lead to an exact reconstruction of the state variables. These emerge as singular values with small variation in the Hankel matrices.

For the noise-corrupted models we choose the order at the noise boundary, for the nominal order, we take the characteristic elbow point for the nonlinear models. For further comparison we also take an extended model with a larger model order, to compare the performance under the addition of further modes. All the parameters of the identification cycle are presented in Table 3-2.

Table 3-2: Estimated parameters for the identification of the Toy model, Duffing oscillator and Van der Pol oscillator.

	Toy model	Duffing oscillator	Van der Pol oscillator
Time step	0.1	0.2	0.2
Noisy order	3	3	3
Nominal order	3	6	8
Extended	3	14	16
Korda	6	28	32

The identified autonomous dynamics are presented in Figure 3-7. We can only really gauge the initial response for each model, since they do not include control and therefore converge to an equilibrium or a periodic attractor. We find that for most systems the identified models capture the general dynamics in the system. Furthermore, we present the VAF for a validation set of autonomous responses in Figure 3-8 and Table 3-3.

Table 3-3: VAF with standard deviation for state estimates in the Toy model, Duffing oscillator and Van der Pol oscillator for different model orders.

	State	Toy model	Duffing oscillator	Van der Pol oscillator
Noisy	x_1	25 ± 30	69 ± 31	44 ± 5
	x_2	97 ± 10	33 ± 16	33 ± 3
Nominal	x_1	96 ± 16	71 ± 26	90 ± 3
	x_2	98 ± 9	30 ± 16	68 ± 2
Extended	x_1	96 ± 16	25 ± 22	96 ± 3
	x_2	98 ± 9	21 ± 17	91 ± 6
Korda	x_1	100 ± 0	82 ± 33	96 ± 0
	x_2	100 ± 1	82 ± 32	55 ± 1

Noise-corrupted models

We find that the impact of noise on measurement data for the Toy model is significantly larger than for the Duffing and Van der Pol oscillators. Looking at the responses in Figure 3-7 we

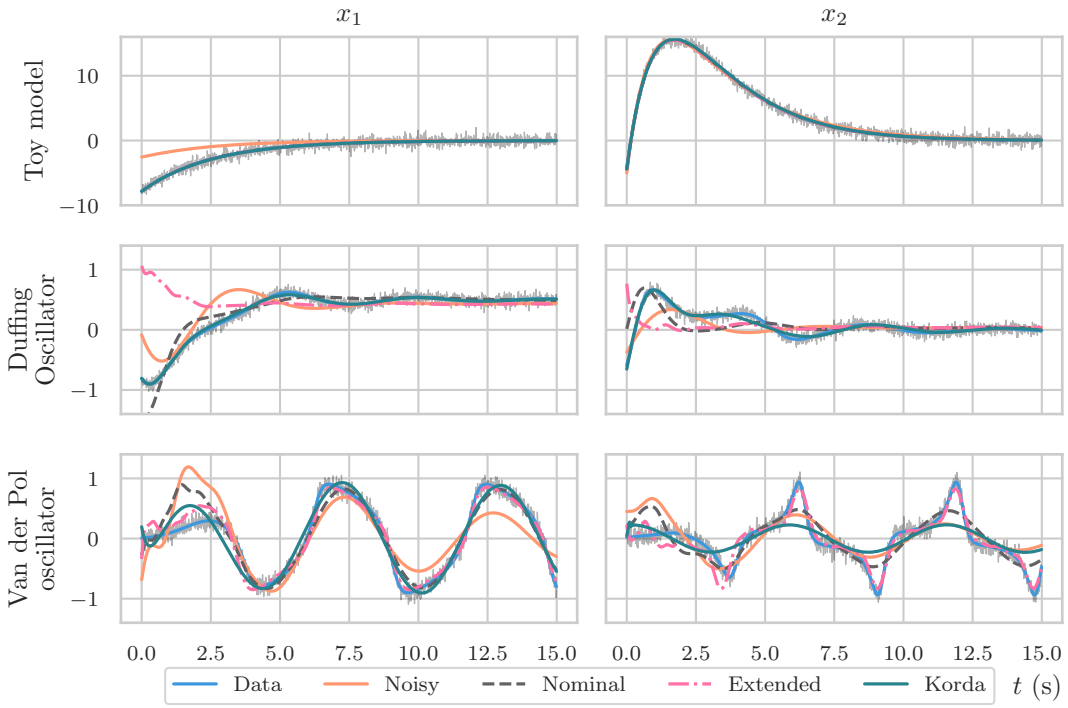

Figure 3-7: Autonomous dynamics of the Toy model, Duffing oscillator and Van der Pol oscillator over time for both state variables. The validation data is presented in blue, with grey representing an SNR of 20 dB. Furthermore, the dynamics identified from noise-corrupted data (orange), the nominal model order (dark grey), the extended model (pink) and by Korda and Mezić (cyan).

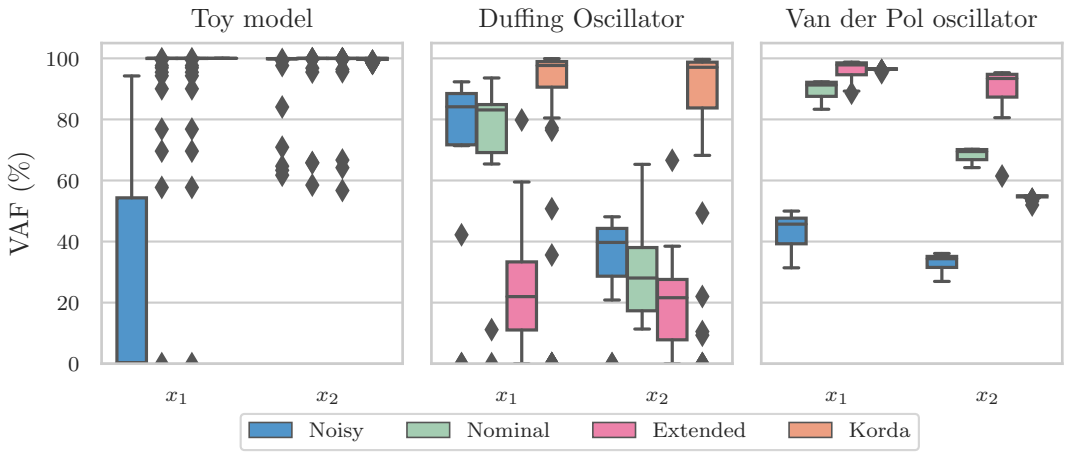

Figure 3-8: VAF in the single states for the multi-step prediction in autonomous dynamics of the models identified from noisy data (blue), for a nominal model (green) and for the extended model (pink). Furthermore, the results by Korda and Mezić (orange) are presented for comparitive purposes.

find that this can mostly be attributed to the inaccuracy in the initial state estimate. This can mostly be attributed to the interpolation problem being performed at noisy states x. For the toy model, the response characteristic occurs at a similar time-scale as the true dynamics. For the other models, it is difficult to compare the dynamics due to the smaller number of eigenvalues used.

Nominal and extended models

The performance of the nominal and extended models for the Toy model is mostly similar, with a slight discrepancy. This can be attributed to the interpolation method used. Due to the convexity of the state space in which the dynamics take place, Pythons' interpolation method cannot always produce an estimate. Hence we had to resort to nearest-neighbour interpolation, producing less accurate results, especially around points that have different trajectories along slight variations. Matlabs' linear interpolation method does allow for extrapolated values, allowing the results by Korda and Mezić to be slightly more accurate.

We note that subspace methods appear unable to reconstruct the dynamics for the Duffing oscillator. The Duffing oscillator is the only model that has two fixed points, causing the system to be unable to be described by a linear model, which is one of the underlying assumptions in using subspace methods. The original method proposed by Korda and Mezić does reproduce the dynamics accurately, possibly due to the extra flexibility introduced by using an optimisation algorithm.

From Figure 3-7 we also find that the additional states from the extended models are required to reconstruct the attractor dynamics for the Van der Pol oscillator accurately. This suggests that the elbow point used for linear systems is not a good indicator for the required Koopman model order.

The forced responses in Figure 3-9 is poor for the proposed models. Since the autonomous response for the Duffing oscillator was already quite poor, this is expected. For the other models however, the difference can be attributed to the fact that linear models are a poor fit Koopman dynamics, illustrated in Section 2-6.

Models by Korda and Mezić

The results originally presented by Korda and Mezić could not be reproduced on time scales larger than presented in the original paper, both for the autonomous response in Figure 3-7 and the forced response in Figure 3-9 [1]. The time frame originally presented are all within 2 seconds of the dynamics, suggesting that their method is not identifying a true Koopman model, but overfitting to the small time scale. This originally caused the method to fail for the Van der Pol oscillator, which has attractor dynamics with a longer time scale. This was not discussed in the original work and poses a significant challenge to identify the forcing gain B in multi-step prediction. This suggests that the work by Korda and Mezić identifies an k-step ahead predictor, where k is limited by the timescale of the used training data for optimisation.

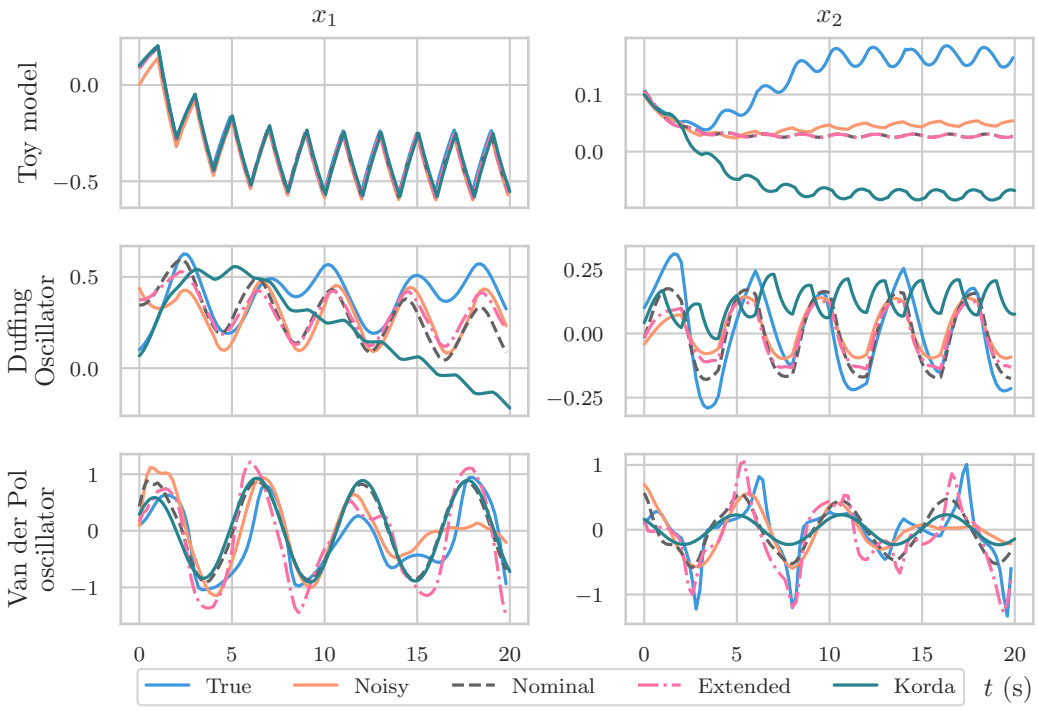

Figure 3-9: Prediction performance of the identified system dynamics under the influence of a square-wave forcing with a period of 2 seconds.

3-4 Stiffness of the Van der Pol oscillator

The results presented in the previous section are limited to a small subset of the entire state-space. In this section we take the Van der Pol oscillator as an example. The vector field in Figure 2-2 showed that the dynamics yield a very fast and sharp response for $\|x\| > 1$, while maintaining slow dynamics for $\|x\| < 1$.

The nonlinearity dominates the initial response, converging slowly to the limit cycle. This behaviour is illustrative of the *stiffness* of the Van der Pol oscillator. To define this property more clearly, we consider the stiffness ratio, defined by

Definition 3.3 (Stiffness ratio [44])**.** Given the Jacobian J of the vector field f of a differential equation, the stiffness ratio of f is defined as the ratio between the largest and smallest eigenvalue of $J(x)$, where x is allowed to vary over a small region X.

$$S = \frac{\max_{\lambda \in \sigma(J(x)), x \in X} |Re(\lambda)|}{\min_{\lambda \in \sigma(J(x)), x \in X} |Re(\lambda)|}.$$

The stiffness ratio is typically used to characterize the performance of numerical integration methods. A high stiffness ratio requires an integration method to adapt the step size to both fast dynamics and slow dynamics within as a trajectory moves through the region X.

To fast and slow dynamics will be associated to different Koopman modes, which need to both be identified by our subspace method. The quick dynamics take up a comparatively small fraction of the data, while the slower dynamics are dominant until a limit cycle or equilibrium point is reached. Consequently, the variation is attributed to a comparatively small singular value in ESPRIT.

For the Van der Pol oscillator with our parameters we can compute the jacobian,

$$J(x) = \begin{bmatrix} 0 & 2 \\ -0.8 - 20x_1x_2 & 2 - 10x_1^2 \end{bmatrix}$$

The eigenvalues of the jacobian can be computed analytically in terms of the state x,

$$\lambda_1 = -x_2 + 5x_1^2 + \sqrt{(2x_2 - 10x_1^2) + 4(1.6 + 20x_1x_2)}$$
$$\lambda_2 = -x_2 + 5x_1^2 - \sqrt{(2x_2 - 10x_1^2) + 4(1.6 + 20x_1x_2)}$$

The first state variable x_1 dominates the magnitude of the eigenvalues. Consequently, we can expect that when x_1 is much larger than x_2 that the stiffness ratio S grows large. For large initial states, the dynamics are first attracted to the axis $x_2 = 0$, moving the dynamics into a region with high stiffness ratio. This causes the presented identification method to fail whenever the nonrecurrent set includes states with $|x| > 2$. We present the stiffness ratio for the Van der Pol oscillator in Figure 3-10, where we take for X a disk of radius $r = 0.05$ around the given coordinate. The fast dynamics previously mentioned are clearly visible as curved bands in the stiffness ratio. The stiffness ratio inside these bands are of a much larger order of magnitude than outside of these bands. The vertical bands emerge due to the eigenvalues with 0 real part of the Jacobian at $x_1 = \sqrt{0.2}$.

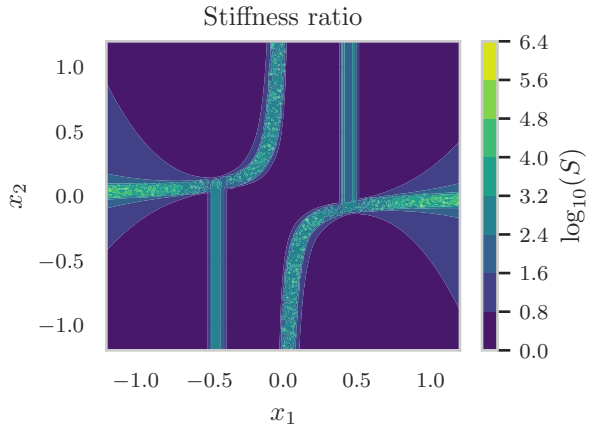

Figure 3-10: Stiffness ratio for the Van der Pol oscillator computed for a disk of radius 0.05 around the given coordinate.

3-5 Conclusion

By applying DMD to Hankel matrices we can correctly identify the eigenvalues required for reconstruction of the nonlinear dynamics, whilst reducing the model order originally required by Korda and Mezić. The autonomous response of these models have an average VAF ranging from 96% for the Toy model and Van der Pol oscillator to 25% for the Duffing oscillator. This is an improvement compared to the reproduction of the original work by Korda and Mezić on the considered time scales.

It was found that a large model order is required for accurate reconstruction, and that there are significant limitations on the subset for which the model is valid. These properties emerged due to the stiffness of the Van der Pol oscillator. Furthermore, the assumption of a linear model for a nonlinear system with multiple equilibria causes the proposed method to fail, whereas the original optimisation problem is capable with dealing with such systems.

We were unable to recover the forced dynamics from the proposed models. Previous results in Chapter 2 suggested that a linear model is a poor fit for the nonlinear forced dynamics in eigenfunction coordinates.

We can answer the questions posed at the start of this chapter positively. We succesfully exploited the structure in the optimisation problem to extract eigenvalues and initial conditions from data. At the same time, we were able to combine both models into a single model, of a significantly reduced order when compared to the results by Korda and Mezić.

Chapter 4

Derived bilinear models for Koopman eigenfunction construction

The properties of the Koopman operator eigenfunctions presented in Proposition 2.1 can be exploited to construct an output map purely from Koopman eigenfunctions. In this chapter we aim to provide answers to the following research questions.

1. Can we improve the estimation performance of systems where the output map is not contained in the linear span of the constructed eigenfunctions.
2. Can we construct bilinear models using eigenfunction construction to improve predictive performance?

To illustrate the problem covered by the first research question, we expand on the derivation in Example 2-3.1. The two eigenfunctions posed at the end of the derivation would lead to a model where the state measurement function $g(x) = x_2$ is not contained in the span of eigenfunctions, $g \notin \text{span}(\varphi_1, \varphi_2)$ associated to eigenvalues ν and σ respectively. That is, there do not exist c_1 and c_2 such that

$$x_2 = c_1 x_1 + c_2 \left(x_2 - \frac{\sigma}{\sigma - 2\nu} x_1^2 \right)$$

for all x_1 and x_2. However, there is a solution when we include φ_1^2 in the output vector span

$$x_2 = 1 \left(x_2 - \frac{\sigma}{\sigma - 2\nu} x_1^2 \right) + \frac{\sigma}{\sigma - 2\nu} (x_1)^2$$

This motivates the question whether we can extend eigenfunction construction to models with a polynomial output map, in order to get improved state reconstruction estimates.

The results presented in Section 3-3 correctly identify that a third order model is required for the dynamics to be recovered by a linear model. However, the derivation for the Van der Pol

model in Example 2-3.2 shows that the eigenfunctions quickly become intricate expressions, causing the linear model to require a large model order to accurately capture the dynamics. The aim then becomes to construct an output model, improving the approximation of the underlying state, without requiring the construction of additional eigenfunctions.

Before moving toward the body of this work, we introduce some notation to make the expressions in this chapter more readable.

Notation

Multi-index notation: Given a sequence, or vector $\chi = \begin{bmatrix} \chi_1 & \chi_2 & \cdots & \chi_{N_g} \end{bmatrix}^\mathsf{T}$, we define the multi-index $\varsigma = (\varsigma_i)_{i=i}^{N_g}$ such that

$$\chi^\varsigma = \chi_1^{\varsigma_1}\chi_2^{\varsigma_2}\cdots\chi_{N_g}^{\varsigma_{N_g}}.$$

We define the norm of this multi-index as

$$|\varsigma| = \sum_{i=1}^{N_g} \varsigma_i$$

4-1 Polynomial output

A finite set of Koopman operator eigenfunctions does not necessarily have the underlying states in their linear span. It is unclear whether a finite set of eigenfunctions in Example 2-3.2 will lead to the ability to reconstruct the underlying state x from a linear combination of lifted coordinates. However, under certain conditions on the set of eigenfunctions φ_i, there should exist an inverse map H to reconstruct the underlying state x from the eigenfunctions,

$$x = h(\varphi_1(x), \varphi_2(x), \ldots, \varphi_{N_g}).$$

In the context of system identification, we have no prior knowledge on this map. However, we can attempt to approximate it from data. To expand on this, we start by considering autonomous linear dynamics with a smooth nonlinear output h,

$$\begin{aligned} \chi[k+1] &= A\chi[k] \\ x[k] &= h(\chi[k]). \end{aligned} \tag{4-1}$$

In the Koopman operator context, this is the linear evolution of the lifted state variable $\chi \in \mathcal{G}(\mathcal{X})$, with a map h to reconstruct the underlying state.

The idea is to approximate the output map h with a polynomial p instead of the linear estimate in existing works. We can write this polynomial in multi-index notation, with sequences of indices ς_i and ς_i' and coefficients c_i,

$$h(\chi[k]) \approx p(\chi[k]) = \sum_{i=0}^{\infty} c_i \chi^{\varsigma_i}[k](\chi^*)^{\varsigma_i'}[k]. \tag{4-2}$$

The existence of this map p is a consequence of the Stone-Weierstrass theorem. The Stone-Weierstrass theorem is typically only stated for single-variable polynomials. However, starting from the generalised formulation presented in Theorem 4.1, we can show that the multivariable polynomials can approximate any complex-valued continuous function.

Theorem 4.1 (Stone-Weierstrass [26]). *Let K be a compact set and suppose Y is a subspace of $C(K)$ such that*

1. *$1 \in Y$;*
2. *$g, h \in Y$ implies $gh \in Y$;*
3. *$g \in Y$ implies $\overline{g} \in Y$;*
4. *Y separates points in K. That is, for any two points $x, y \in K$ with $x \neq y$, there exists an $f \in Y$ such that $f(x) \neq f(y)$.*

Then Y is dense in $C(K)$.

The abstract formulation in Theorem 4.1 does not directly correlate to the desired properties in the present work. The idea is to identify K with the complex-valued lifted coordinates χ in Equation (4-1) and Y with the space of multi-variable polynomials. The density if Y in $C(K)$ then implies that we can approximate any continuous h arbitrarily well. If we write for the functions $\chi_i : \mathbb{C}^n \to \mathbb{C}$ to denote the i'th element of the lifted state, $\chi_i(x) \mapsto x_i$, we can formally state the result in Proposition 4.1.

Proposition 4.1 (Stone-Weierstrass for multivariable functions). The set of all multivariable polynomials

$$Y = \left\{ \sum_{i \geq 0} c_i \chi^{\varsigma_i} (\chi^*)^{\varsigma_i'} : c_{i,j} \in \mathbb{C} \right\}.$$

is dense in the complex-valued continuous functions $C(\mathbb{C}^n)$.

Proof. We let $K = \mathbb{C}^n$, then

1. $1 \in Y$, with $c_0 = 1$, $\varsigma_0 = \varsigma_0' = 0$ and $c_i = 0$ for $i \geq 0$.
2. The product of two multivariable polynomials is again a multivariable polynomial
$$\left(\sum_{i \geq 0} a_i \chi^{\varsigma_i} (\chi^*)^{\varsigma_i'} \right) \left(\sum_{i \geq 0} b_i \chi^{\nu_i} (\chi^*)^{\nu_i'} \right) = \sum_{i \geq 0} \left(\sum_{k+j=i} a_k b_j \right) \chi^{\varsigma_i + \nu_i} (\chi^*)^{\varsigma_i' + \nu_i'}$$
3. $g \in Y$ implies $\overline{g} \in Y$ by setting $\varsigma \mapsto \varsigma'$, $\varsigma' \mapsto \varsigma$ and $c_i \mapsto c_i^*$.
4. Given $x, y \in \mathbb{C}^n$ with $x \neq y$. Let i be the index of the element in the vectors such that $x_i \neq y_i$. The polynomial $p(x) = x_i$ then satisfies $p(x) \neq p(y)$ if $x \neq y$.

Thus by Theorem 4.1 we conclude that Y is dense in $C(\mathbb{C}^n)$. □

Thus, given a map h, there exists some polynomial p such that $h \approx p$. Two questions remain,

1. Does the application of Dynamic Mode Decomposition (DMD) result in unbiased estimates for the eigenvalues λ?
2. How can we apply these results to the model produced by eigenfunction construction?

These questions are investigated in the next two subsections.

Estimation of eigenvalues

The convergence of Koopman modes computed by DMD applied to Hankel matrices was previously investigated by Arbabi and Mezić [45]. This suggests that the application of DMD should result in unbiased estimates for the Koopman eigenvalues in the limit of an infinite Hankel delay $s \to \infty$. In this section we investigate their original results in our context.

Proposition 4.2 (Convergence of the Hankel-DMD algorithm inside the basin of attraction [45])**.** On an ergodic attractor with basin of attraction B, we consider the Koopman operator acting on the space of square-integrable functions, with continuous observable $f \in L_2(\mathcal{X})$. For almost all initial conditions $x[0] \in B$ the eigenvalues of the DMD estimate converge to the Koopman operator eigenvalues as $s \to \infty$.

This result only holds in the limit $s \to \infty$ of the Hankel matrices, suggesting that the identified eigenvalues are biased in the estimation. However, the result in Proposition 4.2 is stated for the Koopman operator acting on $L_2(\mathcal{X})$, for which the Koopman operator has a much more intricate spectrum [27]. In the present work we consider the Koopman operator on the space of continuous functions $C(K)$, for which we have shown that almost any complex number $|\lambda| \leq 1$ is a Koopman eigenvalue with associated eigenfunction. Thus whilst the estimation of the true eigenvalues might be biased, this does not pose a problem to the proposed method, although the predictive performance of the identified models might be limited.

The ergodicity of in Proposition 4.2 implies that the system to be identified explores the state space $\mathcal{X}$ sufficiently. This is mostly a technical requirement, and can be resolved by using multiple trajectories instead [45].

Derived linear model

The linear model presented in Equation (4-1), combined with the polynomial expression allows us to bring the polynomial output into the dynamics of the linear model through an expansion of the underlying state-space. This follows from the properties presented in Proposition 2.1, making each term in the output polynomial p an Koopman eigenfunction with known eigenvalue. Thus given an initial estimate of Koopman operator eigenvalues and eigenfunctions, constructed from the method presented in Chapter 3, we can estimate the output map p by augmenting the states with eigenfunction products. We use the term *derived* states for these eigenfunction products. This allows us to bring the lifted dynamics in Equation (4-1) into a model of the form

$$\begin{aligned} \chi'[k+1] &= A\chi'[k] \\ x[k] &= Cx. \end{aligned}$$

Where we use the notation χ' to denote the augmented state. The matrix C now consists of the coefficients of the polynomial p in Equation (4-2).

We define the lattice set of order d to construct the derived Koopman modes from products of eigenfunctions and eigenvalues,

$$\Lambda_d(\lambda) = \left\{ \lambda^{\varsigma}(\lambda^*)^{\varsigma'} : \varsigma_i, \varsigma_i' \in \mathbb{N}, \sum_{i=1}^{N_g} \varsigma_i + \varsigma_i' \leq d \right\}$$

We can now define the derived Koopman model of a given order.

Definition 4.1 (Derived Koopman model)**.** Assume a linear model of Koopman eigenfunctions of the form in Equation (4-1) whose state $\chi \in \mathbb{C}^{N_g}$ consists of Koopman modes $(\lambda_i^\tau, \varphi_i)$ associated to state elements χ_i.
We define the derived Koopman model of order d as the model constructed from the Koopman modes with multi-indices ς_i,

$$\varphi^{\varsigma_i} = \varphi^{\varsigma_i}(\varphi^*)^{\varsigma_i'}.$$
$$(\lambda^\tau)^{\varsigma_i} = (\lambda^\tau)^{\varsigma_i}(\lambda^{\tau *})^{\varsigma_i'}.$$

Where ς_i iterates over all multi-indices satisfying $1 < |\varsigma_i + \varsigma_i'| \leq d$. The matrix A' is then constructed

$$A' = \text{diag}\left(\begin{bmatrix} 1 & \lambda_1^\tau & \lambda_2^\tau & \cdots & \lambda_{N_g}^\tau & (\lambda^\tau)^{\varsigma_1} & (\lambda^\tau)^{\varsigma_2} & \cdots & (\lambda^\tau)^{\varsigma_\ell} \end{bmatrix}\right).$$

This yields a complete model of the form,

$$\begin{aligned} \chi'[k+1] &= A'\chi'[k] \\ \chi'[k] &= \begin{bmatrix} 1 & \varphi_1(x[k]) & \varphi_2(x[k]) & \cdots & \lambda_{N_g}^\tau & \varphi^{\varsigma_1}(x[k]) & \varphi^{\varsigma_2}(x[k]) & \cdots & \varphi^{\varsigma_\ell} \end{bmatrix}^\mathsf{T} \\ x[k] &= C\chi'[k] \end{aligned}$$

We denote the dimension of this model as N'.

Derived state dimension

The state dimension of the derived system grows quickly with the number of states N_g in the original system and the order of the included polynomial. Suppose we have χ_1, χ_2 and χ_3, with the goal of constructing a polynomial of degree 2. Then, a sequence of exponents j_1, j_2 and j_3 should satisfy

$$j_1 + j_2 + j_3 \leq 2.$$

That is, if we assume that the state dimension N_g is larger than the desired polynomial degree d, we partition d into N_g parts, with the inclusion of 0. We denote this number by $p_d(N_g)$. A detailed investigation is beyond the scope of the present work. To illustrate the quick growth of $p_d(N_g)$ we present the $p_3(N_g)$ and $p_5(N_g)$ in Figure 4-1.

As a consequence we are severely limited in the number of terms in the polynomial estimate of the output map due to the exponential growth in additional dynamics. Considering Figure 4-1, we can expect to include only a small number of states in the derived state dimension.

4-2 Bilinear models

The motivation for constructing bilinear models is twofold. For one, the results on the generator of the Koopman operator in Section 2-6 suggest that bilinear models are more fit for

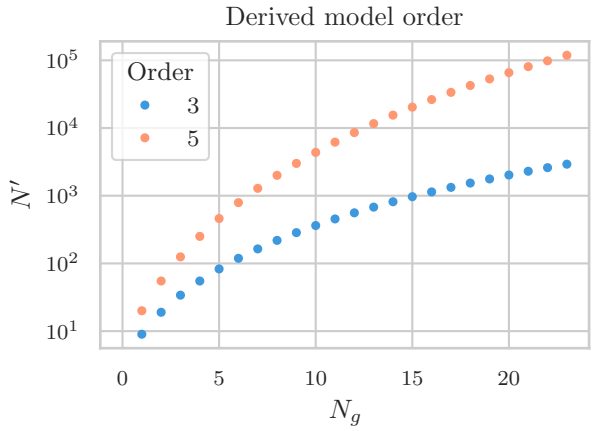

Figure 4-1: The state dimension N' for derived models of order 3 and 5 in terms of an initial state dimension N_g

the estimation of the forcing dynamics. Second, the inclusion of the derived Koopman modes in the dynamics, requires the introduction of bilinearity into the dynamics. The inclusion of forcing dynamics can be shown by considering the dynamics of a single forced state x,

$$x[k+1] = x[k] + u[k].$$

The squared dynamics are then given by

$$x^2[k+1] = x^2[k] + 2x[k]u[k] + u^2[k].$$

We conclude that the inclusion of bilinear terms and squared control signal u^2 should result in more accurate models. We write the following for a bilinear model in lifted coordinates χ

$$\begin{aligned}\chi[k+1] &= A\chi[k] + Bu[k] + \sum_{i=1}^{p} u_i[k]B_i\chi[k] \\ y[k] &= C\chi[k]\end{aligned}$$

In the following, we assume that the matrices A and C are known, constructed by the methods discussed in the previous section. Furthermore, we assume the availability of a lifting map, relating $\chi[k]$ to the true state $x[k]$ for each time instance k. The goal is now to construct a least-squares optimisation problem for the matrices B and B_i. We start by relating the measurement at time instance k in terms of an initial state and subsequent inputs.

$$y[k] = CA^k\chi[0] + \sum_{j=0}^{N-1}\left[CA^{N-j-1}Bu[j] + \sum_{i=1}^{p} CA^{N-j-1}B_i\chi[j]u[j]\right]$$

By vectorizing the matrices B and B_i, we can formulate a linear data equation. For this purpose we define two matrices,

$$\mathcal{C}_B = \begin{bmatrix} u^\intercal[0]\otimes C \\ u^\intercal[0]\otimes CA + u^\intercal[1]\otimes C \\ \vdots \\ \sum_{k=0}^{N-1} u^\intercal[k]\otimes CA^{N-k-1}\end{bmatrix}, \qquad \mathcal{C}_{B_i} = \begin{bmatrix} (u_i[0]\chi[0])^\intercal\otimes C \\ (u_i[0]\chi[0])^\intercal\otimes CA + (u_i[1]\chi[1])^\intercal\otimes C \\ \vdots \\ \sum_{k=0}^{N-1}(u_i[k]\chi[k])^\intercal[k]\otimes CA^{N-k-1}\end{bmatrix}. \tag{4-3}$$

Now, the data equation becomes

$$\underbrace{\begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[N] \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}}_{\mathcal{O}_A} \chi[0] + \mathcal{C}_B \text{vec}(B) + \sum_{i=1}^{p} \mathcal{C}_{B_i} \text{vec}(B_i). \tag{4-4}$$

By performing Koopman eigenfunction construction on the autonomous dynamics we have previously derived the matrices C and A, and a function to construct $\chi[k]$ from measurements $y[k]$. Consequently, the only unknowns in Equation (4-4) are the forcing gains B and B_i. To find the gains B and B_i we can solve a least squares problem. To collect terms, we define

$$\Theta_{\mathcal{C}} = \begin{bmatrix} \mathcal{C}_B & \mathcal{C}_{B_1} & \cdots & \mathcal{C}_{B_p} \end{bmatrix}, \qquad \Theta_B = \begin{bmatrix} \text{vec}(B)^\intercal & \text{vec}(B_1)^\intercal & \cdots & \text{vec}(B_p)^\intercal \end{bmatrix}^\intercal$$

Then, we can formulate the least squares problem

$$\min_{\Theta_B} \|Y - \mathcal{O}_A \chi[0] + \Theta_{\mathcal{C}} \Theta_B\|_2 . \tag{4-5}$$

Where $\Theta_B \in \mathbb{C}^{(N_g+1)p \times N_g}$. The solution for B and B_i is then given by

$$\begin{bmatrix} B & B_1 & \cdots & B_p \end{bmatrix} = \text{vec}^{-1} \left(\Theta_{\mathcal{C}}^{\dagger} \left(Y - \mathcal{O}_A \chi[0] \right) \right)$$

By combining the bilinear model with the aforementioned state extension, the reduced dynamics can be seen as a model of the form,

$$\chi[k+1] = A\chi[k] + \sum_{i=1}^{p} p_i(\chi[k]) u_i[k].$$

The bilinear gain on input i in the extended systems is then effectively a polynomial gain $p_i(\chi[k])$. The polynomial p_i aims to approximate the terms $(\nabla g)^\intercal B$ or $(\nabla g)^\intercal \eta$ from the examples in Section 2-6 in the context of systems that are linear in control or control-affine respectively. This comes at the cost of computing additional gains on products of eigenfunctions.

State-by-state optimisation

The memory requirements op the problem in Equation (4-5) quickly grows in size. For a lifted state dimension of N' we find that each $\mathcal{C}_{B_i}$ is of size $NN' \times N'^2$. With derived states, N' grows extremely quickly, making the problem size infeasible to be solved on a normal computer. As an example, if we take a derived state dimension of $N' = 100$ with a training data size of $N = 2000$, a computer would need on the order of 10^2 gigabyte of memory to store $\mathcal{C}_{B_i}$.

Instead we propose to solve the problem in Equation (4-5) row-by-row, or equivalently, state-by-state. This is possible because A is diagonal, as was illustrated in Equation (2-10), and we know the lifted state χ at each time step k.

Table 4-1: Parameters used for the identification of derived bilinear models.

	Toy model	Duffing oscillator	Van der Pol oscillator
Initial order N_g	3	5	15
Polynomial order d	2	4	2
Derived order N'	9	125	152
Time step τ (s)	0.1	0.2	0.2
Sequences	50	50	50
Sequence length N	2000	2000	2000

The observability matrix $\mathcal{O}_A$ then becomes a single-column Vandermonde matrix, and the kronecker product terms in the matrices $\mathcal{C}_B$ and $\mathcal{C}_{B_i}$ disappears,

$$\underbrace{\begin{bmatrix}\chi_j[1]\\ \chi_j[2]\\ \vdots\\ \chi_j[N]\end{bmatrix}}_{\chi_j} = \underbrace{\begin{bmatrix}\lambda_j\\ \lambda_j^2\\ \vdots\\ \lambda_j^N\end{bmatrix}}_{\mathcal{O}_{\lambda_j}} \chi[0] + \begin{bmatrix}u^\intercal[0]\\ u^\intercal[0]\lambda_j + u^\intercal[1]\\ \vdots\\ \sum_{k=0}^{N-1} u^\intercal[k]\lambda_j^{N-k-1}\end{bmatrix} \mathrm{vec}(B)$$
$$+ \sum_{i=1}^{p} \begin{bmatrix}u_i[0]\chi^\intercal[0]\\ u_i[0]\chi^\intercal[0]\lambda_j + u_i[1]\chi^\intercal[1]\\ \vdots\\ \sum_{k=0}^{N-1} u_i[k]\chi^\intercal[k]\lambda_j^{N-k-1}\end{bmatrix} \mathrm{vec}(B_i)$$

The optimisation problem then has a reduced form that can be solved seperately for each j. We use the same notation introduced in the previous section, with the matrix $C = 1$ in the matrices in Equation (4-3)

$$\min \|\chi_j - \mathcal{O}_{\lambda_j}\chi_j[0] + \Theta_C\Theta_B\| \tag{4-6}$$

4-3 Numerical study

Setup

We estimate the eigenvalues of the Koopman dynamics using the DMD modes derived from Hankel matrices, as derived in Chapter 3. To illustrate the expanded state space, we use the same system order derived in Section 3-3, taking the elbow point for the parameter s in the noise-free case, except for the Van der Pol oscillator, for which it was found that a system order of 15 was needed to recover the periodic dynamics. All the problem parameters are summarised in Table 4-1, remaining mostly identical to the parameters used in Section 3-3.

Autonomous dynamics

The autonomous response of the identified polynomial models are presented in Figure 4-2. All models were able to predict the correct response from a given initial condition. Comparing the variance accounted for (VAF) of the polynomial models in Table 4-2 to the previously obtained results in Table 3-3, we find slightly improved results, while simultaneously working with fewer eigenfunctions.

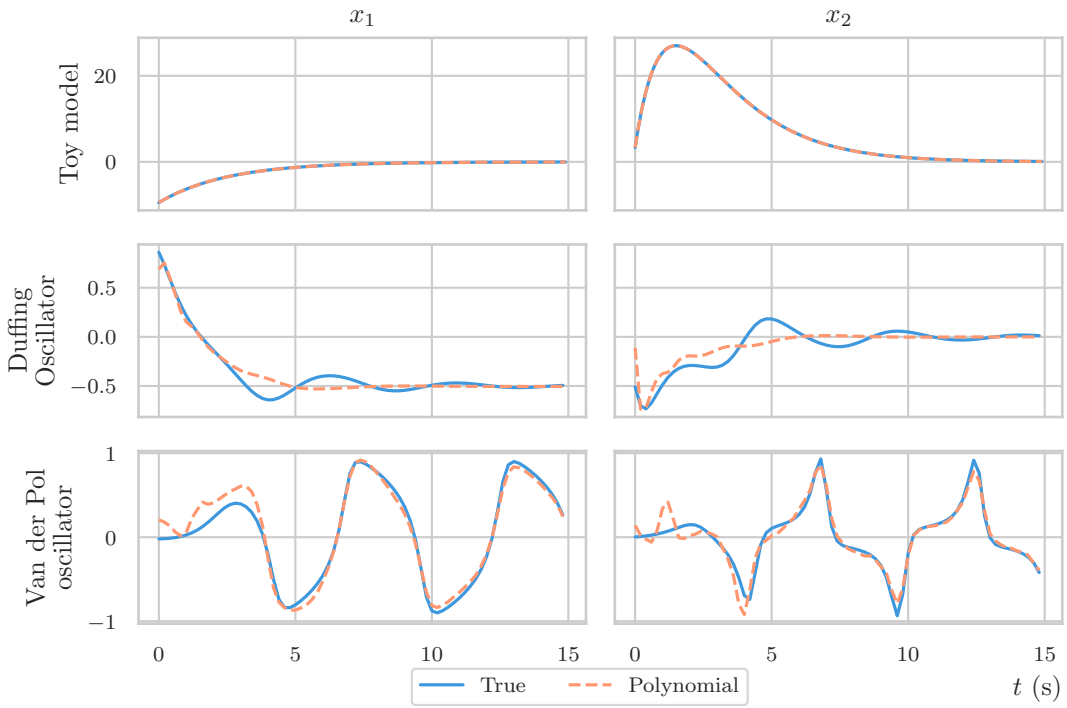

Figure 4-2: The autonomous response to an initial condition of the Toy model, Duffing oscillator and Van der Pol oscillator, compared to the identified model dynamics.

Most notably, the Duffing oscillator has a better VAF than the previously obtained models. The introduction of the eigenvalue $\lambda = 1$ allows the model to precondition an offset for the attractor on the presented domain. Note however, similar to the previously identified models, expansion of the subspace $\mathcal{X}$ causes the identification process to fail.

Table 4-2: Average VAF and standard deviation of the identified autonomous response to 25 initial conditions of length 2500 for the Toy model, Duffing oscillator and Van der Pol oscillator.

State	Toy model	Duffing Oscillator	Van der Pol oscillator
x_1	99 ± 10	90 ± 15	97 ± 2
x_2	100 ± 2	83 ± 8	93 ± 6

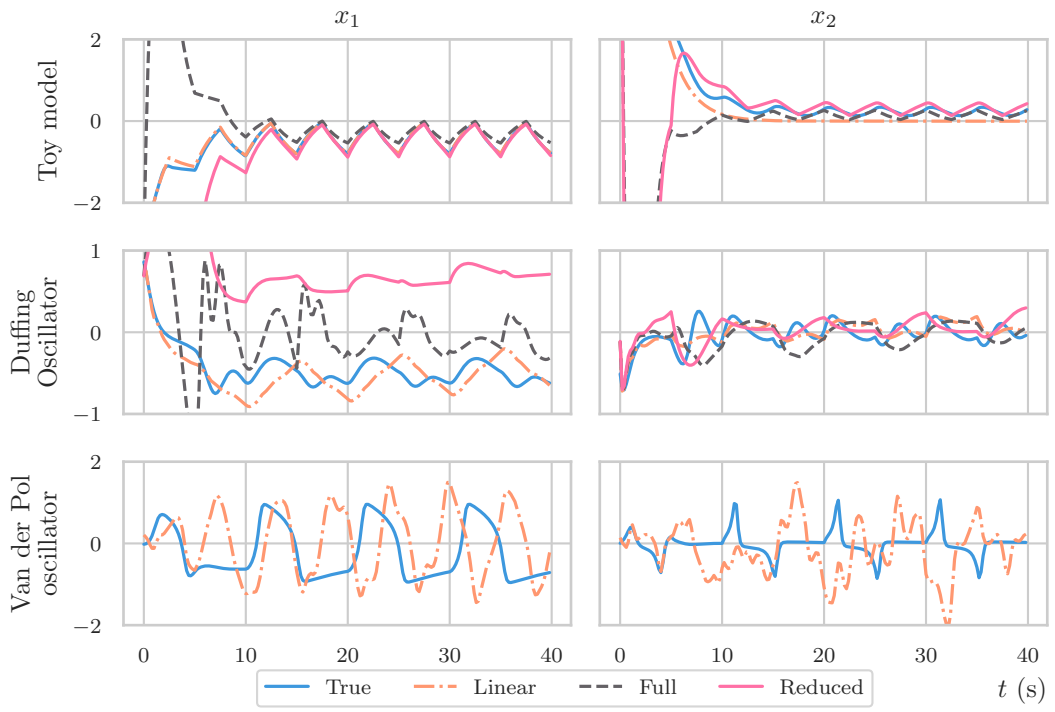

Figure 4-3: The forced response of the Toy model, Duffing oscillator and Van der Pol oscillator to a unit square-wave forcing with a period of 5 s. Models with polynomial states are derived with both linear, and bilinear forcing terms. The model labeled "Full" was identified with an additional forcing a squared forcing signal $u^2[k]$. The bilinear models for the Van der Pol oscillator did not yield stable dynamics, and were left out of the figure for readability.

Bilinear dynamics

In Figure 4-3 the response to a square wave forcing of the identified models is presented. Almost all bilinear models fail to capture the dynamics induced by the square-wave . Furthermore, the linear models appear to yield mostly better predictive performance than the bilinear models. For the Van der Pol oscillator, the presented method failed to get a good estimate of the dynamics. Therefore, these dynamics were intentially left out from Figure 4-3 for readability.

Once the periodic behaviour for the Toy model dynamics have been reached, the bilinear models perform better than the linear model, being able to model both the directly influenced dynamics in x_1 and the coupling in x_2. This coupling cannot be modeled with a linear system, as illustrated in the introduction of the present chapter.

The key reason for the poor performance of the bilinear models is due to the requirement of using the lifted state estimate $\chi[k]$ in the state estimation problem. However, when we look at the residual of the state reconstruction,

$$\varepsilon = \|C\varphi(x) - x\|,$$

we find that the residual is sizable for each of the model. The residual is presented in Figure 4-4 on the subset of the state space for which the models were trained.

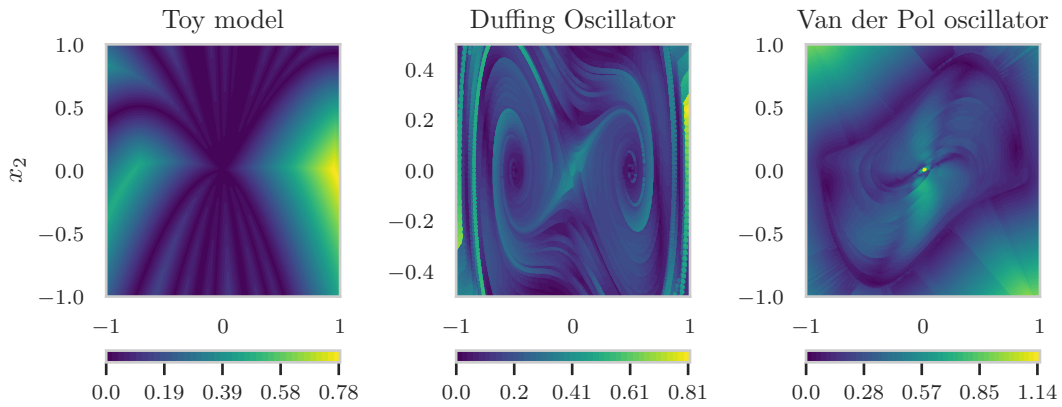

Figure 4-4: State reconstruction error $\|C\varphi(x) - x\|$ for the Toy model, Duffing oscillator and Van der Pol oscillator on the region identified by the dynamics.

The state reconstruction error is relatively large for the region considered. Consequently, a lifted state $\varphi(x)$ has a poor correlation to the underlying state x. The use of these coordinates in the optimisation problem in Equation (4-6) causes the optimal solution to be partially optimised towards this error instead.

The worse initial state response in Figure 4-3 can also be explained by the state reconstruction error in Figure 4-4. The state reconstruction error is only small around the fixed points for the Toy model and Duffing oscillator, or around the periodic dynamics for the Van der Pol oscillator. As a consequence, the forced dynamics away from these equilibria or attractors has a comparatively large effect on the reconstructed state dynamics.

The poor state reconstruction error suggests that the addition of polynomial terms does not introduce enough degrees of freedom to get a good output map estimation. This is mostly due to the limited number of states that can be introduced due to the exponential growth of the underlying state dimension. Smaller initial state orders were attempted in favour of a larger derived state order. However, it was found that this yielded worse overall performance.

4-4 Conclusion

We presented a method to augment a linear Koopman model with a nonlinear output map, approximated as a multivariable polynomial. This polynomial can be included in the underlying state dynamics to obtain an augmented linear model for autonomous dynamics. Combining this derived model with the method of eigenfunction construction presented in Chapter 3 results in a slightly improvemed and more robust VAF for the Toy model and Van der Pol oscillator, the derived model for the Duffing oscillator outperforms the original results by Korda and Mezić. We conclude that the estimation performance of the proposed systems can be improved by inclusion of a nonlinear output map.

The inclusion of derived dynamics proposes the use of a bilinear model to estimate the forced dynamics. By exploiting the structure in the underlying problem, the complexity of the optimisation problem can be reduced to a small least-squares problem. The dynamics under

influence of a square-wave forcing of the Van der Pol oscillator and Duffing oscillator are unable to be reconstructed by the proposed model. The dynamics of the Toy model are captured around the fixed point, but at the cost of a large initial state error.

A state reconstruction error of a similar order of magnitude of the state itself suggests that the proposed models grow too quickly in size to capture an accurate low-dimensional representation of the Koopman operator for models. We conclude that, for the proposed methods, a bilinear model can improve the performance of eigenfunction construction, given that the state reconstruction error is small enough to ensure that the optimisation problem for the derivation of a bilinear system is well-conditioned.

Chapter 5

Neural networks for eigenfunction construction

To further investigate the limitations of eigenfunction construction in application to nonlinear systems we construct a neural network model for the learning of Koopman eigenfunctions. This has been previously attempted in works by Lusch et al., which successfully learned reduced linear representations for the Koopman operator [35]. In the present work we introduce a different optimisation problem, aimed at multi-step prediction. The presented network model lends itself to learning eigenvalues and eigenfunctions of the Koopman operator, allowing is to do a more detailed comparison into the eigenfunctions constructed in the previous chapters.

5-1 Network Architecture

We want the Koopman network architecture to reflect the eigenvalue evolution constructed in previous sections. Consequently, we want the network to reflect the dynamics of a system of the form

$$\begin{aligned} \chi[k+1] &= A\chi[k] \\ x[k] &= \varphi^{-1}(\chi[k]) \\ \chi[k] &= \begin{bmatrix} \varphi_1(x[k]) & \varphi_2(x[k]) & \cdots & \varphi_{N_g}(x[k]) \end{bmatrix}^\mathsf{T}. \end{aligned} \tag{5-1}$$

Here, A is a diagonal matrix containing the eigenvalues of eigenfunctions φ, represented as a lifted state χ. Instead of the previously considered linear output maps, we now consider an arbitrary nonlinear map h for output reconstruction.

The proposed network architecture is presented in Figure 5-1, which represents each of the components in Equation (5-1) a seperate block. This model allows to do a k-step ahead prediction by repeatedly applying the eigenvalue gain in the eigenvalue block A. This network is effectively an auto-encoder, constructing a latent space of lifted coordinates χ, where the

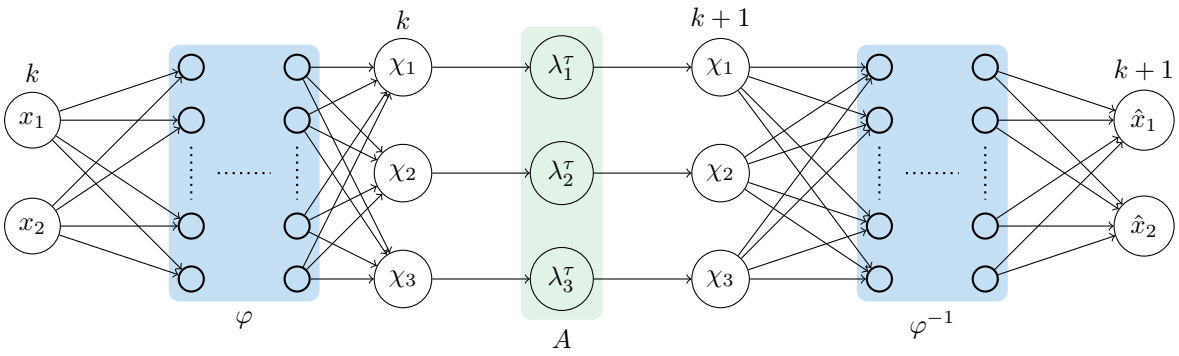

Figure 5-1: Proposed network architecture, consisting of a multilayer linear eigenfunction block φ, the linear evolution A and a multilayer linear output map φ^{-1}. The block structure represents the format of a linear system, where the time step is indicated above the states x and lifted states χ.

dynamics is linear. The blocks φ and φ^{-1} in Figure 5-1 consist of small neural networks with 2 hidden layers of 20 hidden nodes each, with a split hyperbolic tangent activation function, defined in the next subsection.

Complex-valued networks

Eigenvalues of the Koopman operator are inherently complex-valued, making the eigenfunctions and output map complex-valued as well. Consequently, our network needs to be capable of learning complex-valued parameters. There is relatively limited research toward the properties and uses of complex-valued networks. We elaborate on the choice of activation function for learning linear complex-valued neural networks [46].

Not all classic activation functions are applicable to complex-valued networks. For example, the Rectified linear unit (ReLU),

$$\mathrm{ReLU}(x) = \max\{x, 0\}$$

does not have a complex extension due to lack of ordering in the complex numbers. In addition, a notion of differentiability is required, quickly moving toward holomorphic functions. Both the complex-valued holomorphic hyperbolic tangent and the split hyperbolic tangent were tested in our context. It was found that the holomorphic hyperbolic tangent failed to converge, while the split hyperbolic tangent stanh : $\mathbb{C} \to \mathbb{C}$,

$$\mathrm{stanh}(z) = \tanh(\mathrm{Re}(z)) + i\tanh(\mathrm{Im}(z)),$$

did converge. Although not holomorphic, this function is differentiable with respect to the real and complex part separately, which is sufficient for a convergent network [46].

Learning problem

The learning problem considers two seperate learning objectives,

1. State reconstruction: $h(\varphi(y[k])) = y[k]$,
2. Prediction: $h(A^j\varphi(y[k])) = y[k+j]$ for $j = 1, 2, \cdots$.

Note that state reconstruction is effectively a special case of the prediction step with $j = 0$.

The complete loss function can then be defined as an optimisation problem in terms of the network parameters $\theta \in \Theta$,

$$\min_{\theta\in\Theta} \sum_{j=0}^{N} \left\| y[k+j] - h\left(A^j\varphi\left(y[k]\right)\right)\right\|_2 + \alpha_1 \sum_{j=1}^{N_g} \|\lambda_j\|_1 + \alpha_2\|\theta\|_2 \tag{5-2}$$

For the optimisation problem, ℓ_2 regularisation is employed on the network parameters and ℓ_1 regularisation on the eigenvalues Λ. This drives any unnecessary eigenvalues to 0, constructing a minimal description for the neural network model.

Inspired by the work by Verhoek et al., we consider a multi-step optimisation problem [47] where the prediction horizon N is increased over the epochs. The optimisation problem in Equation (5-2) is solved for a batch size of 256 trajectories, making the optimisation procedure significantly more stable. Furthermore, the eigenvalues are kept fixed until the model has learned stable dynamics up to a horizon of $N = 10$. The network was implemented in Pytorch, using the Adam optimiser with default parameters and a variable learning rate. All hyperparameters used for the learning of the Koopman network are presented in Table 5-1.

Table 5-1: Hyperparameters for the learning of Koopman eigenfunctions.

Group	Parameter	Value
Architecture	Hidden layers φ	2
	Hidden size φ	20
	Hidden layers φ^{-1}	2
	Hidden size φ^{-1}	20
Learning problem	Learning rate	10^{-3}
	α_1	10^{-7}
	α_2	10^{-8}
	Batch size	256

The model parameters are taken at a similar order of magnitude as originally presented in Chapter 3, taken at the slightly modified values in Table 5-2.

Table 5-2: Model hyperparameters for the learning of Koopman eigenfunctions.

Parameter	Toy model	Duffing oscillator	Van der Pol oscillator
Time-step	0.1	0.1	0.1
Lifted dimension N_g	3	5	15

5-2 Results

The presented networks successfully converged, enabling the prediction of nonlinear dynamics. We present the validation loss after 34 epochs of 700 iterations in Table 5-3. The training took approximately 1 hour on a NVIDIA GeForce GTX970.

Table 5-3: Validation loss after 34 epochs of 700 iterations for the proposed network architecture applied to the Toy model, Duffing oscillator and Van der Pol oscillator

Parameter	Toy model	Duffing oscillator	Van der Pol oscillator
Loss	2.3×10^{-4}	2×10^{-2}	5.3×10^{-4}

The network architecture with the proposed hyperparamaters is best adapted to learning the dynamics of the Toy model. This is expected, since the Toy model can be approximated well by a linear system, as we have seen in the results of the previous chapters.

Dynamics

The small loss function in Table 5-3 suggest that the models are able to accurately capture the dynamics in the system. In Figure 5-2 we present the multi-step prediction performance of the learned networks, at longer timescales than the model was learned for. The dynamics remaining stable for $N > 100$ suggests that the lifted space can be interpreted as set of eigenfunctions.

The performance of these models can be made arbitrarily well by tuning the hyperparameters of the learning problem. The performance under the relatively small network suggests that this work can be extended for further study. However, the primary goal of the present chapter is to compare the learned eigenfunctions with the eigenfunctions constructed in Chapter 3.

Interestingly, the dynamics for the Duffing oscillator deviate significantly when compared to the dynamics for the Van der Pol oscillator. This suggests that the Duffing oscillator would benefit from using a larger lifted state dimension, allowing the inclusion of additional eigenvalues.

Comparison to eigenfunction construction

We first compare the learned eigenvalues of the network to the eigenvalues identified by Dynamic Mode Decomposition (DMD), applied to Hankel matrices. These are plotted for the Toy model, Duffing oscillator and Van der Pol oscillator in Figure 5-3. Interestingly, the distribution of eigenvalues for the two methods are mostly different. The eigenvalues identified by the network are typically more concentrated around 1, with a smaller complex term. This suggests that the periodicity introduced by the complex eigenvalues is not exploited by the network. This can partially be derived from the relatively short-time scale for which the models were trained and the characteristics of the dynamics in the example models.

The Duffing oscillator and Toy model only have a small complex contribution in the eigenvalues, since their periodicity is not that pronounced in their unforced dynamics, which is also

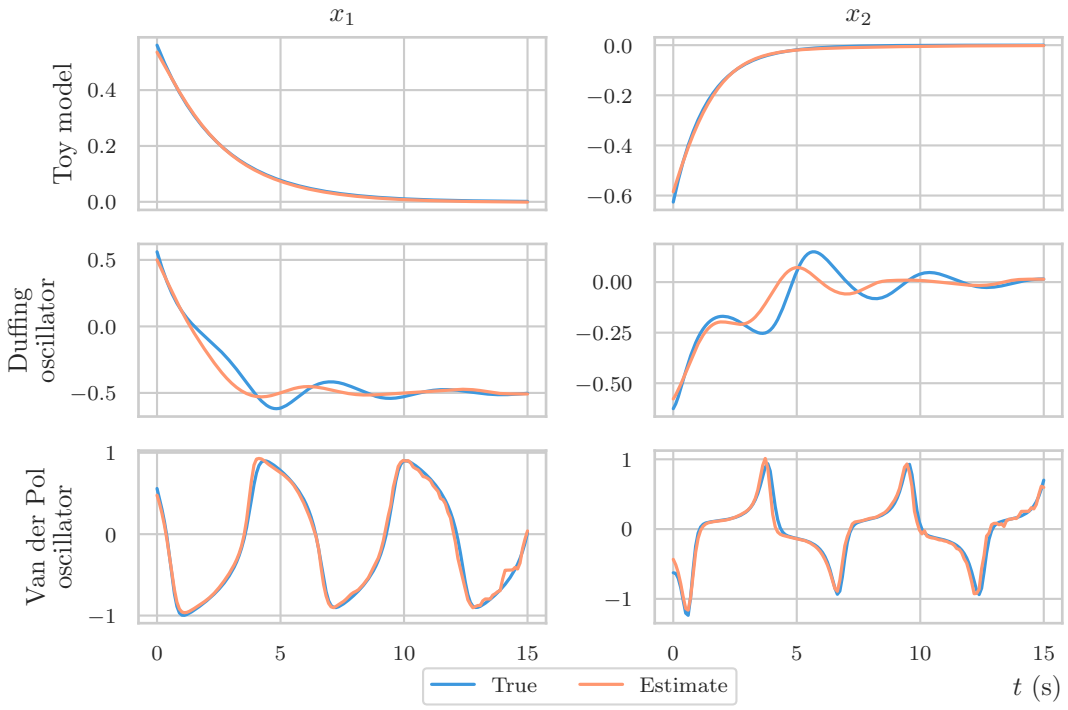

Figure 5-2: Dynamics estimated by the learned network for the Toy model, Duffing oscillator and Van der Pol oscillator, originating from a random initial condition.

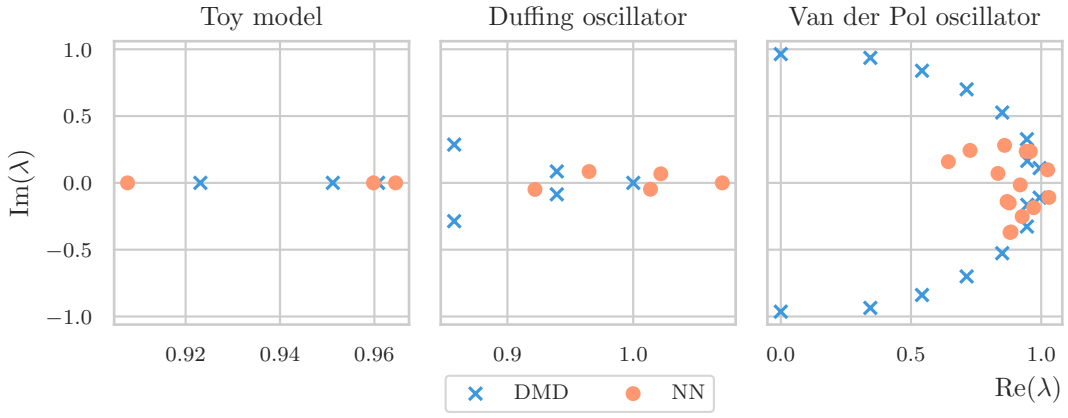

Figure 5-3: Eigenvalues estimated by the learned network, and the eigenvalues estimated by application of DMD to hankel matrices.

illustrated by the responses in Figure 5-2. Similarly, the 100 time-steps for the Van der Pol oscillator captures only 1 period of the limit cycle, limiting the required periodicity of the learned eigenfunctions. Unfortunately we were unable to increase the prediction horizon of the learned models further, due to hardware limitations. Again, a full study was beyond the scope of the present work.

In Figure 5-4 we study two learned eigenfunctions from the network for the Van der Pol oscillator and compare it to the eigenfunctions constructed from data. The associated eigenvalues

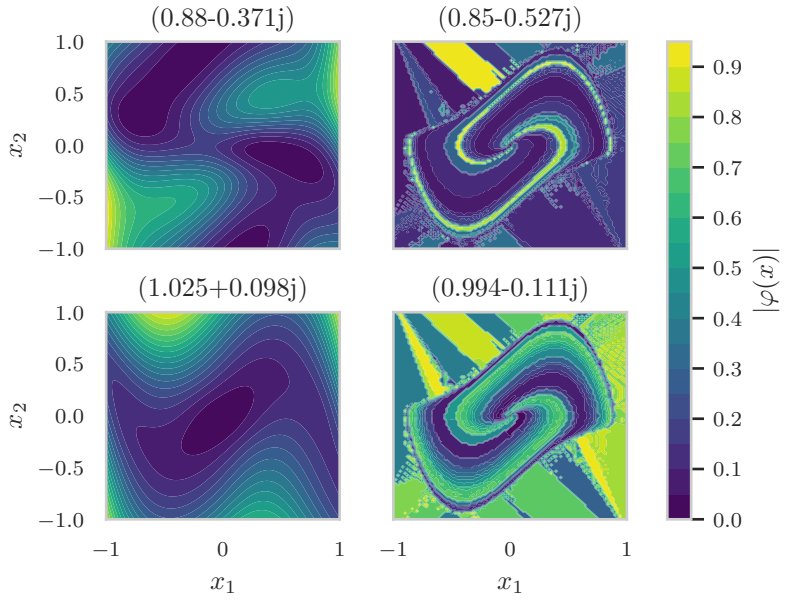

Figure 5-4: Norm of the eigenfunctions for the Van der Pol oscillator, learned by the network and the eigenfunctions constructed directly from data.

were chosen such that two close DMD estimates also existed. A clear limitation of the models constructed from the proposed eigenfunctions emerges. The choice of Γ appears to limit the functions that can be approximated, illustrated by the clear bands along the trajectories originating from Γ. Any initial condition g on the set Γ can be propagated through λ^τ, causing the pointwise norm of the initial condition along the given trajectory to satisfy,

$$|\lambda^\tau g| \leq |\lambda^\tau||g|.$$

As a result, the norm can only decay, or remain constant along trajectories of the system. However, the eigenfunctions learned by the network in Figure 5-4 suggests that periodicities in the propagated magnitude are required for eigenfunctions that accurately capture the dynamics in the system.

Chapter 6

Discussion and outlook

6-1 Discussion

The results presented in Chapter 3 suggests that the original method proposed by Korda and Mezić for models with eigenfunction construction is not suitable to construct accurate dynamics in its present form. Even when exploiting gradient descent for the derivation of eigenvalues, the constructed dynamics are unsatisfactory at time scales larger than discussed in the original work.

This breaks down into two main issues that culminated into the conclusions of the previous chapters; the underlying assumption for the existence of linear dynamics and the state reconstruction error. We discuss these issues seperately.

Assumption of linear dynamics

The assumption of linear dynamics is twofold. We first discuss the linear autonomous dynamics, which should not limit the accuracy of the proposed models. The assumption of the existence of a linear model in the form of Equation (3-3) lays at the foundation for the eigenvalue estimates. This suffers from the same problem discussed in the section on Proposition 4.2; Estimated Koopman eigenvalues will remain biased unless $s \to \infty$. This is less of a problem than it initially might seem, since we have shown in Section 2-5 that any number λ^τ with $|\lambda| \leq 1$ can be used to construct an eigenfunction for a stable system.

However, the assumption on a linear input gain B makes it difficult for the proposed methods to capture the forced dynamics. This is in line with the derivations in Section 2-6. This raises the question whether there are more efficient eigenfunctions when taking the forced dynamics into account, which we will discuss in the outlook.

State reconstruction

Initially the autonomous dynamics presented for both the original models and derived models show promising results. However, the state reconstruction error, following the results in

Section 4-3, sketches a different picture. The number of states required to obtain a small state reconstruction error appears to be extremely large for more complex systems. This is in line with results presented by a range of authors when Koopman methods are presented in the context of control [5, 29].

As long as the state reconstruction error is large, the construction of a bilinear model will remain difficult. One could consider identifying the bilinear model directly instead, proposed in the outlook. This bilinear model can then be used to construct eigenfunctions in the same way as in Section 2-5.

A linear model for nonlinear systems will never capture the full nonlinear dynamics. This error can be accurately quantified in the frequency domain, illustrating the limit of linear models clearly. Therefore, linear Koopman methods remain primarily an analysis tool for autonomous dynamics when applied to nonlinear systems.

System complexity

For the Duffing oscillator and Van der Pol oscillator clear problems can be formulated for the proposed method. This illustrates the difficulty in approximating their autonomous dynamics presented in Section 3-3.

The Duffing oscillator ultimately has three isolated fixed points in the autonomous dynamics. This can never be reconstructed by a linear model. A discrete-time linear model has either a unique fixed point at the origin, or an entire fixed subspace, associated to the eigenvalue 1. The subspace would emerge as an entire line on which the dynamics are in equilibrium. In autonomous dynamics, the proposed methods can initialise the dynamics at the origin. This allows the system to initialise one of the two stable equilibria, towards which the dynamics will converge explaining the reasonable performance. However, once a forcing is introduced, this initialisation breaks down.

The Van der Pol oscillator is suggested to break down due to the dynamics on different time-scales. The stiffness ratio captured this in the discussion in Section 3-3. In theory, this should not truly limit the proposed methods, since subspace methods can deal with multi-scale dynamics. The difficult mainly lies in capturing enough of the periodic dynamics, whilst also having a small enough time-scale for the large initial response. To solve this, one could consider doing multiple iterations of Dynamic Mode Decomposition (DMD) at two different time scales and matching them by scaling the eigenvalue accordingly. This allows eigenvalues of both time-scales to be captured without running into limitations for capturing both.

Dependence on hyperparameters

Throughout this study, tuning the hyperparameters for the identification processes was an intricate processes. Much of this culminated in the recommendations throughout the previous sections. This puts a limit on the interpretability of the presented results. In particular, variations on the initial state set Γ or variations in the chosen time step τ were found to significantly impact the performance of the presented methods in application to the Duffing oscillator and Van der Pol oscillator.

Note that these difficulties can partially be attributed to the aforementioned predictive limitations for these models, due to the difficulty of using linear techniques in identifying these models. A simple further study would investigate the effect of these parameters in more detail.

The results presented in Chapter 5 suggest that the choice of the initial condition set Γ might have a large effect on the capabilities of the model. Currently, there are only a limited set of requirements on the set Γ. Further work should focus on deriving results on the effects of a given set, in order for the derived method to be able to capture a broader range of eigenfunctions.

6-2 Outlook

We can make two larger recommendations for further study., which we present in two seperate sections. First, we already mentioned in Chapter 2 that full state measurements are typically not available. Although the methods discussed in the present work are still in their infancy, practical studies can be made to investigate the practical feasibility and identify points for further research.

Second, the two-step approach limits the performance of systems under the influence of forcing. If the end goal of a model is to design a control algorithm for the nonlinear system, the identification of the forced dynamics should inherently be part of the eigenvalue identification step.

6-2-1 Time delay embedding for Koopman operator models

Typically we do not have access to full measurements of the states associated to the dynamical system. This problem has not been investigated in detail up to this point. In the context of methods based on DMD authors often point to Takens' theorem without further study. However, in the context of controlled systems, the application of Takens' theorem is more intricate, as was illustrated in Section 2-1. Previously we discussed background literature on Takens' theorem in Theorem 2.1. In this section we apply the results by Stark to linear and bilinear models to derive embedding results for Koopman models of nonlinear systems. We start by repeating the full embedding results [22].

Theorem 6.1 (Takens embedding theorem for forced systems [22])**.** *Let $\mathcal{X}$ and $\mathcal{U}$ be compact manifolds of dimension n and p respectively. If $d \geq 2n+1$, and the set of periodic orbits with period smaller then d of $h \in D^r(\mathcal{U})$ has zero Lebesgue measure in $\mathcal{U}$. Then for $r \geq 1$ there exists a residual set of $(f, \phi) \in^r (\mathcal{X}) \times C^r(\mathcal{X}, \mathbb{R})$ such that for any (f, ϕ) in this set there is an open dense set of u of full Lebesgue measure such that the delay embedding map $\phi_{f,h} : \mathcal{X} \to \mathbb{R}^d$ is an embedding, defined by*

$$\phi_{f,h}(x) \coloneqq \begin{bmatrix} h(x) & h(f(x)) & h(f^2(x)) & \cdots & h(f^{d-1}(x)) \end{bmatrix}.$$

By Theorem 6.1, taking a time-delay embedding of the measurement variable y dimension $d \geq 2n+1$, there exists a diffeomorphism ϕ onto $\mathcal{X}$. This diffeomorphism can be used to

extend the eigenfunctions to output systems, by taking

$$\varphi' = \varphi \circ \phi^{-1} \tag{6-1}$$

as the lifting functions. Suppose we identify a model for a system with nonlinear dynamics,

$$\begin{aligned} x[k+1] &= f(x[k]) \\ y[k] &= h(x[k]). \end{aligned}$$

Then, by the developments in Chapter 3, we assume the existence of a model of the form

$$\begin{aligned} \chi[k+1] &= A\chi[k] \\ \chi[k] &= \begin{bmatrix} \varphi_1(x[k]) & \varphi_2(x[k]) & \cdots & \varphi_{N_g}(x[k]) \end{bmatrix}^\mathsf{T} \\ y[k] &= Cx[k]. \end{aligned}$$

Assuming that $h \in \text{span}\{\varphi_1, \varphi_2, \cdots, \varphi_{N_g}\}$. With the extension of the model in the previous chapter, we consider a projection onto the algebra of eigenfunctions.

Effectively we now have a linear autonomous model, which now allows us to extend the model to time-delay coordinates. We start by showing that eigenfunctions are preserved under the application of time-delay coordinates.

Proposition 6.1 (Delay embedding eigenfunctions)**.** Suppose φ is a Koopman operator eigenfunction with eigenvalue λ, and ϕ the delay embedding defined in Theorem 6.1. Then $\varphi \circ \phi^{-1}$ is also an eigenfunction with eigenvalue λ in embedded coordinates.

Proof. Suppose we have a Koopman operator eigenfunction φ with eigenvalue λ associated to a vector field f. Let ϕ be the delay embedding from Theorem 6.1. Then,

$$K_\tau \varphi \circ \phi^{-1} = \varphi \circ \phi^{-1} \circ F'_\tau = \varphi \circ F_\tau \circ \phi^{-1} = \lambda \varphi \circ \phi^{-1}$$

We conclude that $\varphi \circ \phi^{-1}$ is therefore an eigenfunction of the nonlinear system with output map. □

Using the notation of x' for time-delay coordinates we construct a complete model of the form

$$\begin{aligned} \chi[k+1] &= A\chi[k] \\ \chi[k] &= \begin{bmatrix} \varphi_1(x[k]) & \varphi_2(x[k]) & \cdots & \varphi_{N_g}(x[k]) \end{bmatrix}^\mathsf{T} \\ x'[k] &= C\chi[k] \\ y[k] &= x'_1[k] \end{aligned}$$

We conclude that the proposed method can be used to construct an autonomous model with time-delay coordinates x'.

The inclusion of a forcing term in the embedding ϕ makes the eigenfunctions φ in these coordinates dependent on the forcing u. Adapting the eigenfunctions to include a forcing u

removes the ability to construct these from an interpolation problem, requiring significant modifications to the developed methods.

However, under the assumption that a measurement is lifted at time t, where $u[k] = 0$ for $k < t$, the proposed lifting function construction can be applied. The lifted dynamics χ are independent of the use of time-delay coordinates. Therefore, given some initial state χ, constructed from unforced time-delay coordinates, the models can then be used to estimate the future dynamics with some known forcing u. In practice, these models would be probably be combined with some form of Kalman filter, making this a soft requirement.

A seperate study should be done on whether the proposed methods yield sufficient performance when using time-delay coordinates for forced dynamics. The poor performance of the forced models in Chapter 3 suggests that this is difficult, since the problem complexity becomes larger.

6-2-2 Estimating Koopman modes with forced dynamics

One of the core questions arising in the present work is whether the estimated eigenvalues λ^τ are sufficient to reconstruct the forced dynamics. The forcing might actuate dynamics that are not necessarily captured by the autonomous dynamics used for the identification of Koopman eigenvalues.

The model proposed by Korda and Mezić in Equation (2-10) is standard in linear system identification methods. Methods such as MOESP and N4SID can identify such models directly from forcing data [48, 49]. This suggests the application of these methods to the proposed problem. However, these methods assume a specific disturbance model,

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] + Ke[k] \\ y[k] &= Cx[k] + Du[k] + e[k]. \end{aligned}$$

Where $e[k]$ is a white-noise sequence. This requirement is needed to ensure that the estimate for A is unbiased [20]. Whenever we attempt to model a nonlinear system as a finite-dimensional linear system, the residual error $e[k]$ is highly correlated to the underlying state $x[k]$. Furthermore, the residual is also correlated to the forcing $u[k]$ because the forced Koopman dynamics do not lend themselves to a linear system, as illustrated in Section 2-6. So even in the limit of infinite state dimension the residual will be correlated to the state and forcing. Consequently, eigenvalue estimates produced by MOESP and N4SID methods will be severly biased, causing the autonomous dynamics to be poorly reconstructed.

We attempted the application of MOESP to identify eigenvalues for eigenfunction construction. However, it was found that the found eigenfunctions were unable to reconstruct the autonomous dynamics. The forced dynamics were reasonably estimated however.

One of the problems in subspace identification is the application of these methods whenever the forcing x is correlated to the underlying state x, typically occurring whenever the system is operating in closed loop. This problem is similar to the residual $e[k]$ being correlated to the underlying state.

Additionally, extensions to subspace methods for bilinear systems exist [50]. Consequently, work on combining closed-loop identification methods with the bilinear estimation methods could culminate in results applicable to the present work. The field of subspace identification is somewhat mature however, and the present problem does not appear to be a solved problem.

Chapter 7

Conclusion

By performing eigenfunction construction on Koopman operator methods we identify models for nonlinear systems with linear properties. In order to make these models more viable in practical applications it is crucial that the identification of eigenvalues becomes simpler and that the identified models reflect the nonlinearity of the systems.

In the present work we have shown an equivalence of Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) and Dynamic Mode Decomposition (DMD) in the context of Hankel matrices, a relationship not explicitly noted before. This removes a complex optimisation problem from the procedure by Korda and Mezić whilst simultaneously reducing the model order by computing a full model for all states at once. A numerical study shows that the constructed models can obtain a variance accounted for (VAF) up to 98 % for autonomous dynamics on the Toy model and Van der Pol oscillator on a small part of the state-space. The response to a square-wave forcing shows that the suggested linear models are unable to accurately predict a forced response.

We were unable to recover the results presented by Korda and Mezić on time scales larger than presented in their original paper. Furthermore, the results suggest that the original optimisation method is overfitting to the data.

The properties of the Koopman operator eigenvalues allowed us to estimate a linear autonomous system with nonlinear output as a linear system with a high-dimensional state order. This improved the VAF on autonomous dynamics to at least 80% for all autonomous responses. These models were then used to construct a bilinear model, whose optimisation procedure was simplified by exploiting the structure in the optimisation problem. The bilinear model failed to accurately capture the forcing dynamics due to a large error in the underlying state estimate. This underlying state estimate was quantified and investigated in terms of the state reconstruction error. For the constructed models, the state reconstruction error $\|C\varphi(x) - x\|$ was of the same order of magnitude as x. This caused the optimisation problem to be ill-conditioned.

Neural networks can effectively be applied to construct eigenfunctions with linear dynamics for nonlinear systems. Such a network was employed to compare learned eigenfunctions with

the constructed eigenfunctions. The comparison pointed out that the optimal eigenfunctions might nog have sufficient freedom with our choice of Γ. This questions the effect of Γ on the eigenfunctions that can be constructed, and what choice of Γ is optimal for the construction of eigenfunctions.

The use of time-delay coordinates was discussed briefly in the present work, functioning as a starting point for further studies. This already shows that the lifting functions can only be constructed for time-delay coordinates with a sequence of unforced measurements.

We give two primary starting points for future work. First, future work should aim at improving the estimate of the forced dynamics by exploiting existing methods for the identification of bilinear systems, using the estimated eigenvalues for the eigenfunction construction problem. Second, understanding the effects of the initial condition set Γ is paramount to constructing more accurate eigenfunctions.

Bibliography

[1] Korda, M., & Mezić, I. (2020). Optimal construction of Koopman eigenfunctions for prediction and control. *IEEE Transactions on Automatic Control*, *65*(12), 5114–5129.

[2] Ljung, L. (2010). Perspectives on system identification. *Annual Reviews in Control*, *34*(1), 1–12.

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

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

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

[6] Takeishi, N., Kawahara, Y., & Yairi, T. (2017). Subspace dynamic mode decomposition for stochastic Koopman analysis. *Physical Review E*, *96*(3).

[7] Brunton, B. W., Johnson, L. A., Ojemann, J. G., & Kutz, J. N. (2016). Extracting spatial-temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, *258*, 1–15.

[8] Berger, E., Sastuba, M., Vogt, D., Jung, B., & Amor, H. B. (2015). Estimation of perturbations in robotic behavior using dynamic mode decomposition. *Advanced Robotics*, *29*(5), 331–343.

[9] Abraham, I., & Murphey, T. D. (2019). Active learning of dynamics for data-driven control using Koopman operators. *IEEE Transactions on Robotics*, *35*(5), 1071–1083.

[10] Georgescu, M., & Mezić, I. (2015). Building energy modeling: A systematic approach to zoning and model reduction using Koopman mode analysis. *Energy and Buildings*, *86*, 794–802.

[11] Goldschmidt, A., Kaiser, E., DuBois, J., Brunton, S., & Kutz, J. (2021). Bilinear dynamic mode decomposition for quantum control. *New Journal of Physics*, *23*(3), 033035.

[12] Taylor, R., Kutz, J. N., Morgan, K., & Nelson, B. A. (2018). Dynamic mode decomposition for plasma diagnostics and validation. *Review of Scientific Instruments*, *89*(5), 053501.

[13] Wu, H., Nüske, F., Paul, F., Klus, S., Koltai, P., & Noé, F. (2017). Variational Koopman models: Slow collective variables and molecular kinetics from short off-equilibrium simulations. *The Journal of Chemical Physics*, *146*(15), 154104.

[14] Williams, M. O., Hemati, M. S., Dawson, S. T., Kevrekidis, I. G., & Rowley, C. W. (2016). Extending data-driven Koopman analysis to actuated systems [10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016]. *IFAC-PapersOnLine*, *49*(18), 704–709.

[15] Schoukens, J., Vaes, M., & Pintelon, R. (2016). Linear system identification in a nonlinear setting: Nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation. *IEEE Control Systems Magazine*, *36*(3), 38–69.

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

[17] Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., & Kutz, J. N. (2014). On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, *1*(2), 391–421.

[18] Kaiser, E., Kutz, J., & Brunton, S. (2021). Data-driven discovery of Koopman eigenfunctions for control. *Machine Learning: Science and Technology*, *2*.

[19] Roy, R., Paulraj, A., & Kailath, T. (1986). ESPRIT–a subspace rotation approach to estimation of parameters of cisoids in noise. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *34*(5), 1340–1342.

[20] Verhaegen, M., & Verdult, V. (2007). *Filtering and system identification: A least squares approach*. Cambridge University Press.

[21] Takens, F. (1981). Detecting strange attractors in turbulence. *Lecture Notes in Mathematics*, *898*, 336–381.

[22] Stark, J. (1999). Delay embeddings for forced systems. I. Deterministic forcing. *Journal of Nonlinear Science*, *9*, 255–332.

[23] Verdult, V., Verhaegen, M., & Scherpen, J. (2000). Identification of nonlinear nonautonomous state space systems from input-output measurements. *Proceedings of the IEEE International Conference on Industrial Technology 2000*, *1*, 410–414.

[24] Wiggins, S. (1990). *Introduction to applied nonlinear dynamical systems and chaos* (Vol. 2). Springer.

[25] van der Pol, B. (1926). Lxxxviii. on "relaxation-oscillations". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 978–992.

[26] van Neerven, J. (2022). *Functional analysis*. Cambridge University Press.

[27] Brunton, S. L., Budišić, M., Kaiser, E., & Kutz, J. N. (2021). *Modern Koopman theory for dynamical systems*.

[28] Haberman, R. (2013). *Applied partial differential equations: With fourier series and boundary value problems*. Pearson.

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

[30] Colbrook, M. J. (2022). The mpEDMD algorithm for data-driven computations of measure-preserving dynamical systems.

[31] Baddoo, P. J., Herrmann, B., McKeon, B. J., Nathan Kutz, J., & Brunton, S. L. (2023). Physics-informed dynamic mode decomposition. *Proceedings of the Royal Society A*, *479*(2271), 20220576.

[32] Salova, A., Emenheiser, J., Rupe, A., Crutchfield, J. P., & D'Souza, R. M. (2019). Koopman operator and its approximations for systems with symmetries. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *29*(9), 093128.

[33] Haseli, M., & Cortés, J. (2022). Learning Koopman eigenfunctions and invariant subspaces from data: Symmetric subspace decomposition. *IEEE Transactions on Automatic Control*, *67*, 3442–3457.

[34] Pan, S., Arnold-Medabalimi, N., & Duraisamy, K. (2021). Sparsity-promoting algorithms for the discovery of informative Koopman-invariant subspaces. *Journal of Fluid Mechanics*, *917*, A18.

[35] Lusch, B., Kutz, J. N., & Brunton, S. L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, *9*(1).

[36] Bevanda, P., Kirmayr, J., Sosnowski, S., & Hirche, S. (2022). Learning the Koopman eigendecomposition: A diffeomorphic approach. In *Proceedings of the american control conference*. Institute of Electrical; Electronics Engineers.

[37] Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, *113*(15), 3932–3937.

[38] Pisarenko, V. F. (1973). The Retrieval of Harmonics from a Covariance Function. *Geophysical Journal International*, *33*(3), 347–366.

[39] Roy, R., Paulraj, A., & Kailath, T. (1989). ESPRIT–estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(7), 984–995.

[40] Epps, B. P., & Techet, A. H. (2010). An error threshold criterion for singular value decomposition modes extracted from piv data. *Experiments in Fluids*, *48*(2), 355–367.

[41] Brenden P. Epps, E. M. K. (2019). Singular value decomposition of noisy data: Mode corruption. *Experiments in Fluids*, *60*(8), 121.

[42] Huangfu, Q., & Hall, J. A. J. (2018). Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, *10*(1), 119–142.

[43] Bauer, D. (2001). Order estimation for subspace methods. *Automatica*, *37*(10), 1561–1573.

[44] Lambert, J. D., et al. (1991). *Numerical methods for ordinary differential systems* (Vol. 146). Wiley New York.

[45] Arbabi, H., & Mezić, I. (2017). Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator. *SIAM Journal on Applied Dynamical Systems*, *16*(4), 2096–2126.

[46] Lee, C., Hasegawa, H., & Gao, S. (2022). Complex-valued neural networks: A comprehensive survey. *IEEE/CAA Journal of Automatica Sinica*, *9*(8), 1406–1426.

[47] Verhoek, C., Beintema, G. I., Haesaert, S., Schoukens, M., & Toth, R. Deep-learning-based identification of lpv models for nonlinear systems [Cited by: 0]. English. In: *2022-December*. Cited by: 0. Institute of Electrical; Electronics Engineers Inc., 2022, 3274–3280. ISBN: 978-166546761-2.

[48] Verhaegen, M., & Dewilde, P. (1992). Subspace model identification part 1. the output-error state-space model identification class of algorithms. *International Journal of Control*, *56*(5), 1187–1210.

[49] Van Overschee, P., & De Moor, B. (1994). N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems [Special issue on statistical signal processing and control]. *Automatica*, *30*(1), 75–93.

[50] Favoreel, W., De Moor, B., & Van Overschee, P. (1999). Subspace identification of bilinear systems subject to white inputs. *IEEE Transactions on Automatic Control*, *44*(6), 1157–1165.