Title
Koopman Operators and System Identification for Stochastic Systems

Author
Wanner, Mathias Thomas

Publication Date
2023

Peer reviewed|Thesis/dissertation

University of California
Santa Barbara

Koopman Operators and System Identification for Stochastic Systems

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Mechanical Engineering

by

Mathias Wanner

Committee in charge:

Professor Igor Mezić, Chair
Professor Jeffrey Moehlis
Professor Joao Hespanha
Professor Bassam Bamieh

September 2023

The Dissertation of Mathias Wanner is approved.

Professor Jeffrey Moehlis

Professor Joao Hespanha

Professor Bassam Bamieh

Professor Igor Mezić, Committee Chair

August 2023

Koopman Operators and System Identification for Stochastic Systems

To my family

Acknowledgements

I would like to thank my academic advisor Dr. Igor Mezić. In addition to your academic mentorship, you have always respected my research and given me kindness, encouragement, and patience when necessary. I would also like to thank my research group, especially Allan Avila, Gowtham Seenivasaharagavan, and Michael Banks. Your advice and friendship helped me through this process. To my other friends in the Mechanical Engineering department, your companionship made my studies more enjoyable than they had any right to be. Finally, I'd like to thank my parents. You have always been there to support, admonish, encourage, and humble me. You taught me to understand what is truly important; I would never be here without you.

Curriculum Vitæ
Mathias Wanner

Education

2023	Ph.D. in Mechanical Engineering (Expected), University of California, Santa Barbara.
2020	M.S. in Mechanical Engineering, University of California, Santa Barbara.
2018	B.S. in Mechanical Engineering, Villanova University.

Publications

- Wanner, Mathias and Igor Mezić. On Numerical Methods for Stochastic SINDy. *arXiv preprint arXiv*:2306.17814, 2023.
- Wanner, Mathias, and Igor Mezić. Robust Approximation of the Stochastic Koopman Operator. *SIAM Journal on Applied Dynamical Systems* 21.3 : 1930-1951, 2022.
- Ma, Shilin, Kevin McGown, Devon Rhodes, and Mathias Wanner. Explicit bounds for small prime nonresidues. *Journal of Number Theory* 204 : 599-607, 2019.
- Ma, Shilin, Kevin McGown, Devon Rhodes, and Mathias Wanner. On the number of primes for which a polynomial is Eisenstein. *Integers* 18 : 101A 2018.

Conference Presentations

- SIAM Dynamical Systems 2023
  "Finite Difference Approximations for SINDy on Stochastic Systems."
- SIAM Dynamical Systems 2021
  "Robust Dynamic Mode Decomposition for Random Dynamical Systems."

Abstract

Koopman Operators and System Identification for Stochastic Systems

by

Mathias Wanner

The use of the Koopman operator framework in dynamical systems has greatly expanded in recent years. Instead of considering the evolution of the state of a system, the Koopman semigroup tracks the evolution of observables on the state. Since the Koopman operator defined for an arbitrary dynamical system is linear, it allows us to use linear system theory and spectral methods to analyze nonlinear systems. This framework has also been extended to stochastic systems. Since the evolution of observables can only be defined probabilistically for random systems, stochastic Koopman operators are defined by taking the expectation of the future value of observables.

In the first part of this thesis, we review the basic theory of random dynamical systems and stochastic Koopman operators. We can use these operators to represent a nonlinear RDS as an infinite dimensional linear operator. The basic theorems and definitions are given in this section, which will help form the foundation for the algorithms discussed in the second and third sections. Further, some simple examples are given for which the stochastic Koopman operator is well understood. These examples will recur as we use them to test the algorithms in the second section.

The second section is devoted to the analysis of Dynamic Mode Decomposition (DMD) algorithms. DMD algorithms approximate a finite section of the (stochastic) Koopman operator using data from a trajectory. However, these methods are sensitive to noise, and will give a biased approximation if the observables contains randomness. To combat this, we introduce an new DMD variant which can approximate a finite section of the stochastic Koopman operator even when the data contains measurement noise. Further, we extend this algorithm for use with time delayed observables to create a variant of Hankel DMD which will converge for stochastic systems. We then demonstrate these algorithms on numerical examples.

In the final section, we will discuss the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm for stochastic differential equations. The SINDy algorithm allows one to generate a representation of an ODE using a dictionary of functions and data from a trajectory. This algorithm has been extended to SDEs, but the accuracy is limited by the numerical approximations of the drift and diffusion functions. We demonstrate how we can use higher order approximations to these functions to generate a far more accurate representation of the SDE. We then test these approximations on several examples.

Contents

Chapter 1

Stochastic Koopman Operators

1.1 Introduction

The Koopman framework for dynamical systems was originally introduced by Bernhard Koopman, John Von Neumann, and Torsten Carlemann in the 1930s [33, 34, 12]. This new framework represented a shift in viewpoints for the analysis of dynamical systems. Rather than study objects in the state space of the system (e.g. trajectories, fixed points, invariant sets) the Koopman framework focuses on the evolution of observables, or functions, on the state. The Koopman operator, which evolves an observable with the flow of the system, is a linear operator (albeit an infinite dimensional one). Since the Koopman operator is linear, this allows us to the methods of linear algebra and functional analysis to study nonlinear dynamical systems; in particular, we can study its spectrum [54, 48, 52].

The study of the spectrum of the Koopman operator has been fruitful. Eigenfunctions of the Koopman operator have been linked to the geometric objects in the state space of the system [51, 43, 42]. It can be used for linearization and model reduction of large, complex, nonlinear systems [38, 64]. The framework as also been further extended for the control of nonlinear systems [44, 7, 56, 60].

For random dynamical systems, we can similarly define a stochastic Koopman operator, which gives the expected evolution of an observable [54, 48]. Like its deterministic counterpart, this operator also converts a nonlinear system into an infinite dimensional linear operator. For Markov processes, the semigroup of stochastic Koopman operators (or Markov semigroup) can be used to study these nonlinear systems using spectral methods [15]. Connections have also been made between the eigenfunctions of the stochastic Koopman operator and the geometry of the system. For example, level sets of eigenfunctions have been connected to the stochastic isochrons [28] and have been shown to give deterministic factor maps if they have unitary eigenvalues [48].

In this section, we will review the basics of Koopman operator theory and its extension for random dynamical systems. We will also cover some of the basic definitions of the stationary and ergodic measures which will be used in later sections. Finally, we will present some motivating examples for which the spectrum of the stochastic Koopman operator can easily be computed.

1.1.1 Koopman Operators for Deterministic Systems

Consider the dynamical system (discrete or deterministic) on the measurable space $(M, \mathfrak{B})$,

$$\dot{x} = F(x) \qquad \text{or} \qquad x^+ = F(x), \qquad x \in M. \tag{1.1}$$

For a discrete time system, the state space M can be an arbitrary set, but for the continuous time system M will be some manifold.

Rather than track the evolution of x within M, which may not have coordinates, the Koopman framework tracks the evolution of observables, or functions, on M.

Definition 1 *An observable is any $\mathfrak{B}$-measurable function $f : M \to \mathbb{C}$.*

Definition 2 *Let S^t be the flow of* 1.1. *The Koopman family of operators is defined by*

$$U^t f = f \circ S^t.$$

Studying the Koopman evolution of observables has several benefits. First, the state of the system can be reconstructed from a sufficient set of observables, so no information is lost moving to the observable space. Second, a certain choice of observables may lead to a simple representation of the system with linear dynamics. Additionally, since the Koopman operator is a linear operator, it allows us to use spectral methods to study the system.

1.2 Random Dynamical Systems

We will consider a random dynamical systems as defined in [3]. A random dynamical system can be thought of having two parts: a driving flow, θ on a probability space and a set of maps acting on the state space which form a cocycle over θ.

Definition 3 *Let $(\Omega, \mathfrak{F}, P)$ be a probability space, and $\{\theta_t\}_{t\in\mathbb{T}}$ be a group or semigroup of measurable transformations on Ω which preserve the measure P. (Here $\mathbb{T}$ represents the time.) Now, let $(M, \mathfrak{B})$ be a measurable space, and let $T : \Omega \times \mathbb{T} \times M \to M$ be a measurable map. We say T forms a random dynamical system (RDS) on M if the maps $T^t_\omega := T(\omega, t, \cdot)$ from $M \to M$ form a cocycle over $\theta(\cdot)$, i.e.*

$$T^0_\omega = id_M, \quad \textit{and} \quad T^{t+s}_\omega = T^t_{\theta_s(\omega)} \circ T^s_\omega. \tag{1.2}$$

Intuitively, we are interested in the random flow T^t_ω on M. The dynamics on Ω represents the "unknown" system which will drive the randomness of the flow on M. We call $(\Omega, \mathfrak{F}, P, \theta)$ a driving dynamical system and θ_t a driving flow. Further conditions can be imposed upon the cocycle to define continuous, smooth, or linear random dynamical systems.

The semigroup representing time, $\mathbb{T}$, will reflect whether we are dealing with a continuous or discrete time system. For discrete time systems, we will have $\mathbb{T} = \mathbb{N}$ or $\mathbb{N}^+$. In this case we will also denote $T_\omega := T^1_\omega$. For continuous time systems, $\mathbb{T} = \mathbb{R}$ or $\mathbb{R}^+$. If $\mathbb{T}$ is a group (i.e. $\mathbb{T} = \mathbb{R}$ or $\mathbb{N}$), then T^t_ω is invertible, with inverse T^{-t}_ω [3].

For a given RDS, the trajectory of the system will depend on the initial state, x_0, as well as the initial condition of the driving flow, ω_0. We will denote the trajectories of the RDS and the driving flow as

$$x_t = T^t_{\omega_0} x_0 \quad \text{and} \quad \omega_t = \theta_t \omega_0.$$

1.2.1 Markov Processes and Ergodic Measures

Random dynamical systems commonly arise from three different cases: products of random mapping, random differential equations, and stochastic differential equations [3]. Of particular interest to us will be the RDS generated by identically and independently distributed random maps (discrete time) and stochastic differential equations driven by Brownian motion (continuous time). For these system, the process $\{T^t_\omega x, x \in M\}$ has the time homogeneous Markov property (see [14],[31]).

The Markov property states that the future evolution of the system depends only on the current state, not the past, while the time homogeneous property states that the transition probabilities law does not vary over time. To put this precisely, let $\{\mathcal{F}_t\}_{t\in\mathbb{T}}$ be a filtration on $(\Omega, \mathcal{F}, P)$. Given an initial condition (or probability distribution) for x_0, the trajectory x_t will be a random process on Ω.

Definition 4 *The process x_t is Markov with respect to the filtration $\{\mathcal{F}_t\}$ if x_t is $\mathcal{F}_t$ adapted and*

$$P(x_t|\mathcal{F}_s) = P(x_t|x_s)$$

for all $t > s$. It is time homogeneous if, in addition, it satisfies

$$P(x_{t+s}|\mathcal{F}_s) = P(x_t|\mathcal{F}_0).$$

Put in terms of the transition maps, this means that $T^t_{\omega_s}$ is independent of the past $\mathcal{F}_s$. Often, the filtration considered will be the natural filtration of x_t, the one generated by the past of the process:

$$\mathcal{F}^x_t = \sigma\{x_s : s \leq t\}.$$

The two canonical examples of these processes are processes generated by i.i.d. random maps (in discrete time) and stochastic differential equations driven by Brownian motion.

Example 1 (RDS Generated by i.i.d Random Maps) *Let S be a set of maps from M to M. Let $(\Omega, \mathcal{F}, P)$ be a probability space and let $T_{\omega_0}, T_{\omega_1}, T_{\omega_2}, ...$ be an i.i.d. sequence of S valued random variables. Then we can define the cocycle of maps*

$$T_\omega^0 = I, \qquad T_\omega^t = T_1(\omega) \circ T_2(\omega) \circ \ldots \circ T_t(\omega),$$

where $\omega = (\omega_0, \omega_1, \omega_2, ...) \in S^{\mathbb{Z}^+}$. Here the driving flow is the shift map acting on the probability space $(S^{\mathbb{Z}^+}, \mathcal{F}^{\mathbb{Z}^+}, P^{\mathbb{Z}^+})$,

$$\theta\omega = \theta(\omega_0, \omega_1, \omega_2, ...) = (\omega_1, \omega_2, \omega_3, ...).$$

Here $\mathcal{F}^{\mathbb{Z}^+}$ is the σ-algebra of cylinder sets and $P^{\mathbb{Z}^+}$ is the product measure. For more on this representation of the RDS, see [31, 20].

Example 2 (RDS Generated by an SDE) *Consider the SDE*

$$dX = a(X)dt + b(x)dW_t.$$

The cocycle of maps for this system is given by

$$T_\omega^t x = x + \int_0^t a(x)dt + \int_0^t b(x)dW_t.$$

Here ω is identified with a realization of a Weiner process, $\{W_t : t \in \mathbb{R}^+\}$, $\mathcal{F}$ is the Borel σ-algebra, and P is the measure associated with the Weiner process. The driving flow is the Weiner shift

$$\theta_s W_t = W_{t+s} - W_t.$$

(See [3], Appendix A, [15].)

1.2.2 Stationary and Ergodic Measures

When we proceed to the analysis of system identification methods using DMD and SINDy algorithms, specializing to Markov systems will give us tools to evaluate integrals using time averages. To do this, we will also need to assume that our system is stationary, meaning it has a stationary measure, μ.

Definition 5 *A measure μ is called invariant, or stationary, if*

$$\mu(A) = \int_M \int_\Omega \chi_A(T_\omega x) dP \, d\mu,$$

where χ_A is the indicator function for $A \subset M$.

If μ is a stationary measure, we have the equality

$$\int_M \int_\Omega f(T_\omega^{t_1+s} x, ..., T_\omega^{t_n+s} x) \, dP d\mu = \int_M \int_\Omega f(T_\omega^{t_1} x, ..., T_\omega^{t_n} x) \, dP d\mu \tag{1.3}$$

for any $s, t_1, ..., t_n$ ([20], p.86). The stationary measure on M is related to the disintegration of the invariant measure of the skew product system on $M \times \Omega$ (See [3],[4],[55]).

With a stationary system, we will be able to evaluate integrals with respect to the measure using time averages over a trajectory. However, using a single trajectory will only evaluate the time average over whichever invariant set the trajectory starts within. To sample the entire space using a single trajectory, we require the measure μ to be ergodic.

Definition 6 *A set $A \subset M$ is called invariant if*

$$\int_\Omega \chi_A(T_\omega^t x) = \chi_A(x)$$

for almost every x.

Definition 7 *A stationary probability measure μ is called ergodic if every invariant set has measure 0 or 1.*

With these assumptions in place, the ergodic theorem gives us a tool to evaluate integrals using the data from a trajectory.

Lemma 1 *Suppose μ is an ergodic measure. Let*

$$h(x,\omega) = \hat{h}(T_\omega^{t_1}x, T_\omega^{t_2}x, ..., T_\omega^{t_n}x)$$

for some $t_1, t_2, ..., t_n$, with $h \in L^1(\mu \times P)$. Then we have

$$\lim_{m\to\infty} \frac{1}{m} \sum_{j=0}^{m-1} h(x_j, \omega_j) = \int_M \int_\Omega h(x,\omega) dP d\mu \tag{1.4}$$

for almost every (x_0, ω_0) with respect to $\mu \times P$.

Proof: This is theorem 2.2 in chapter 1 of [31], applied to the sum on the left hand side of (1.4). ■

1.3 Stochastic Koopman Operators

The stochastic Koopman family of Operators is defined similarly to deterministic Koopman operators. However, since a random system can have many possible realizations (depending on ω) we cannot directly define the stochastic Koopman operator before. Instead, the stochastic Koopman operators are defined using the expectation of the evolution of observables.

Definition 8 *The stochastic Koopman operator, $\mathcal{K}^t$, is defined for for random systems by*

$$\mathcal{K}^t f(x) = \mathbb{E}_P(f \circ T^t_\omega(x)) = \int_\Omega f \circ T^t_\omega(x) dP.$$

It is easy to see that the stochastic Koopman operator is linear, since it follows from the linearity of function composition and integration.

Definition 8 is valid for any random dynamical system. However, in order for definition 8 to be more useful, we require this family of operators to be consistent in a certain sense: the stochastic Koopman family of operators should form a semigroup:

$$\mathcal{K}^{t+s} f = \mathcal{K}^s \circ \mathcal{K}^t f, \qquad s, t \geq 0. \tag{1.5}$$

To guarantee this, we need the process $\{T^t_\omega x, x \in M\}$ to have the time homogeneous Markov property. When this is the case, (1.5) is guaranteed by the Chapman-Kolmogorov equation ([15]). As mentioned above, an RDS generated by products of i.i.d. maps and stochastic differential equations driven by Brownian motion will have these properties. In the context of Markov processes, the stochastic Koopman operator is also known as the Markov propagator or transition operator [18],[75].

When we have a semigroup of stochastic Koopman operators, we can represent the semigroup using a single operator which generates the semigroup. For discrete time systems, this comes from the stochastic Koopman operator for the one time step map

Definition 9 *For a discrete time system, the Koopman generator is the map given by*

$$\mathcal{K} f = \mathcal{K}^1 f = \mathbb{E}(f(T^1_\omega))$$

so that

$$\mathcal{K}^t = (\mathcal{K})^t$$

as the notation would suggest.

For continuous time systems, this operator is the infinitesimal generator of the subgroup.

Definition 10 *Suppose the Koopman semigroup is strongly continuous. For a continuous time system, the Koopman generator is given by*

$$\mathcal{K}^S = \lim t \to 0^+ \frac{K^t f - f}{t}$$

if it exists. Then

$$\mathcal{K}^t f = e^{\mathcal{K}^S} f.$$

To guarantee that this limit exists for all functions f, we require the Koopman family to be a strongly continuous semigroup. For an SDE driven by a Weiner process we can show that this limit exists for sufficiently smooth functions, and find an explicit expression of the generator.

Example 3 (Stochastic Differential Equation) *For an RDS generate by an SDE, as given in Example 2,*

$$dX_t = a(x)dt + b(x)dW_t.$$

The limit above will exists for all twice differentiable functions with bounded first and second derivatives. For this system, the stochastic Koopman generator is the backwards Kolmogorov operator:

$$\mathcal{K}^S f(x) = a(x)\nabla f(x) + \frac{1}{2}Tr\left(b(x)\nabla^2 f(x) b(x)^T\right),$$

where Tr denotes the trace of a matrix and $\nabla^2 f(x)$ is the hessian of f [15].

1.3.1 Stochastic Koopman Operators on L^2 Spaces

Definition 8 gives the action of the stochastic Koopman operator on an arbitrary function, provided the expectation in the definition is finite. However, to truly define the operator, we also need to specify the function space on which it acts. For an RDS with an invariant measure μ, we consider the function space $L^2(\mu)$. This space of functions (or rather, equivalence classes of functions) gives us a Hilbert space structure and is naturally suited to the stochastic Koopman operator for the RDS.

Operator Norm

When considering the Hilbert space $L^2(\mu)$, the stochastic Koopman operator is a continuous linear operator with unit norm. To see this, we first note that for any $f \in L^1(\mu)$, we have

$$\int_M \mathcal{K}^t f d\mu = \int_M \int_\Omega (f \circ T^t_\omega(x)) dP d\mu(x) = \int_M f d\mu \tag{1.6}$$

which follows almost immediately from definition 5. We can then bound $\|\mathcal{K}\|_2 \leq 1$ using

$$\begin{aligned} \|\mathcal{K}^t f\|_2^2 &= \int_M \left| \int_\Omega f \circ T^t_\omega(x) dp \right|^2 d\mu \leq \int_M \int_\Omega |f|^2 \circ T^t_\omega(x) dP d\mu = \|f\|_2^2 \\ &= \int_M \mathcal{K}^t |f|^2 d\mu = \int_M |f|^2 d\mu = \|f\|_2^2. \end{aligned}$$

It follows that $\|\mathcal{K}\|_2 = 1$ since $\mathcal{K}$ meets this bound with its action on the constant function.

Adjoint of the Stochastic Koopman Operator in $L^2(\mu)$

Since $L^2(\mu)$ is a Hilbert space, we can also consider the L^2 adjoint of $\mathcal{K}^t$. The adjoint of $\mathcal{K}^t$ propagates densities forward in time. For a measure ν, the time evolution of ν is given by

$$\mathcal{L}^t\nu(B) = E\left(\nu\left(\left(T_\omega^{-t}\right)B\right)\right).$$

The adjoint of $\mathcal{K}$, then, evolves densities functions which are in $L^2(\mu)$. If $g \in L^2(\mu)$ and $\nu = g\mu$, then we have

$$\mathcal{L}^t\nu = \mathcal{L}^t(g\mu) = (\mathcal{K}^t)^*g^*\mu.$$

To verify that this is the adjoint, we have

$$\begin{aligned}\langle \mathcal{K}^t f, g\rangle &= \int_M \mathbb{E}_P\left(f(T_\omega^t(x))\right)g^*(x)d\mu = \mathbb{E}_P\left(\int_M f(T_\omega^t(x))d\nu\right)\\ &= \int_M f(x)d(\mathcal{L}^t\nu)(x) = \int_M f(x)(\mathcal{K}^t)^*g^*(x)d\mu = \langle f, (\mathcal{K}^t)^*g\rangle.\end{aligned}$$

We can write the adjoint as

$$(\mathcal{K}^t)^*g(x) = \int_M g(y)P(t,dy,x) = \int_\Omega g(T_\omega^{-t}x)dP(x),$$

where $P(t,B,x) = \int_\Omega \chi_B(T_\omega^t x)dP$ is the transition probability from x to B over time t. The last inequality is only valid if T_ω^t is almost surely invertible (such as when the RDS is defined over two sided time).

We can also consider the adjoint of the infinitesimal generator, $\mathcal{K}^S$. For Markov systems, the generator $\mathcal{K}^S$ is given by the backwards Kolmogorov operator. For sufficiently smooth functions, its adjoint is given by the forward Kolmogorov or Fokker-Planck operator [57, 21]. Given the SDE defined in example 2, this operator reads

$$\mathcal{K}^S g = \sum_{i=1}^{d} \frac{\partial}{\partial x^i}[a^i(x)g(x)] + \sum_{i,j=1}^{d} \frac{\partial^2}{\partial x^i \partial x^j}[D^{i,j}(x)g(x)], \tag{1.7}$$

where $D(x) = b(x)b(x)^T$.

1.4 Spectrum of the Stochastic Koopman Operator

One of the primary motivations for the using the Koopman operator framework is to use spectral methods of analysis for nonlinear systems. The stochastic Koopman operator represents the possibly nonlinear evolution of a system as an infinite dimensional linear operator on the space of observables. Since $\mathcal{K}^t$ a linear operator, we can find its eigenvalues and eigenfunctions, and use them to study the system.

We will make a distinction between the eigenfunctions for the discrete and continuous time cases.

Definition 11 *We say the a function ϕ is a stochastic Koopman eigenfunction with eigenvalue λ if*

$$\mathcal{K}^t \phi = \lambda^t \phi$$

in the discrete time case and

$$\mathcal{K}^t \phi = e^{\lambda t}$$

in the continuous time case.

In discrete time, this definitions corresponds to eigenvalues of the one step Koopman operator $\mathcal{K}$. For continuous time systems (SDEs), it corresponds to eigenvalues of the generator provided the eigenfunction ϕ is sufficiently smooth [15].

Given a set of eigenfunctions, $\phi^1, \phi^2, ..., \phi^d$ which seperate the points of the state space M, we can find a representation of the RDS using those functions as coordinates. This gives us a system with a linear expected evolution, but the noise of the system may behave nonlinearly. For a discrete time system, we have

$$\Phi_{t+1} = A\phi_t + n(\omega, x), \tag{1.8}$$

where

$$\Phi_t = \begin{bmatrix} \phi_t^1 \\ \phi_t^2 \\ \vdots \\ \phi_t^d \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_d \end{bmatrix}.$$

The noise term is simply defined by $n(\omega, x) = \Phi(T_\omega x) - A\Phi(x)$, and has zero mean.

For the SDE case, we get a representation linear drift but a nonlinear diffusion function.

$$d\Phi_t = A\Phi_t dt + b(\Phi_t)dW_t, \tag{1.9}$$

where $b(\Phi_t)$ is the nonlinear diffusion function. This representation and the diffusion function b can be computed using Ito's formula.

1.4.1 Spectral Expansions

Given an eigenfunction, the expected evolution is easy to compute using definition 11. Further, the representations (1.8) and (1.9) allow us to compute the evolution of eigenfunctions as a linear system with some zero-mean additive noise. We can also use this to compute the evolution of an arbitrary (vector valued) function f using a spectral expansion. Suppose we have a set of eigenfunctions $\phi_1, \phi_2, ...$ which form a basis for

$L^2(\mu)$. Then we can write

$$f = \sum_{i=1}^{\infty} v_i \phi_i.$$

The stochastic Koopman evolution of f is given (in discrete time) by

$$\mathcal{K}^t f = \sum_{i=1}^{\infty} v_i \lambda_i^t \phi_i. \tag{1.10}$$

(For continuous time, we would replace λ_i^t with $e^{\lambda_i t}$.) This spectral expansion is in analogy to the deterministic Koopman mode decomposition given in [49, 52]. The quantities v_i are called the Koopman modes associated with the observable f, and are calculated using

$$v_i = \langle f, \phi_i \rangle.$$

Here $\langle \cdot, \cdot \rangle$ denotes the $L^2(\mu)$ inner product.

However, in order for expansion (1.10) to be valid, we need the eigenfunctions to span a dense subset of $L^2(\mu)$. This will be the case for the random rotation and linear system examples below. More generally, if $\mathcal{K}^t$ is a compact operator, the spectrum contains only eigenvalues (except at the accumulation point 0, [65] Theorem 4.25), and we have this type of expansion. The expansion (1.10) can be extended to vector valued observables. In this case, the Koopman modes v_i would be vectors whose elements are the modes for each component of f.

1.4.2 Example: Random Rotation

A first, simple example of a RDS for which we can compute the eigenvalues and eigenfunctions of the stochastic Koopman operator is that of a random rotation on a circle

$$\theta_{t+1} = \theta_t + n(\omega_t) \pmod{2\pi},$$

where $n(\omega_t)$ is an i.i.d. random variable in $[-\pi, \pi)$. For this system, the eigenfunctions are the complex harmonics $\phi_i = e^{i\theta},\ i \in \mathbb{Z}$, with the eigenvalues given by

$$\lambda_i = \int_\Omega e^{in(\omega)} dP.$$

(See [25],[15] Example 2.) We can generalize this example to all compact Abelian groups.

Example 4 *Let A be a compact Abelian group. Consider the random dynamical system given by $T_\omega(x) = x + n(\omega)$, or*

$$x_{t+1} = x_t + n(\omega_t),$$

where $n(\omega_t)$ is an i.i.d. random element of A.

Theorem 1 *The stochastic Koopman operator for this system has discrete spectrum, and the eigenfunctions are the continuous characters, ϕ_i, of the group A. The eigenvalues are given by*

$$\lambda_i = \int_\Omega \phi(n(\omega)) dP.$$

Proof: Since A is a compact Abelian group, A has a unique Haar measure μ which can be normalized such that $|\mu| = 1$ ([66], 1.1.3). The measure μ is the invariant measure for the RDS. In fact, since μ is a Haar measure, μ is preserved by T_ω for all ω.

Now, consider a continuous character ϕ_i of A. Since ϕ is multiplicitive, we have

$$\begin{aligned} \mathcal{K}\phi(x) &= \int_\Omega \phi(T_\omega x) dP = \int_\Omega \phi(x + n(\omega)) dP = \int_\Omega \phi(x)\phi(n(\omega)) dP \\ &= \left(\int_\Omega \phi(n(\omega)) dP \right) \phi(x) = \lambda \phi(x). \end{aligned}$$

This establishes that ϕ is an eigenfunction with eigenvalue

$$\lambda = \int_\Omega \phi(n(\omega)) dP.$$

The fact that these are all of the eigenfunctions and that the spectrum is discrete follows from the Plancherel Theorem ([66], Theorem 1.6.1), which shows that the continuous characters are dense in $L^2(\mu)$. ■

For these systems, all of the eigenfunctions are known (and do not depend on the distribution of $n(\omega)$), and we can compute all of the eigenvalues. The Koopman mode decomposition (1.10) follows directly from the Fourier expansion on compact Abelian groups. The continuous time analogue to these systems would be an SDEs on a n-dimensional torus with constant drift.

1.4.3 Example: Linear System

Another example for which we can establish some results about the spectrum is an affine system. A similar example is also considered in [15], Proposition 1.

Example 5 *Let $A : \Omega \to \mathbb{R}^{d\times d}$ and $\nu : \Omega \to \mathbb{R}^d$ be such that $A(\omega_t)$ and $\nu(\omega_t)$ are i.i.d. random variables. We can define the one step map $T_\omega(x) = A(\omega)x + \nu(\omega)$, which gives the system*

$$x_{t+1} = A(\omega_t)x_t + \nu(\omega_t).$$

For this system, we will consider the stochastic Koopman operator acting on the space of analytic functions, $\mathcal{H}$. Depending on the invariant measure μ, this space may be dense in $L^2(\mu)$. For example, if μ is a Gaussian distribution, this will be true since the Hermite polynomials form an orthonormal basis for this space. For the operator acting on $\mathcal{H}$, we have the following results

Theorem 2 *Consider the system in Example 5. Suppose that all of the moments and joint moments of A and ν up to order n are finite. The following are true*

1. *The subspace $\mathcal{P}_n$ of all polynomials up to order n is invariant under $\mathcal{K}$. The re-*

striction of $\mathcal{K}$ to this subspace using monomials as a basis will be block triangular.

2. *If the monomials form a basis of $L^2(\mu)$, the spectrum of $\mathcal{K}$ on $L^2(\mu)$ is discrete.*

3. *If the matrices $A(\omega)$ commute and are diagonalizable, the eigenvalues of $\mathcal{K}$ are of the form*
$$\lambda = \mathbb{E}_P \left(\prod_{i=1}^{d} \lambda_i(\omega)^{a_i} \right) P,$$
where $\lambda_1(\omega), ..., \lambda_d(\omega)$ are the eigenvalues of A_ω and $a_i \in \mathbb{Z}^+$.

4. *If $A(\omega) = A$ is a constant matrix and is diagonalizable, then the eigenvalues are of the form*
$$\lambda = \prod_{i=1}^{d} \lambda_i^{a_i},$$
where $\lambda_1, ..., \lambda_d$ are the eigenvalues of A.

Proof:

1. The assumption that all of the moments are finite guarantees that the action of $\mathcal{K}$ on any monomial is finite, so $\mathcal{K}p$ is well defined for any polynomial. For any monomial of degree n, $p = \prod_{i=1}^{d} (x^i)^{a_i}$, we have
$$\mathcal{K}p = \mathbb{E} \left(\prod_{i=1}^{d} \left(\sum_{i=1}^{d} A^{i,j} x^j + \nu^i \right)^{a_i} \right),$$
which is a polynomial of degree n. The block triangular structure of the restriction of $\mathcal{K}$ follows from the nested structure of $\mathcal{P}_n$.

2. Assertion 2 immediately follows from assertion 1, since every monomial falls in a finite dimensional invariant subspace.

3. If the matrices $A(\omega)$ commute and are diagonalizable, they are simultaneously

diagonlizable, so we can find a matrix T such that

$$TA(\omega)T^{-1} = \begin{bmatrix} \lambda_1(\omega) & 0 & \dots & 0 \\ 0 & \lambda_2(\omega) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_d(\omega) \end{bmatrix},$$

where $\lambda_1, ..., \lambda_d$ are the eigenvalues of $A(\omega)$. Letting $y = Tx$, we can use monomials in y as a basis for $\mathcal{P}_\backslash$. The action of $\mathcal{K}$ on the monomial $p = \prod_{i=1}^d (y^i)^{a_i}$ is given by

$$\mathcal{K}p = \mathbb{E}\left(\prod_{i=1}^d (\lambda_i(\omega)y^i + \nu_i)^{a_i}\right) = \mathbb{E}\left(\prod_{i=1}^d \lambda_i(\omega)\right) p + p',$$

where p' is a polynomial of degree $n-1$. This shows that restriction of $\mathcal{K}$ to $\mathcal{P}_n$ is triangular, with diagonal elements $\mathbb{E}\left(\prod_{i=1}^d \lambda_i(\omega)^{a_i}\right)$, which proves the assertion.

4. Assertion 4 immediately follows from assertion 3.

■

From Theorem 2, we can find a spectral expansion as in (1.10) for analytic functions. From assertion 2, this will also allow us to find spectral expansions for $L^2(\mu)$ provided the polynomials span $L^2(\mu)$. For example, if $A(\omega) = A$ is constant and $\nu(\omega)$ has a guassian distribution, μ will also be a gaussian and this will be the case. Similarly, for the continuous time analogue of this system, the Ornstein-Uhlenbeck process, the polynomials also form a basis of $L^2(\mu)$ and the spectrum is discrete (the Hermite-polynomials are the eigenfunctions [39]).

Chapter 2

Noise Resistant Dynamic Mode Decomposition

2.1 Introduction

While the Koopman or stochastic Koopman operators provide many tools for the analysis of dynamical systems, computing the spectrum of the operator is nontrivial. Even for systems generated from SDEs (which are often simpler than discrete time systems), solving for the stochastic Koopman eigenfunctions equates to finding the eigenfunctions of a partial differential operator, often on an unbounded domain. Further, for many complex systems and processes the governing equations cannot be derived through first principles or the models generated by them may be too complicated, meaning we cannot solve for the eigenfunctions analytically.

Instead, the eigenvalues and eigenfunctions are often computed in a data-driven setting. Given an observable, we would like to find the Koopman mode decomposition of f 1.10 using the data from f measured along a trajectory $x_0, x_1, ..., x_n$. An early method of Koopman mode decomposition is generalized Laplace analysis, which was based on the generalized Laplace transform [47]. However, this method of decomposition requires knowledge of the eigenvalues a priori and does not work for random systems.

Another data driven method is Dynamic Mode Decomposition (DMD), which was introduced in [67] and shown to be connected to KMD in [64]. DMD algorithms attempt to find a matrix which approximates a finite section of the Koopman operator [53]. There are many different variants of DMD (e.g. [68, 59, 37, 24]) and it can be used for a wide array of applications such as fluid flows [64, 49, 50], soft robotics [5, 22], and infectious disease [61]. These algorithms can be used for both deterministic systems and for random systems generated by i.i.d. random maps [74]. However, many DMD algorithms possess a major drawback; they can fail if the data from the observables contains measurement noise or other randomness. In this case, the results from standard DMD algorithms are biased [17].

Total Least Squares (TLS) DMD ([17], [23]) was developed to remove the bias for systems with measurement noise, but only converges when the underlying dynamics are deterministic. In [70], subspace DMD was introduced to converge for observables with additive noise even when the underlying dynamics are random. While many of these methods can combat the bias from measurement noise in DMD, they impose relatively strict assumptions on either the dynamics or the structure of the noise.

Of particular interest are Krylov subspace based DMD methods, where the iterates of a single observable under the Koopman evolution is used to (approximately) generate an invariant subspace of the Koopman operator [8],[53]. For deterministic systems, Hankel DMD uses time delays of a single observable to generate the Krylov subspace, and the DMD spectrum was shown to converge to the spectrum of the Koopman operator[2, 1]. This allows us to generate a model of a deterministic system using the data from a single trajectory of a single observable. However, for random systems, the time delayed observables contain randomness from the dynamics, and Hankel DMD does not converge. Further, the noise introduced is neither i.i.d. nor independent of the state. In [15], a new stochastic Hankel DMD algorithm was shown to converge, but it requires the Stochastic Koopman evolution of the observable, which in general requires multiple realizations of the system.

In this section, we describe a new DMD algorithm which allows us to work with a more general set of noisy observables. This algorithm provably approximates the stochastic Koopman operator in the large data limit and allows for more general randomness in the observables than i.i.d. measurement noise. With these weaker conditions, we can use time delayed observables to form a Krylov subspace of observables, which gives us a variation of Hankel DMD for random systems. This allows us to compute a realization of the stochastic Koopman operator using data from a single observable over a single realization of the system.

2.2 Dynamic Mode Decomposition

Dynamic Mode Decomposition is an algorithm which allows the computation of an approximation of a finite section of the Koopman operator from data. Since we are dealing with data, for the remainder of this chapter we will assome our RDS acts on discrete time, and the operator $\mathcal{K}$ is the one step stochastic Koopman operator. Assuming the eigenfunctions, ϕ_j, of $\mathcal{K}$ span our function space, we can decompose any (possibly vector valued) observable $\mathbf{f}$ using the Koopman mode decomposition (1.10)

$$\mathbf{f} = \sum_j v_j \phi_j.$$

The expected evolution of f is then given by

$$\mathbb{E}_P(\mathbf{f}(T_\omega x)) = \sum_j v_j \mathcal{K}\phi_j(x) = \sum_j \lambda_j v_j \phi_j(x). \tag{2.1}$$

As noted in Section 1.4.1 the functions ϕ_j are the Koopman eigenfunctions with eigenvalue λ_j, and the vectors v_j are called the Koopman modes associated with $\mathbf{f}$. However, the expansion above can contain an infinite number of terms. In order to work with (2.1) using finite arithmetic, we must restrict ourselves to a finite dimensional subspace of our original function space.

Let $\mathscr{F}$ be a finite dimensional subspace of $L^2(\mu)$ and $\bar{\mathscr{F}}$ be its orthogonal complement. Let P_1 and P_2 be the projections on to $\mathscr{F}$ and $\bar{\mathscr{F}}$. For any function $g \in L_2(\mu)$, we can compute the Koopman evolution as

$$\mathcal{K}g = P_1\mathcal{K}g + P_2\mathcal{K}g = P_1\mathcal{K}P_1 g + P_2\mathcal{K}P_1 g + P_1\mathcal{K}P_2 g + P_2\mathcal{K}P_2 g.$$

The operator $P_1\mathcal{K}P_1$ maps $\mathscr{F}$ into itself. For any $g \in \mathscr{F}$, we have $P_2 g = 0$, so we can view $P_1\mathcal{K}P_1$ as an approximation of $\mathcal{K}$ provided $\|P_2\mathcal{K}P_1\|$ is small. If $\mathscr{F}$ is an invariant subspace under $\mathcal{K}$, we have $\|P_2\mathcal{K}P_1\| = 0$, and $\mathcal{K}g = P_1\mathcal{K}P_1 g$ for all $g \in \mathscr{F}$. If we let $f_1, f_2, ..., f_k$ be a basis for $\mathscr{F}$, we can represent the restriction of $P_1\mathcal{K}P_1$ to $\mathscr{F}$ as a matrix $\mathbf{K}$ that acts on the basis by

$$\mathbf{K}\begin{bmatrix} f_1 & f_2 & \dots & f_k \end{bmatrix}^T = \begin{bmatrix} \mathcal{K}f_1 & \mathcal{K}f_2 & \dots & \mathcal{K}f_k \end{bmatrix}^T. \tag{2.2}$$

Remark 1 *The matrix K can also be thought of as the matrix acting (on the right) on the vector of coefficients of functions represented in the basis $f_1, \dots, f_k$: for any function $g \in \mathscr{F}$ we can write*

$$g = \sum_{j=1}^{k} a_j f_j = \mathbf{a}\begin{bmatrix} f_1 & \dots & f_k \end{bmatrix}^T,$$

and $\mathbf{a} = \begin{bmatrix} a_1 & \dots & a_k \end{bmatrix}$ is the row vector of coefficients of g. Then $(\mathbf{aK})$ is the row vector of coefficients for $\mathcal{K}g$, since

$$\mathcal{K}g = \mathcal{K}(\mathbf{a}\begin{bmatrix} f_1 & \dots & f_k \end{bmatrix}^T) = \mathbf{a}\begin{bmatrix} \mathcal{K}f_1 & \dots & \mathcal{K}f_k \end{bmatrix} = \mathbf{aK}\begin{bmatrix} f_1 & \dots & f_k \end{bmatrix}^T$$

2.3 Basic DMD Algorithms

Dynamic mode decomposition algorithms compute the spectral expansion (1.10) by approximating the matrix $\mathbf{K}$ from data. If we can measure the observables $f_1, f_2, ..., f_k$ along a trajectory $x_0, x_1, ..., x_n$, we can form the vector valued observable $\mathbf{f} : M \to \mathbb{R}^k$ by

$$\mathbf{f} = \begin{bmatrix} f_1 & f_2 & \dots & f_k \end{bmatrix}^T.$$

Each $\mathbf{f}(x_t)$ is called a data snapshot. Given a data matrix whose columns are snapshots of $\mathbf{f}$,

$$D = \begin{bmatrix} \mathbf{f}(x_0) & \mathbf{f}(x_1) & \dots & \mathbf{f}(x_n) \end{bmatrix},$$

we can construct an operator $A : \mathbb{R}^k \to \mathbb{R}^k$, called the DMD operator, which (approximately) maps each data snapshot to the next one, i.e.

$$A\mathbf{f}(x_i) \approx \mathbf{f}(x_{i+1}).$$

Standard DMD algorithms (see [67, 74, 53],[37] and the sources therein) construct a matrix C to minimize the squared error

$$C = \operatorname*{argmin}_{A} \sum_{i=0}^{n-1} \|A\mathbf{f}(x_i) - \mathbf{f}(x_{i+1})\|_2^2. \tag{2.3}$$

Algorithm 1: Extended Dynamic Mode Decomposition (EDMD)

Let $x_0, x_1, ..., x_n$ be a trajectory of our random dynamical system and $\mathbf{f} : M \to \mathbb{C}^k$ be a vector valued observable on our system.

1: Construct the data matrices

$$X = \begin{bmatrix} \mathbf{f}(x_0) & \mathbf{f}(x_1) & \dots & \mathbf{f}(x_{n-1}) \end{bmatrix}, \qquad Y = \begin{bmatrix} \mathbf{f}(x_1) & \mathbf{f}(x_2) & \dots & \mathbf{f}(x_n) \end{bmatrix}.$$

2: Form the matrix

$$C = YX^\dagger,$$

where $X^\dagger$ is the Moore-Penrose psuedoinverse.

3. Compute the eigenvalues and left and right eigenvectors, (λ_i, w_i, v_i) $i = 1, 2, ..., k$, of

C. Then the dynamic eigenvalues are λ_i, the dynamic modes are v_i, and the numerical eigenfunctions are given by

$$\hat{\phi}_i = w_i^T X.$$

Let $f_1, f_2, ..., f_k$ be the components of $\mathbf{f}$. If we let $\hat{f}_i$ be the i^{th} row of X,

$$\hat{f}_i = \begin{bmatrix} f_i(x_0) & f_i(x_1) & \dots & f_i(x_{n-1}) \end{bmatrix},$$

we see that $\hat{f}_i$ represents f_i by evaluating it along a trajectory. With standard DMD, we construct the DMD operator C represented in the basis $\hat{f}_1, \hat{f}_2, ..., \hat{f}_k$. The numerical eigenfunctions, $\hat{\phi}_i$ will be approximations of eigenfunctions of the (stochastic) Koopman operator evaluated along our trajectory.

The realization of the matrix C will depend on the basis $\hat{f}_1, ..., \hat{f}_k$. This choice will also affect the conditioning of the pseudo-inversion in EDMD. If the basis leads to an ill conditioned matrix X, the algorithm will be numerically unstable. To combat this, EDMD is usually implemented using a truncated singular value decomposition of X. This leads to the second algorithm [67].

Algorithm 2: SVD implemented EDMD

Let $x_0, x_1, ..., x_n$ be a trajectory of our random dynamical system and, $f_1, f_2, ..., f_l$, $l \geq k$, be a set of l observables on our system.

1: Construct the data matrices

$$X = \begin{bmatrix} \mathbf{f}(0) & \mathbf{f}(1) & \dots & \mathbf{f}(n-1) \end{bmatrix}, \qquad Y = \begin{bmatrix} \mathbf{f}(1) & \mathbf{f}(2) & \dots & \mathbf{f}(n) \end{bmatrix}.$$

2: Compute the truncated SVD of X using the first k singular values.

$$X = W_k S_k V_k^*.$$

3: Form the matrix

$$A = S_k^{-1} W_k^* Y V_k.$$

4. Compute the eigenvalues and left and right eigenvectors, (λ_i, w_i, u_i) $i = 1, 2, ..., k$, of A. Then the dynamic eigenvalues are λ_i, the dynamic modes are

$$v_i = W S u_i,$$

and the numerical eigenfunctions are given by

$$\hat{\phi}_i = w_i^T V_k^*.$$

The benefit of SVD based EDMD is that it offers more numerically stability. If X has a large condition number, the pseudoinversion of X can introduce large errors to the DMD operator which may make Algorithm 1 unusable. To combat this, Algorithm 2 computes the SVD of X and truncates the matrix S_k to include only the dominant singular values. Since S_k has a smaller condition number than X, the inversion of S_k in Algorithm 2 is more numerically stable than the psuedoinversion of X. Intuitively, the truncated SVD used in Algorithm 2 chooses a better conditioned basis of observables to construct the DMD operator. The matrix A generated in Algorithm 2 is the same as the matrix C produced by Algorithm 1 using the $k-$dimensional observable $\mathbf{f}_{new} = S_k^{-1} W^* \mathbf{f}$.

2.4 Convergence of DMD for Random Systems

The utility of Algorithms 1 and 2 comes from the convergence of the dynamic eigenvalues and numerical eigenfunctions to eigenvalues and eigenfunctions of $\mathcal{K}$.

Proposition 1 *Let T be an i.i.d. random system with ergodic measure μ. Let $\mathscr{F}$ be a k dimensional subspace of $L^2(\mu)$ which is invariant under the action of $\mathcal{K}$, and let $f_1, f_2, ..., f_k$ span $\mathscr{F}$. Let $\lambda_{j,n}$ be the dynamic eigenvalues and $v_{j,n}$ be the dynamic modes produced by EDMD using the trajectory $x_0, x_1, ..., x_n$. Then, as $n \to \infty$, the dynamic eigenvalues converge to the eigenvalues of $\mathcal{K}$ restricted to $\mathscr{F}$ for almost every initial condition (x_0, ω_0) with respect to $(\mu \times P)$. If the eigenvalues of $\mathcal{K}$ are distinct, the numerical eigenfunctions converge to a sampling of the eigenfunctions along the trajectory.*

The proof of Proposition 1 is fairly standard in the DMD literature (e.g. [74]) and does not differ from the deterministic case, but we include it for completeness. The key idea is that we can use the ergodic time average to evaluate the integrals

$$\int_M \int_\Omega f_i(T_\omega) f_j^* dP \, d\mu.$$

Proof: Let $f_1, f_2, ..., f_k$, and $\mathbf{K}$ be as described in (2.2). Let X_n, Y_n, and C_n be the matrices produced by Algorithm 1 for the trajectory $x_0, x_1, ..., x_n$, and let $\omega_0, \omega_1, ..., \omega_n$ be the evolution of the noise. Let $\mathbf{f} = \begin{bmatrix} f_1 & f_2 & \dots & f_k \end{bmatrix}^T$ as above. Define the matrices

$$G_0 = \int_M \begin{bmatrix} f_1 & f_2 & \dots & f_k \end{bmatrix}^T \begin{bmatrix} f_1^* & f_2^* & \dots & f_k^* \end{bmatrix} d\mu = \int_M \mathbf{f}\,\mathbf{f}^* \, d\mu$$

and

$$G_1 = \int_M \begin{bmatrix} \mathcal{K} f_1 & \mathcal{K} f_2 & \dots & \mathcal{K} f_k \end{bmatrix}^T \begin{bmatrix} f_1^* & f_2^* & \dots & f_k^* \end{bmatrix} d\mu = \int_M \mathbf{K}\,\mathbf{f}\,\mathbf{f}^* \, d\mu = \mathbf{K} G_0.$$

We can see that G_0 has full rank, since if $\mathbf{v}$ was in its nullspace we would have

$$\|\mathbf{f}^*\mathbf{v}\|^2 = \mathbf{v}^* G_0 \mathbf{v} = 0,$$

which implies $\mathbf{v} = 0$ since $f_1, f_2, ..., f_k$ are linearly independent. This gives us $\mathbf{K} = G_0^{-1} G_1$.

Now, let $G_{0,n} = \frac{1}{n} X_n X_n^*$ and $G_{1,n} = \frac{1}{n} X_n Y_n^*$. We have $G_{0,n} \to G_0$ and $G_{1,n} \to G_1$ for almost every initial condition (x_0, ω_0). To see this, by Lemma 1 we have

$$\begin{aligned}\lim_{n\to\infty} G_{1,n} &= \lim_{n\to\infty} \frac{1}{n} \sum_{m=0}^{n-1} \mathbf{f}(x_{m+1})\mathbf{f}^*(x_m) = \lim_{n\to\infty} \frac{1}{n} \sum_{m=0}^{n-1} \mathbf{f}(T_{\omega_m} x_m)\mathbf{f}^*(x_m) \\ &= \int_M \int_P \mathbf{f}(T_\omega x)\mathbf{f}^*(x)\, dP d\mu = \int_M \mathbf{K}\,\mathbf{f}(x)\,\mathbf{f}^*(x)\, d\mu = G_1,\end{aligned}$$

and similarly for G_0, we have

$$\lim_{n\to\infty} G_{0,n} = \lim_{n\to\infty} \frac{1}{n} \sum_{m=0}^{n-1} \mathbf{f}(x_m)\mathbf{f}^*(x_m) = \int_M \int_\Omega \mathbf{f}(x)\mathbf{f}^*(x)\, dP d\mu = G_0.$$

Since G_0 has full rank and $G_{0,n} \to G_0$, $G_{0,n}$ is full rank for n large enough, so $G_{0,n}^{-1}$ exists and

$$\lim_{n\to\infty} G_{0,n}^{-1} G_{1,n} = G_0^{-1} G_1 = \mathbf{K}.$$

Because $G_{0,n} = \frac{1}{n} X_n X_n^*$, we know X_n has full row rank for n large enough, so

$$C_n = Y_n (X_n)^\dagger = Y_n X_n^* (X_n X_n^*)^{-1} = \left(\frac{1}{n} Y_n X_n^*\right)\left(\frac{1}{n} X_n X_n^*\right)^{-1} = G_{0,n}^{-1} G_{1,n},$$

which shows that $C_n \to \mathbf{K}$. This shows that the dynamic eigenvalues, $\lambda_{j,n}$, converge to the eigenvalues of $\mathbf{K}$, λ_j, as $n \to \infty$.

To show the numerical eigenfunctions converge to samplings of our eigenfunctions, let $w_{j,n}$ and w_j be the left eigenvectors of C_n and $\mathbf{K}$, respectively. Consider the functions $\phi_{j,n} = w_{j,n}^T \mathbf{f}$ and $\phi_j = w_j^T \mathbf{f}$. We know ϕ_j is a Koopman eigenfunction, since

$$\mathcal{K}\phi_j = \mathcal{K}(w_j^T \mathbf{f}) = w_j^T \mathbf{K}\, \mathbf{f} = \lambda_j w_j^T \mathbf{f} = \lambda_j \phi_j.$$

If $\mathbf{K}$ has distinct eigenvalues, the vectors $w_{j,n}$ each converge to w_j, so $\phi_{j,n} \to \phi_j$. The numerical eigenfunctions, $\hat{\phi}_{j,n}$, are the values of the function $\phi_{j,n}$ sampled along the trajectory $x_0, ..., x_{n-1}$. ■

The proof of Proposition 1 is based on the convergence of time averages to inner products of functions in $L^2(\mu)$. In particular, the i, j^{th} entry of $G_{0,n}$ and $G_{1,n}$ converge to $\langle f_i, f_j \rangle$ and $\langle \mathcal{K} f_i, f_j \rangle$, respectively, where $\langle \cdot, \cdot \rangle$ is the $L^2(\mu)$ inner product. As such, we cannot glean any information about dynamics outside the support of μ. There could be an eigenvalue/eigenfunction pair, (λ, ϕ), such that ϕ is zero on the support of μ. Such a pair cannot be captured by EDMD from a single trajectory since $\phi = 0$ almost everywhere with respect to μ. In particular, if μ is a singular measure concentrated on some attractor, the eigenvalues governing the dissipation to the attractor cannot be found using ergodic sampling.

2.5 Linear System Identification

The proof above shows that Dynamic Mode Decomposition converges for random dynamical systems with i.i.d. dynamics. However, it is important to note that although the systems can have randomness, the observables cannot. The stochastic Koopman operator acts on functions, $f : M \to \mathbb{C}$, which depend only on the state of the system. If we allow our observables to have some noise (i.e. dependence on ω), the proof fails. In particular, observables with i.i.d. measurement noise and time delayed observables (used in Hankel DMD) both have some dependence on ω, and therefore cannot be used with the above DMD methods.

Examining the failure of standard DMD on linear systems with measurement noise is instructive. Consider the system

$$x_{t+1} = Ax_t + \nu_t, \qquad y_t = x_t + n_t. \tag{2.4}$$

where $\nu, n : \Omega \to \mathbb{R}^d$ are i.i.d. Gaussian random variables on $(\Omega, \mathcal{F}, P)$. Here x_t would be the state of our system, while y_t would be the observable with measurement noise. We will assume that the noise n_t has zero mean and the system has invariant measure μ. Given data $y_0, y_1, ..., y_N$ from a trajectory of the system, let

$$Y_0 = \begin{bmatrix} y_0 & y_1 & \dots & y_{N-1} \end{bmatrix} \quad \text{and} \quad Y_1 = \begin{bmatrix} y_1 & y_2 & \dots & y_N \end{bmatrix}.$$

The DMD matrix C for this system is given be the least squares estimate

$$C = \operatorname*{argmin}_A \sum_{i=0}^{n-1} \|Ay_i - y_{i+1}\|_2^2$$

or $C = Y_1 Y_0^\dagger$. If the measurement noise n_t is identically zero (i.e. $y_t = x_t$), we can find an unbiased estimate of A using this least squares estimate (see [40] Chapter 7 and Appendix II). However, if n_t is nonzero, we cannot directly use the least squares method to produce an unbiased estimate.

To see why this fails with noise, we rewrite the least squares estimate as

$$C = Y_1 Y_0^\dagger = \left(\frac{1}{N} \sum_{i=0}^{N-1} y_{i+1} y_i^* \right) \left(\frac{1}{N} \sum_{i=0}^{N} y_i y_i^* \right)^{-1} = G_1 G_0^{-1}$$

assuming G_0 is of full rank. Then as $N \to \infty$

$$G_0 = \sum_{i=0}^{N-1} (x_i + n_i)\,(x_i^* + n_i^*) \to \int_{\mathbb{R}^d} x\,x^* d\mu + \mathbb{E}_P(n_{i+1} n_i *) = \int_{\mathbb{R}^d} x\,x^* d\mu + cov(n_i, n_i),$$

where $cov(a, b)$ denotes the covariance matrix of two random variables. Similarly

$$G_1 \to \int_{\mathbb{R}^d} Ax\,x^* d\mu + cov(n_{i+1}, n_i) = A\left(\int_{\mathbb{R}^d} x\,x^* d\mu\right) + cov(n_{i+1}, n_i) + cov(\nu_i, n_i).$$

Here the G_0 and G_1 matrices are biased by the covariance of the noise as opposed to the noiseless case. Solving $C = G_1 G_0^{-1}$ will propagate the bias to the DMD operator C.

2.5.1 Instrumental Variables

One method of correcting the bias in the least squares estimate is to use instrumental variables [62, 63, 40, 69]. The idea is to use an extra set of variables $\zeta = [\zeta^1 \ \ldots \ \zeta^d]^T$, called instruments, which are independent from the noises n_t, n_{t+1}, and ν_t. If we have measurements of ζ_t along the trajectory in addition to y_t, we can construct the time averages

$$G_0 = \frac{1}{N}\sum_{i=0}^{N-1} y_i \zeta_i^* \to \mathbb{E}_P\left(\int_{\mathbb{R}^d} x\,\zeta^* d\mu\right) + cov(n_t, \zeta_t) = \mathbb{E}_P\left(\int_{\mathbb{R}^d} x\,\zeta^* d\mu\right)$$

and

$$G_1 = \frac{1}{N}\sum_{i=0}^{N-1} y_{i+1} \zeta_i^* \to A\,\mathbb{E}_P\left(\int_{\mathbb{R}^d} x\,\zeta^* d\mu\right).$$

where all of the covariance terms go to zero due to independence. Then we can compute $A = G_1 G_0^{-1}$, assuming G_0 is full rank. More explicitly, the instrumental variable (IV) method is given by

$$A = \underset{B}{\operatorname{argmin}} \left| \sum_{i=0}^{N-1} (By_i - y_i)\zeta_i^* \right|_2 . \qquad (2.5)$$

The argument above gives us three conditions on our instruments to compute an unbiased estimate for A.

1. ζ_t is independent from n_t and n_{t+1},
2. ζ_t is independent from ν_t,
3. And the matrix

   $$G_0 = \mathbb{E}_P \left(\int_{\mathbb{R}^d} x\, \zeta^* d\mu \right)$$

   has full rank.

One of the potential drawbacks of instrumental variables is the necessity for an extra set of observables meeting conditions 1-3. However, we can often generate the instruments by taking time shifts of other observables. For the system described in (2.4) the three conditions above can be met by using $\zeta_t = y_1$, provided A is full rank. Using this choice of instruments, if we let

$$Z = \begin{bmatrix} y_{-1} & y_0 & y_1 & \dots & y_{N-2} \end{bmatrix}$$

then we can solve the instrumental variables regression (2.5) as

$$A = Y_1 Z^* \left(Y_0 Z^* \right)^{-1} .$$

Remark 2 *We note that we choose ζ to be an d-dimensional variable, so G_0 will be a square matrix. However, we could define ζ to be and l-dimensional variable for $l > d$. In this case, the regression (2.5) becomes overdetermined, and would be solved using the* *pseudoinverse of G_0. This is called an extended IV method. This can be beneficial, since the overdetermined problem may be better conditioned.*

2.6 DMD with Noisy Observables

In order to adapt the IV method for use in DMD, we must first define our requirements for "noisy observables."

Definition 12 *A noisy observable is a measurable map $\tilde{f} : M \times \Omega \to \mathbb{C}$. This means that the random function $\tilde{f}_\omega = \tilde{f}(\,\cdot\,, \omega) : M \to \mathbb{C}$ is a $\mathfrak{B}$ measurable function for almost every ω.*

For notation, we will always denote a noisy observable, $\tilde{f}$, with a tilde and denote the space of noisy observables as $\mathscr{H}$. We will also define f to be its mean:

$$f(x) = \int_\Omega \tilde{f}_\omega(x) dP.$$

In what follows, we will assume that f exists and is in $L^2(\mu)$. To avoid some clutter in the equations and algorithms, we will also denote the time samples of an observable with a hat: $\hat{f}(t) = \tilde{f}(x_t, \omega_t)$.

With these definitions, we can interpret f as the "true" observable on the system, whereas $\tilde{f}$ is the "measured" observable, which comes with some degree of uncertainty. We are interested in the evolution of f rather than $\tilde{f}$, since it depends only on the evolution in the state space and not the noise. Computing the DMD operator using standard EDMD directly with the data from $\tilde{f}$ can fit the model to the noise and give a poor approximation of the system.

2.6.1 DMD Observables and Instruments

In order to extend the DMD algorithms using IV methods, we will need to make use of two sets of observables. We will call them the "DMD observables" and "instruments" to avoid confusion. First, we will have the DMD observables, $\tilde{\mathbf{f}} = [\tilde{f}^1 \ ... \ \tilde{f}^k]^T$. These will be analogous to the observables used in standard DMD; they will form the basis for the finite section of the stochastic Koopman operator. The second set will be the instruments, $\tilde{\mathbf{g}} = [\tilde{g}^1 \ ... \ \tilde{g}^l]^T$. These will be used purely for regression purposes. The means of these variable will be denoted $\mathbf{f} = \mathbb{E}_P(\tilde{\mathbf{f}})$ and $\mathbf{g} = \mathbb{E}(\tilde{\mathbf{g}})$.

First, in order to approximate integrals from data, we will need some ergodicity assumptions on our observables. Namely, we will need time averages to converge in a similar sense to Lemma 1. In particular, we will need

$$\lim_{n\to\infty} \frac{1}{n} \sum_{j=0}^{n-1} \hat{\mathbf{f}}(t+j)\hat{\mathbf{g}}(t) = \int_M \int_\Omega \tilde{\mathbf{f}}_{\theta_j \omega}(T^j_\omega x)\tilde{\mathbf{g}}_\omega(x) dP(\omega) d\mu(x), \tag{2.6}$$

for two vector valued noisy observables $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$ and almost every initial condition (x_0, ω_0). In order for these time averages to be finite, we also require $\tilde{\mathbf{f}}, \tilde{\mathbf{g}} \in L^2(\mu \times P)$.

Remark 3 *While we make the ergodicity assumption for generality, we will show that (2.6) holds for observables with i.i.d. measurement noise and time delayed observables, the primary observables of interest in this paper. More generally, we can consider the skew product system Θ on $M \times \Omega$ given by $\Theta(x, \omega) = (T_\omega x, \theta\omega)$ and treat $\tilde{f}$ as an observable on $M \times \Omega$. If $\mu \times P$ is an ergodic measure for Θ, we can evaluate time averages as in (2.6).*

DMD Observables and Instruments

The DMD observables are the noisy observables $\tilde{f}^1, ..., \tilde{f}^k$. Their means, $f^1, ..., f^k$ will be the basis of the finite section of $\mathcal{K}$ produced using DMD. In order to compute the stochastic Koopman evolution of f^i, we will need to place further restrictions on $\tilde{f}^i$.

We require that the random function $\tilde{f}^i_{\omega_t}$ to be independent of T_{ω_s} for all $s < t$ (or equivalently, $\tilde{f}^i_{\theta_t\omega}$ is independent of T_ω for all $t \geq 0$). Roughly speaking, these conditions mean the random function $\tilde{f}_{\omega_t}$ cannot be predicted by the past of the dynamics on M. The independence condition gives us

$$\int_\Omega \tilde{f}_{\theta_j\omega}(T^j_\omega x)dP(\omega) = \int_\Omega \int_\Omega \tilde{f}_\psi(T^j_\omega x)dP(\psi)dP(\omega) = \int_\Omega f(T^j_\omega x)dP = \mathcal{K}^j f(x). \tag{2.7}$$

We will also need to select a set of instruments $\tilde{g}_1, ..., \tilde{g}_l$. Since the instruments are only needed for regression, we do not need to find their stochastic Koopman evolution. Instead, we will require that $\tilde{\mathbf{g}}_{\omega_t}$ is independent of T_{ω_t}, $\tilde{\mathbf{f}}_{\omega_t}$, and $\tilde{\mathbf{f}}_{\omega_{t+1}}$ (equivalently, $\tilde{\mathbf{g}}_\omega$ is independent from T_ω, $\tilde{\mathbf{f}}_\omega$, and $\tilde{\mathbf{f}}_{\theta_\omega}$). These independence conditions will allow us to compute the IV regression without bias. Finally, we will impose a rank requirement on $\int_M \mathbf{f}\,\mathbf{g}\,d\mu$ to ensure that the linear regression is fully determined. If $f^1, ..., f^k$ are linearly independent, this can be guaranteed by assuming $\text{span}\{f^k, ..., f^k\} \subset \text{span}\{g^1, ..., g^l\}$.

Assumption 1 *The observables $\tilde{\mathbf{f}}$ and instruments $\tilde{\mathbf{g}}$ satisfy the following:*

1. *The functions $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$ are in $L^2(\mu \times P)$ and ergodic in the sense that (2.6) holds,*

2. *The observable $\tilde{\mathbf{f}}_{\omega_t}$ is independent of T_{ω_s} for $s < t$,*

3. *The instrument $\tilde{\mathbf{g}}_{\omega_t}$ is independent of T_{ω_t}, $\tilde{\mathbf{f}}_{\omega_t}$, and $\tilde{\mathbf{f}}_{\omega_{t+1}}$.*

4. *The matrix*

$$G_0 = \int_M \mathbf{f}(x,\omega)\mathbf{g}(x,\omega)^* d\mu$$

has full rank, where $\mathbf{f} = \mathbb{E}_P(\tilde{\mathbf{f}}_\omega)$ *and* $\mathbf{g} = \mathbb{E}_P(\tilde{\mathbf{g}}_\omega)$.

Recalling that x_t is a Markov process with respect to the filtration $\{\mathcal{F}_t\}$, we can replace the independence assumptions between the variables and random maps with ones on the filtration.

Assumption 2 *The observables* $\tilde{\mathbf{f}}$ *and instruments* $\tilde{\mathbf{g}}$ *satisfy the following:*

1. *The functions* $\tilde{\mathbf{f}}$ *and* $\tilde{\mathbf{g}}$ *are in* $L^2(\mu \times P)$ *and ergodic in the sense that (2.6) holds,*

2. *The observable* $\tilde{\mathbf{f}}_{\omega_t}$ *is independent of* $\mathcal{F}_t$*,*

3. *The instrument* $\tilde{\mathbf{g}}_{\omega_t}$ *is* $\mathcal{F}_t$ *measureable, and*

4. *The matrix*

$$G_0 = \int_M \mathbf{f}(x,\omega)\mathbf{g}(x,\omega)^* d\mu$$

has full rank, where $\mathbf{f} = \mathbb{E}_P(\tilde{\mathbf{f}}_\omega)$ *and* $\mathbf{g} = \mathbb{E}_P(\tilde{\mathbf{g}}_\omega)$.

The conditions of assumption 2 imply those of 1, but are stronger. For example, condition 3 gives us $\tilde{\mathbf{g}}_{\omega_t}$ is independent of T_{ω_s} for all $s \geq t$, not just $s = t$. However, they are simple to state, and provide an interpretation of the dichotomy between the instruments and DMD observables. The instruments contain information that is available up to time t, while the DMD observables contain the information which becomes available at or after time t.

2.6.2 Noise Resistant DMD Algorithms

With the assumptions on our observables in place, we can now present the DMD algorithms for noisy systems.

Algorithm 3: Noise Resistant DMD

Let $\tilde{\mathbf{f}} \in \mathscr{H}^k$, and $\tilde{\mathbf{g}} \in \mathscr{H}^l$, $l \geq k$. As before, let $\hat{\mathbf{f}}(t) = \tilde{\mathbf{f}}(x_t, \omega_t)$ and $\hat{\mathbf{g}}(t) = \tilde{\mathbf{g}}(x_t, \omega_t)$ denote their samples along a trajectory at time t.

1: Construct the data matrices

$$X = \begin{bmatrix} \hat{\mathbf{f}}(0) & \hat{\mathbf{f}}(1) & \dots & \hat{\mathbf{f}}(n-1) \end{bmatrix},$$

$$Y = \begin{bmatrix} \hat{\mathbf{f}}(1) & \hat{\mathbf{f}}(2) & \dots & \hat{\mathbf{f}}(n) \end{bmatrix},$$

and

$$Z = \begin{bmatrix} \hat{\mathbf{g}}(0) & \hat{\mathbf{g}}(1) & \dots & \hat{\mathbf{g}}(n-1) \end{bmatrix}.$$

2: Form the matrices $\tilde{G}_0 = \frac{1}{n} X Z^*$ and $\tilde{G}_1 = \frac{1}{n} Y Z^*$.

3: Compute the matrix

$$C = \tilde{G}_1 \tilde{G}_0^{\dagger}.$$

4: Compute the eigenvalues and left and right eigenvectors, (λ_i, w_i, v_i) of C. The dynamic eigenvalues are λ_i, the dynamic modes are v_i, and the numerical eigenfunctions are given by

$$\hat{\phi}_i = w_i^T X.$$

Proposition 2 *Let $\tilde{\mathbf{f}} \in \mathscr{H}^k$ and $\tilde{\mathbf{g}} \in \mathscr{H}^l$ satisfy assumption 1 or 2. Suppose the components of $\mathbf{f}$, $f_1, ..., f_k$, span a k-dimensional invariant subspace, $\mathscr{F}$, of $\mathcal{K}$. Then the matrix C generated by Algorithm 3 converges to the restriction of $\mathcal{K}$ to $\mathscr{F}$ as $n \to \infty$.*

Proof: Let $\mathbf{K}$ be the restriction of $\mathcal{K}$ to $\mathscr{F}$. Let $\tilde{G}_{0,n}$ and $\tilde{G}_{1,n}$ be the matrices generated in Algorithm 3 with n data points. From the assumption, $\tilde{\mathbf{g}}_\omega$ is independent from $\tilde{\mathbf{f}}$, $\tilde{\mathbf{f}}_{\theta_\omega}$, and T_ω. Using (2.7), define

$$G_0 = \int_M \int_\Omega \tilde{\mathbf{f}}_\omega(x)\tilde{\mathbf{g}}^*_\omega(x) dP d\mu = \int_M \int_\Omega \tilde{\mathbf{f}}_\omega(x) dP \int_\Omega \tilde{\mathbf{g}}^*_\omega(x) dP d\mu = \int_M \mathbf{f}\,\mathbf{g}^* d\mu \tag{2.8}$$

and

$$G_1 = \int_M \int_\Omega \tilde{\mathbf{f}}_{\theta\omega}(T_\omega x)\tilde{\mathbf{g}}^*_\omega(x) dP d\mu = \int_M \int_\Omega \tilde{\mathbf{f}}(T_\omega x) dP \int_\Omega \tilde{\mathbf{g}}^*_\omega(x) dP d\mu = \mathbf{K}\int_M \mathbf{f}\,\mathbf{g}^* d\mu. \tag{2.9}$$

This gives us $\mathbf{K} = G_1 G_0^\dagger$ since G_0 is full row rank. We will show that $\tilde{G}_{0,n} \to G_0$ and $\tilde{G}_{1,n} \to G_1$ as $n \to \infty$. Taking the limit of $G_{0,n}$ with (2.6) and using (2.8), we have

$$\lim_{n\to\infty} \tilde{G}_{0,n} = \lim_{n\to\infty} \frac{1}{n}\sum_{m=0}^{n-1} \hat{\mathbf{f}}(m)\hat{\mathbf{g}}^*(m) = \int_M \int_\Omega \tilde{\mathbf{f}}_\omega(x)\tilde{\mathbf{g}}^*_\omega(x)\, dP d\mu = G_0$$

and similarly $\tilde{G}_{1,n} \to G_1$ using (2.9). Since G_0 has full rank and $\tilde{G}_{0,n} \to G_0$, we have $\tilde{G}_{0,n}^\dagger \to G_0^\dagger$, so $\tilde{G}_{1,n}\tilde{G}_{0,n}^\dagger \to \mathbf{K}$. ■

It follows from Proposition 2 that the eigenvalues and eigenvectors of C go to those of $\mathbf{K}$. Therefore, the dynamic eigenvalues limit to Koopman eigenvalues. The numerical eigenfunctions, however, are more complicated. If w_i is a left eigenvector of $\mathbf{K}$, we have $w_i^T \mathbf{f}$ is a Koopman eigenfunction. The numerical eigenfunctions, however, limit to $w_i^T X$, which a sampling of $w_i^T \tilde{\mathbf{f}}$. In this regard, the numerical eigenfunction is a sampling of an eigenfunction with some zero mean noise added to it.

2.6.3 Observables with i.i.d. Measurement Noise

Often, when measuring an observable on a system, the measurement will be imprecise. The error in the measurement are often modeled as an i.i.d. random variable. We call an observable with this type of noise an observable with measurement noise:

Definition 13 *A noisy observable, $\tilde{f}$, is an observable with i.i.d. measurement noise if $\tilde{f}_{\theta_t}\omega$ is an i.i.d. random function and is independent of the random maps $T_{\theta_s\omega}$ for all s.*

Let $f = \mathbb{E}_P(\tilde{f}_\omega)$. We note that for any given ω, the measurement error,

$$\tilde{e}_\omega(x) = \tilde{f}_\omega(x) - f(x),$$

can vary over the state space M; it does not need to be a constant additive noise. Since $\tilde{f}_{\omega_t}$ is an i.i.d. random variable and independent of T_{ω_t} for all t, the ordered pair $(x_t, \tilde{f}_{\omega_t}) \in M \times L^2(M)$ is an ergodic process, with ergodic measure $\nu = \mu \times \tilde{f}_*(P)$, where $\tilde{f}_*(P)$ is the pushforward of P. This allows us to evaluate the time averages as in (2.6). The proof of this follows from the lemma below and the fact that i.i.d. processes are mixing ([20], Theorem 4, page 143).

Lemma 2 *Let x_t and y_t be independent stationary processes. If x_t is ergodic and y_t is mixing, then (x_t, y_t) is ergodic.*

Proof: The result follows from Theorem 6.1 on page 65 of [58], where we can represent the processes as a measure preserving shifts on the space of sequences of x_t and y_t ([58], page 6). ■

If the components of $\tilde{\mathbf{f}}$ are observables with measurement noise, it turns out we don't need second observable to use in Algorithm 3. Instead, we can use a time shift of $\tilde{\mathbf{f}}$ to generate $\tilde{\mathbf{g}}$. The i.i.d. property of $\tilde{\mathbf{f}}$ will give us the independence properties we need.

Corollary 1 *Suppose $\tilde{\mathbf{f}} \in \mathscr{H}^k$ is a vector valued observable with i.i.d. measurement noise, and the components of $\mathbf{f} = \mathbb{E}_P(\tilde{\mathbf{f}}_\omega)$ span a k-dimensional invariant subspace, $\mathscr{F}$, of $L^2(\mu)$. Suppose further that the restriction of $\mathcal{K}$ to $\mathscr{F}$ has full rank. Then Algorithm 3 converges setting $\hat{\mathbf{g}}(t) = \hat{\mathbf{f}}(t-1)$.*

Proof: Let $\mathbf{K}$ be the resriction of $\mathcal{K}$ to $\mathscr{F}$. By Lemma 2, $(x_t, \tilde{\mathbf{f}}_{\omega_t})$ is an ergodic stationary sequence. Then, using ergodicity and the independence properties of $\tilde{\mathbf{f}}$, we have

$$\begin{aligned}\lim_{n\to\infty}\frac{1}{n}\sum_{m=1}^{n}\hat{\mathbf{f}}(m)\hat{\mathbf{g}}^*(m) = \lim_{n\to\infty}\frac{1}{n}\sum_{m=0}^{n-1}\hat{\mathbf{f}}(m)\hat{\mathbf{f}}^*(m-1) &= \int_M\int_\Omega \tilde{\mathbf{f}}_{\theta\omega}(T_\omega^{j+1}x)\tilde{\mathbf{f}}_\omega^*(x)\, dP d\mu \\ &= \int_M\int_\Omega \tilde{\mathbf{f}}_{\theta\omega}(T_\omega x)dP\int_\Omega \tilde{\mathbf{f}}_\omega(x)dPd\mu = \mathbf{K}\int_M \mathbf{f}\,\mathbf{f}^* d\mu,\end{aligned}$$

which has full rank since $\mathbf{K}$ has full rank. Similarly,

$$\lim_{n\to\infty}\frac{1}{n}\sum_{m=1}^{n}\hat{\mathbf{f}}(m+1)\hat{\mathbf{g}}^*(m) = \lim_{n\to\infty}\frac{1}{n}\sum_{m=0}^{n-1}\hat{\mathbf{f}}(m+1)\hat{\mathbf{f}}^*(m-1) = \mathbf{K}^2\int_M \mathbf{f}\,\mathbf{f}^* d\mu.$$

The rest of the proof follows Proposition 2. ■

Remark 4 *It is useful to note that if T_ω and θ were invertible (i.e. the RDS is defined on two-sided time), we would be able to define $\tilde{\mathbf{g}}_\omega = \tilde{\mathbf{f}}_{\theta_{-1}\omega} \circ (T_\omega^{-1})$, and $\tilde{\mathbf{g}}$ would meet the conditions of assumption 1 exactly. However, if they are not invertible, we cannot necessarily define $\tilde{\mathbf{g}}_\omega \in L^2(M)$ explicitly since T_ω may not be invertible. However, since we are still able to evaluate time averages, the proof is nearly identical. Alternatively, we could let $\mathcal{F}_t$ be the sigma algebra generated by the past of $(x_t, \tilde{\mathbf{f}}_{\omega_{t-1}})$,*

$$\mathcal{F}_t = \sigma\{(x_s, \tilde{\mathbf{f}}_{\omega_{s-1}}) : s \leq t\}.$$

With this definition of the filtration, the instruments and DMD observables meet the *condition 2 and 3 of assumption 2 exactly.*

2.7 Time Delayed Observables and Krylov Subspace Methods

Another important type of noisy observable are time delayed observables. Allowing time delayed observables in DMD is useful for two reasons. First, time delays allow us to enrich our space of observables. Oftentimes, there are functions on our state space which cannot be measured by a certain set of observables, but can be observed if we allow time delays. For example, the velocity of a moving mass cannot be observed by any function on the position, but can be approximated using the position at two different times. Second, using time delays allows us to identify an invariant (or nearly invariant) subspace spanned by the Krylov sequence $f, \mathcal{K}f, ..., \mathcal{K}^{k-1}f$.

Of particular interest is an analogue of Hankel DMD for random systems, which uses a Krylov sequence of observables to generate our finite subspace. With Hankel DMD, we use a single observable, f, and its time delays to approximate the sequence $f, \mathcal{K}f, ..., \mathcal{K}^{k-1}f$. If $\tilde{f}$ is an observable with measurement noise (or has no noise), we can define

$$\tilde{\mathbf{f}}(x,\omega) = \begin{bmatrix} \tilde{f}(x,\omega) & \tilde{f}(T_\omega x, \theta\omega) & \dots & \tilde{f}(T_\omega^{k-1}x, \theta_{k-1}\omega) \end{bmatrix}^T.$$

By (2.7), its mean is

$$\int_\Omega \tilde{\mathbf{f}}\, dP = \begin{bmatrix} f & \mathcal{K}f & \dots & \mathcal{K}^{k-1}f \end{bmatrix}^T,$$

where $f = \mathbb{E}_P(\tilde{f})$. We can then use time delays of $\tilde{f}$ to approximate the Krylov sequence $f, \mathcal{K}f, ..., \mathcal{K}^{k-1}f$. Additionally, if we set $\tilde{\mathbf{g}}(t) = \tilde{\mathbf{f}}(t-k)$ in Algorithm 3, we will have the necessary independence conditions, and the time averages will converge as in (2.6) due to the pair $(x_t, \tilde{f}_{\omega_t})$ being an ergodic stationary variable.

Corollary 2 *(Noise Resistant Hankel DMD) Let $\tilde{f}$ be an observable with measurement noise, with time samples $\hat{f}(t) = \tilde{f}(x_t, \omega_t)$. Let its mean, f, be such that the Krylov sequence $f, \mathcal{K}f, ..., \mathcal{K}^{k-1}f$ spans a k-dimensional invariant subspace $\mathscr{F}$ and the restriction of $\mathcal{K}$ to $\mathscr{F}$ has full rank. Let*

$$\hat{\mathbf{f}}(t) = \begin{bmatrix} \hat{f}(t) & \hat{f}(t+1) & \dots & \hat{f}(t+k-1) \end{bmatrix}^T,$$

and

$$\hat{\mathbf{g}}(t) = \hat{\mathbf{f}}(t-k) = \begin{bmatrix} \hat{f}(t-k) & \hat{f}(t-k+1) & \dots & \hat{f}(t-1) \end{bmatrix}^T.$$

Then the matrix A generated by Algorithm 3 converges to the restriction of $\mathcal{K}$ to $\mathscr{F}$. If $\tilde{f}$ has no noise (i.e. $\tilde{f}(x, \omega) = f(x)$) we can use

$$\hat{\mathbf{g}}'(t) = \hat{\mathbf{f}}(t-k+1) = \begin{bmatrix} \hat{f}(t-k+1) & \hat{f}(t-k+2) & \dots & \hat{f}(t) \end{bmatrix}^T.$$

We refer to Corollary 2 as a variant of Hankel DMD for random systems since the X, Y, and Z matrices in Algorithm 3 will be Hankel matrices and it generates a Krylov subspace of $\mathcal{K}$. For a different choice of $\tilde{\mathbf{g}}$ (i.e. $\tilde{\mathbf{g}} = \tilde{\mathbf{f}}$), this is equivalent to Hankel DMD.

Proof: Using (2.7), we can see that the components of $\mathbf{f}$ are $f, \mathcal{K}f, ..., \mathcal{K}^{k-1}f$, which spans $\mathscr{F}$. Additionally, using the independence properties of $\tilde{f}$, we have $\tilde{\mathbf{f}}_{\omega_t}$ and $\tilde{\mathbf{f}}_{\omega_{t+s}}$ are independent for $s \geq k$. Since $(x_t, \tilde{f}_{\omega_t})$ is ergodic by Lemma 2, we can take the time averages

$$\lim_{n\to\infty} \frac{1}{n} \sum_{m=k}^{n+k-1} \mathbf{f}(m)\mathbf{g}^*(m) = \lim_{n\to\infty} \frac{1}{n} \sum_{m=0}^{n-1} \mathbf{f}(m+k)\mathbf{f}^*(m) = \int_M \int_\Omega \tilde{\mathbf{f}}_{\theta^k \omega}(T^k_\omega x)\tilde{\mathbf{f}}^*_\omega(x) dP d\mu$$
$$= \int_M \int_\Omega \tilde{\mathbf{f}}_{\theta_k \omega}(T^k_\omega x)\tilde{\mathbf{f}}^*(x) dP d\mu = \mathbf{K}^k \int_M \mathbf{f}\,\mathbf{f}^* d\mu,$$

which has full rank since $\mathbf{K}$ has full rank. Similarly, we can take the time average

$$\lim_{n\to\infty}\frac{1}{n}\sum_{m=k}^{n+k-1}\mathbf{f}(m+1)\mathbf{g}^*(m) = \mathbf{K}^{k+1}\int_M \mathbf{f}\,\mathbf{f}^* d\mu,$$

and the rest of the proof follows Proposition 2. If $\tilde{f}_\omega = f$, $\tilde{\mathbf{f}}_{\omega_t}$ and $\tilde{\mathbf{f}}_{\omega_{t+k-1}}$ are independent, and we can take the time averages using $\hat{\mathbf{g}}(t) = \hat{\mathbf{f}}(t-k+1)$.

■

Remark 5 *Similar to remark 4, we could define a filtration to meet the conditions of assumption 2. However, since we need to show the rank condition anyways, this does not do much to shorten the proof.*

Corollary 2 allows us to compute an approximation of $\mathcal{K}$ using the data from a single observable evaluated along a single trajectory. However, the method does not require that the we only use time delays of a single observable. In general, even if $\tilde{f}$ is vector valued, we can take time delays of $\tilde{f}$ as in Corollary 2 so long as we span the proper subspace. The instruments, $\tilde{\mathbf{g}}$, is also generated in the same way.

2.8 Conditioning of Algorithm 3

Asymptotically, the convergence rate of Algorithm 3 is governed by the rate at which $G_{0,n}$ and $G_{1,n}$ converges to G_0 and G_1, as defined in the proof of Proposition 2. This is governed by the convergence rate of ergodic sampling. However, Algorithm 3 also requires the pseudo-inversion of $G_{0,n} \approx G_0$. If the matrix G_0 is ill-conditioned, small errors in the time averages approximations of G_0 and G_1 can cause large errors in our DMD operator. The condition number of G_0, $\kappa(G_0)$, can become large if either set of observables, $f_1, ..., f_k$ or $g_1, ..., g_l$, are close to being linearly dependent.

Both of these issues arise particularly often when using Hankel DMD. With Hankel DMD, we use the basis $f, \mathcal{K}f, ..., \mathcal{K}^{k-1}f$ as our basis for $\mathscr{F}$. This is often a poor choice of basis, as f and $\mathcal{K}f$ may be close to being linearly dependent. This is particularly the case when data from a continuous time system is sampled with a short period, such as from a discretization of an ODE or SDE. Similarly, if j is large or $\mathcal{K}$ has eigenvalues close to zero, $\mathcal{K}^j f$ and $\mathcal{K}^{j+1} f$ may be close to being linearly dependent, which will cause conditioning issues.

2.8.1 SVD Based Algorithms

To combat these conditioning issues, we have some leeway in the observables we choose for $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$. Looking at G_0, we have

$$G_0 = \int_M \mathbf{g}\,\mathbf{f}^* \, d\mu = \int_M \begin{bmatrix} f^1 & f^2 & \dots & f^k \end{bmatrix}^T \begin{bmatrix} g^{1*} & g^{2*} & \dots & g^{l*} \end{bmatrix} d\mu. \tag{2.10}$$

Ideally, $\{g^1, ..., g^l\}$ and $\{f^1, ..., f^k\}$ would be orthonormal bases for $\mathscr{F}$, so $\kappa(G_0)$ would be 1. However, we rarely can choose such bases a priori. Instead, we can try to augment $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$ with extra observables and use the singular value decomposition to choose k observables which form a better conditioned basis for $\mathscr{F}$, similar to Algorithm 2. This brings us to the SVD implementation of Algorithm 3.

Algorithm 4: SVD implemented Noise Resistant DMD

Let $\tilde{\mathbf{f}} \in \mathscr{H}^{l_1}$, and $\tilde{\mathbf{g}} \in \mathscr{H}^{l_2}$, $l_1, l_2 \geq k$ be noisy observables on our system. Let $\hat{\mathbf{f}}(t) = \tilde{\mathbf{f}}(x_t, \omega_t)$ and $\hat{\mathbf{g}}(t) = \tilde{\mathbf{g}}(x_t, \omega_t)$ denote the time samples of the observables.

1: Construct the data matrices

$$X = \begin{bmatrix} \hat{\mathbf{f}}(0) & \hat{\mathbf{f}}(1) & \dots & \hat{\mathbf{f}}(n-1) \end{bmatrix},$$

$$Y = \begin{bmatrix} \hat{\mathbf{f}}(1) & \hat{\mathbf{f}}(2) & \dots & \hat{\mathbf{f}}(n) \end{bmatrix},$$

and

$$Z = \begin{bmatrix} \hat{\mathbf{g}}(0) & \hat{\mathbf{g}}(1) & \dots & \hat{\mathbf{g}}(n-1) \end{bmatrix}.$$

2: Form the matrices $\tilde{G}_0 = \frac{1}{n} X Z^*$ and $\tilde{G}_1 = \frac{1}{n} Y Z^*$.

3: Compute the truncated SVD of $\tilde{G}_0$ using the first k singular values:

$$\tilde{G}_0 \approx W_k S_k V_k^*.$$

5: Form the matrix

$$A = S_k^{-1} W_k^* \tilde{G}_1 V_k.$$

6: Compute the eigenvalues and left and right eigenvectors, (λ_i, w_i, u_i) of A. The dynamic eigenvalues are λ_i, the dynamic modes are

$$v_i = W_k S_k u_i,$$

and the numerical eigenfunctions are

$$\hat{\phi}_i = w_i S_k^{-1} W_k^* X.$$

Similar to Algorithm 2, Algorithm 4 uses the SVD to choose a basis of observables to use in Algorithm 1. It is equivalent to performing Algorithm 3 using data from the observable $(S_k^{-1}W_k^*)\tilde{\mathbf{f}}$, while leaving the instruments $\tilde{\mathbf{g}}$ unchanged. It is important to note that Algorithm 4 uses the components of $(S_k^{-1}W_k^*)\mathbf{f}$ to as a basis for $\mathscr{F}$ where $\mathbf{f} = \mathbb{E}_P(\tilde{\mathbf{f}})$ as usual. When we add observables to $\tilde{\mathbf{f}}$, we must ensure that we stay within our invariant subspace. One way to guarantee this is to use time delays of our original observables.

2.8.2 Extended Instrumental Variables

Typically, augmenting $\tilde{\mathbf{f}}$ with extra observables and using Algorithm 4 to truncate the singular values is an effective way to improve the conditioning of the problem. However, we have an alternate tool at our disposal. While each component of $\mathbf{f}$ must lie within $\mathscr{F}$, the components of the instrument $\mathbf{g}$ can be arbitrary, and we do not need to take an SVD to truncate the extra observables in $\mathbf{g}$. Since we do not need to worry about leaving our invariant subspace, we can add arbitrary functions of $\tilde{\mathbf{g}}$ (e.g. powers of $\tilde{\mathbf{g}}$) to our instruments and still expect convergence. This corresponds to an extended instrumental variables method (see [40], chapter 7.6). However, while this can improve conditioning, it also can slow down the convergence of the time averages, and should only be done when the error stems from poor conditioning.

2.9 Numerical Examples

In this section, we will test the various DMD algorithms presented in this paper using both observables with measurement noise and time delayed observables. For each system and each DMD method, we generate five realizations of the DMD operator and compare the eigenvalues with analytically obtained true (or approximate) eigenvalues of the stochastic Koopman eigenvalues. For each system, we compare the NR-DMD algorithm with both standard DMD and TLS-DMD, the last of which is unbiased for deterministic systems with measurement noise. Since the purpose of this paper is to provide a new algorithm that is provably unbaised, we will use parameters for each algorithm which ensures that the regression is sufficiently well conditioned. Comparisons on the speed of convergence and numerical stability of various DMD algorithms not the primary purpose of this paper.

2.9.1 Random Rotation on a Circle

Consider a rotation on the circle. The dynamical system is defined by

$$x_{t+1} = x_t + \nu, \tag{2.11}$$

where $\nu \in S^1$. If we perturb (2.11) by adding noise to the rotation rate we obtain the random system

$$x_t + 1 = x_t + \nu + \pi(\omega_t) \tag{2.12}$$

where $\pi(\omega_t) \in S^1$ is an i.i.d. random variable. For the stochastic Koopman operator associated with (2.12), the functions $\varphi_n(x) = e^{inx}$ are eigenfunctions with eigenvalues $\lambda_i = \mathbb{E}(e^{in(\nu+\pi(\omega)})$, since

$$\mathcal{K}\varphi_i(x) = \mathbb{E}(\varphi_i(T_\omega x)) = \int_\Omega e^{in(x+\nu+\pi(\omega))}dP = e^{inx}\int_\Omega e^{in(\nu+\pi(\omega))}dP = \varphi_i(x)\lambda_i.$$

We can compare these eigenvalues with the results obtained from our different DMD algorithms. We will set our system parameter to $\nu = \frac{1}{2}$ and draw $\pi(\omega_t)$ from the uniform distribution over $[-\frac{1}{2}, \frac{1}{2}]$. In this case the eigenvalues are $\lambda_i = \frac{i-ie^{in}}{n}$. For the first test, we will compare EDMD, TLS-DMD and NR-DMD using a set of observables with measurement noise. We will let our observable be

$$\hat{\mathbf{f}}(t) = [\sin(x_t), ..., \sin(5x_t), \cos(x_t), ..., \cos(5x_t)]^T + \mathbf{m}(t), \tag{2.13}$$

where $\mathbf{m}(t) \in [-0.5, 0.5]^{10}$ is measurement noise drawn from the uniform distribution. EDMD and TLS-DMD are applied directly to the data from measurements of $\tilde{\mathbf{f}}$ and for NR-DMD we let $\tilde{\mathbf{g}}(t) = \tilde{\mathbf{f}}(t-1)$.

For the second test, we let $f = \sin(x) + \sin(2x) + \sin(3x)$, and use time delays to generate $\hat{\mathbf{f}}$:

$$\hat{\mathbf{f}}(t) = \begin{bmatrix} f(x_t) & f(x_{t+1}) & \dots & f(x_{t+d}). \end{bmatrix}^T . \tag{2.14}$$

To perform Hankel DMD, we take five time delays ($d = 5$ in (2.14)) to generate $\tilde{\mathbf{f}}$, and use the data directly in EDMD and TLS-DMD. However, if we try to perform Noise Resistant Hankel DMD using these observables, Algorithm 3 is poorly conditioned and and the eigenvalues are inaccurate. Instead, we use 24 time delays of $\tilde{f}$ to generate $\tilde{\mathbf{f}}$ (setting $d = 24$ in (2.14) and letting $\hat{\mathbf{g}}(t) = \hat{\mathbf{f}}(t-24)$), and use Algorithm 4, the SVD implemented NR-DMD, to truncate to the leading six singular values. Finally, we use Algorithm 4 again using only eight time delays to generate $\tilde{\mathbf{f}}$, but augment $\hat{\mathbf{g}}$ with extra instruments to improve conditioning. We let $\hat{\mathbf{g}}$ to contain the observables $\hat{f}, \hat{f}^2$, and $\hat{f}^3$, as well as 42 time shifts of each of these functions:

$$\hat{\mathbf{g}} = \begin{bmatrix} \hat{f}(t-42) & \hat{f}(t-42)^2 & \hat{f}(t-42)^3 & \dots & \hat{f}(t) & \hat{f}(t)^2 & \hat{f}(t)^3 \end{bmatrix}^T .$$

As can be seen in Figure 2.9.1, EDMD and TLS-DMD fail to accurately approximate the eigenvalues of $\mathcal{K}$ in both tests. For the first test, NR-DMD gives accurate approximations to the eigenvalues of $\mathcal{K}$. Approximating the stochastic Koopman operator using the time delayed observables, (2.14) is more difficult because the conditioning of the matrix

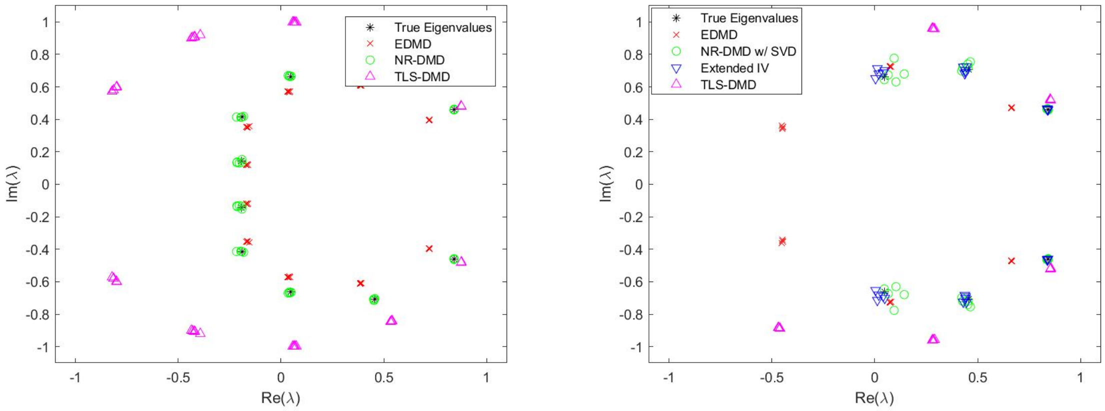

Figure 2.1: (Left) Outputs of EDMD and NR-DMD using (2.13) as observables on (2.12) with 25 000 data points. EDMD and TLS-DMD show a clear bias in the approximate eigenvalues while NR-DMD captures them accurately.
(Right) DMD outputs from EDMD and NR-DMD using (2.14) as observables on (2.12) with 25 000 data points. NR-DMD is implemented using Algorithm 4 to improve conditioning, and is performed a second time with extended instrumental varialbles. EDMD and TLS-DMD show a bias in the eigenvalues while NR-DMD gives an unbiased approximation of the eigenvalues in both cases.
Each algorithm is run five times on different sample trajectories.

G_0 is very poor, which amplifies the errors in our time averages. However, including extra time delays and using Algorithm 4 to truncate to the leading singular values obtains accurate results. Further, the precision is increased when we augment $\tilde{\mathbf{g}}$ with extra instruments.

2.9.2 Linear System with Additive Noise

Consider the linear system in $\mathbb{R}^4$:

$$\mathbf{x}(t+1) = \begin{bmatrix} 0.75 & 0.5 & 0.1 & 2 \\ 0 & 0.2 & 0.8 & 1 \\ 0 & -0.8 & 0.2 & 0.5 \\ 0 & 0 & 0 & -0.85 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} = A\mathbf{x}(t). \tag{2.15}$$

We can perturb (2.15) by perturbing the matrix A with a random matrix δ and adding a random forcing term b. We obtain the random system

$$\mathbf{x}(t+1) = (A + \delta_t)\mathbf{x}(t) + b_t, \tag{2.16}$$

where $b_t \in \mathbb{R}^4$ and $\delta_t \in \mathbb{R}^{4\times 4}$ are i.i.d. random variables. Let $(w_i, \lambda_i), i = 1, ..., 4$ be the left eigenpairs of A. If b_t and δ_t are assumed to have zero mean, $w_i^T\mathbf{x}$ is an eigenfunction of $\mathcal{K}$ with eigenvalue λ_i. For this example we will assume each component of b_t and δ_t is drawn from randomly from a uniform distribution. The components of b_t will be drawn from $[-0.5, 0.5]$ while those of δ_t will be drawn from $[-0.25, 0.25]$. As before, we will test EDMD, TLS-DMD, and NR-DMD using observables with measurement noise and time delayed observables. For the first test, we will use state observables with Gaussian measurement noise:

$$\hat{\mathbf{f}}(t) = \mathbf{x}(t) + \mathbf{m}(t) \tag{2.17}$$

where each component of $\mathbf{m}(t) \in \mathbb{R}^4$ is drawn from the standard normal distribution. As before, will let $\hat{\mathbf{g}}(t) = \hat{\mathbf{f}}(t-1)$.

For the second test, to generate the time delayed observables, we only use the first component of the state, $\hat{f}(t) = x_1(t)$, and use three time delays:

$$\hat{\mathbf{f}}(t) = \begin{bmatrix} \hat{f}(t) & \hat{f}(t+1) & \hat{f}(t+2) & \hat{f}(t+3) \end{bmatrix}. \tag{2.18}$$

We will apply Algorithm 1 directly to this matrix, while for Algorithm 3 we let $\hat{\mathbf{g}}(t) = \hat{\mathbf{f}}(t-3)$.

Figure 2.9.2 shows that the eigenvalues generated by EDMD and TLS-DMD again fail to accurately approximate those of $\mathcal{K}$. However, for both sets of observables, NR-DMD estimates the eigenvalues of $\mathcal{K}$ well. Since we did not run into conditioning issues, we did

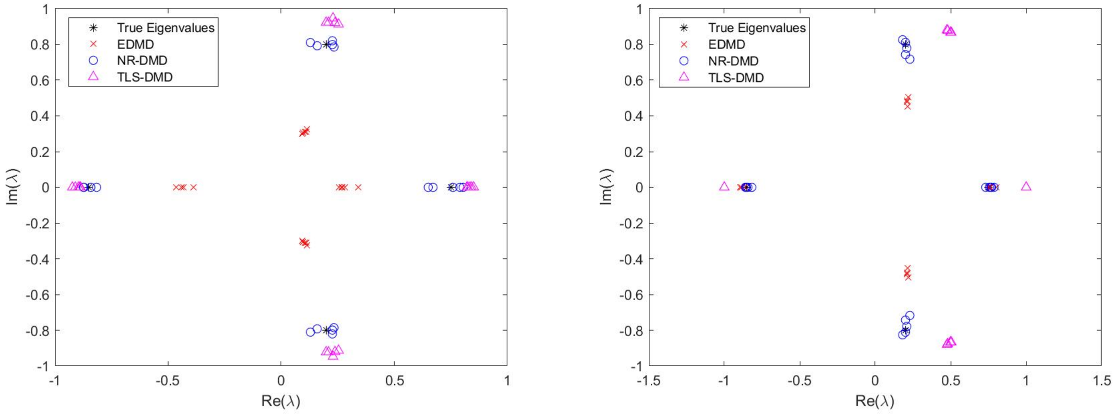

Figure 2.2: (Left) Outputs of EDMD, TLS-DMD, NR-DMD using state observables with measurement noise (2.17) on 5 000 data points from (2.16).
(Right) Outputs of Algorithm EDMD, TLS-DMD, and NR-DMD using (2.18) as observables on (2.16) with 5 000 data points. For both cases, NR-DMD is unbiased in approximating the eigenvalues while EDMD exhibits a clear bias.
Each algorithm is run five times on different sample trajectories.

not test the results using Algorithm 4 or extended instrumental variables.

2.9.3 Stuart Landau Equations

Consider the stochastic Stuart Landau equations defined by

$$dr = (\delta r - r^3 + \frac{\epsilon^2}{r})dt + \epsilon dW_r \tag{2.19}$$

$$d\theta = (\gamma - \beta r^2)dt + \frac{\epsilon}{r}dW_\theta, \tag{2.20}$$

where W_r and W_θ satisfy

$$dW_r = \cos\theta\, dW_x + \sin\theta\, dW_y$$

$$dW_\theta = -\sin\theta\, dW_x + \cos\theta\, dW_y$$

for independent Wiener processes dW_x and dW_y. It was shown in [71] that for small ϵ and $\delta > 0$, the (continuous time) stochastic Koopman eigenvalues are given by

$$\lambda_{l,n} = \begin{cases} -\frac{n^2\epsilon^2(1+\beta^2)}{2\delta} + in\omega_0 + \mathcal{O}(\epsilon^4) & l = 0 \\ -2l\delta + in\omega_0 + \mathcal{O}(\epsilon^2) & l > 0, \end{cases}$$

where $\omega_0 = \gamma - \beta\delta$.

Let $\gamma = \beta = 1$, $\delta = 1/2$, and $\epsilon = 0.05$ in (2.19) and (2.20). Define the observables

$$f_k(r,\theta) = e^{ik(\theta-(\log(2r))}.$$

First, we will let

$$\hat{\mathbf{f}}(t) = [f_1(x_t), f_{-1}(x_t), ..., f_6(x_t), f_{-6}(x_t)]^T + \mathbf{m}_1(t) + i\mathbf{m}_2(t), \tag{2.21}$$

where each component of $\mathbf{m}_1(t)$ and $\mathbf{m}_2(t)$ is drawn independently from a normal distribution with mean 0 and variance 1/4. For NR-DMD, we let $\hat{\mathbf{g}}(t) = \hat{\mathbf{f}}(t-1)$. The (continuous time) eigenvalues generated by the DMD algorithms are shown from a simulation with a time length of 1,000 with a time step of 0.05 (20,000 data points)in Figure 2.9.3.

To test Hankel DMD, we use the observable

$$f = \sum_{k=1}^{6}(f_k + f_{-k}),$$

and let $\tilde{\mathbf{f}}$ contain f and d time delays of f:

$$\hat{\mathbf{f}}(t) = \begin{bmatrix} f(x_t) & f(x_{t+1}) & \dots & f(x_{t+d}) \end{bmatrix}. \tag{2.22}$$

Due to the poor conditioning of Algorithms 1 and 3, the eigenvalues they generate are highly inaccurate, so we instead use the SVD implementation of DMD and NR-DMD. In each case, we let $d = 399$ and truncate the SVD to the leading 12 singular values. As usual, we let $\hat{\mathbf{g}} = \hat{\mathbf{f}}(t-d)$ for Algorithm 4. The results shown in Figure 2.9.3 are from a simulation with 200,000 data points and a time step of 0.05.

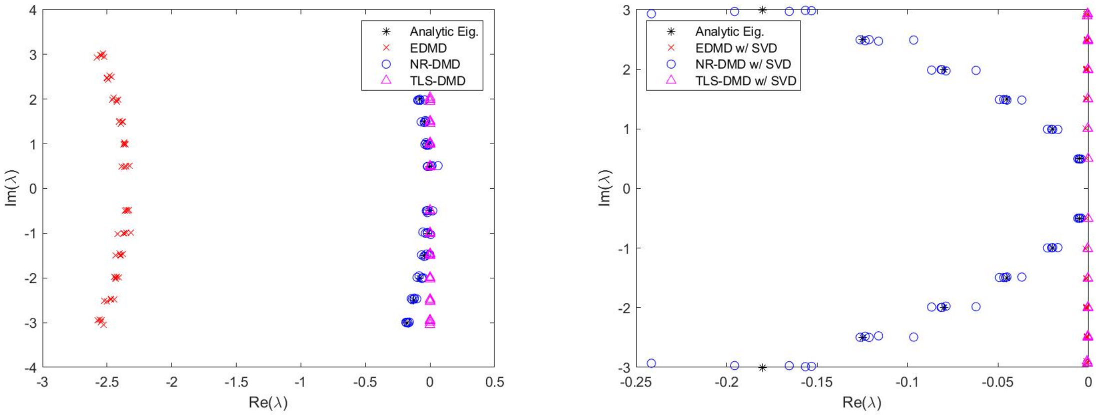

Figure 2.3: (Left) Outputs of EDMD, TLS-DMD, and NR-DMD using observables with measurement noise (2.21). The data is taken over 20 000 data points from (2.19) and (2.20) with a time step of 0.05. The eigenvalues produced by DMD are biased towards the left hand plane and TLS-DMD biases them towards the imaginary axis. The eigenvalues captured by NR-DMD are accurate. (Right) Outputs of SVD implemented EDMD, TLS-DMD, and NR-DMD using (2.22) as observables on (2.19) and (2.20). The DMD operator is truncated to the leading 12 singular values. The Algorithms used 200 000 data points with a time step of 0.05. NR-DMD captures most of the eigenvalues without bias while EDMD and TLS-DMD bias all eigenvalues towards the imaginary axis.
Each algorithm is run five times on different sample trajectories.

As can be seen in Figure 2.9.3, EDMD exhibits a clear bias towards the left of the complex plane and TLS-DMD biases them towards the imaginary axis when using observables with measurement noise, although it appears to accurately estimate the imaginary part of the eigenvalue. NR-DMD, on the other hand, appears to give a mostly accurate spectrum. When using time delayed observables for Hankel DMD, Algorithms 1 and 3 were very poorly conditioned, and gave eigenvalues far outside the windows shown in

Figure 2.9.3. Instead, SVD implementations of DMD were used for EDMD, TLS-DMD, and NR-DMD. The DMD operators were truncated to the 12 dominant singluar values. We again see that the imaginary parts of the eigenvalues seem to be captured, but the real parts are all biased to the right for EDMD and TLS-DMD. The SVD implemented NR-DMD, however, again captures the correct spectrum, but with some error for the most dissipative eigenvalues.

2.10 Permissions and Attributions

1. The contents of chapter 2 are the result of a collaboration with Dr. Igor Mezić. A previous version was published in SIAM Journal on Applied Dynamical Systems [73]. It is reproduced here with the permission of SIAM.

2. The contents of chapter 3 are the result of a collaboration with Dr. Igor Mezić. A version of this paper is under review at SIAM Journal on Applied Dynamical Systems.

Chapter 3

Numerical Methods for Stochastic SINDy

3.1 Introduction

In the previous various dynamic mode decompositions algorithms were discussed. These algorithms provide a powerful tool for estimating system parameters from data. However, DMD might not generate a feasible model for some systems. For example, the system may not have a finite dimensional invariant subspace, the linear model generated by DMD may be too large, or the regression necessary for DMD may be too poorly conditioned.

There is a wide variety of methods that can be used for system identification, ranging from classical methods, [40] to the DMD methods discussed in the previous section to neural networks [36, 35] and many others. These methods vary in their their complexity, training methods, model sizes, and interpretability. DMD methods are advantageous for their simplicity; the models are strictly linear. Sparse Identification of Nonlinear Dynamics (SINDy) is a method which allows for some complexity (allowing nonlinear models over the purely linear ones generated by DMD) while the sparse solution promotes simple, interpretable models.

The SINDy algorithm, developed by Brunton et. al. [9] estimates the parameters of an ordinary differential equation from data. It does this by using a dictionary of functions and finding a sparse representation of the derivative in this dictionary. The data for the derivative can be obtained using finite differences of data from the state. For ODEs, the performance of this algorithm has been analyzed in [76].

SINDy has several extensions and adaptations; it has also been extended to identify control systems [10, 27], adapted to systems with implicit solutions [41, 26], and formulated in ways to improve its robustness to noise [19, 46, 45], to name a few. Additionally, different methods for computing the sparse solution have been proposed, including LASSO [72], the sequential thresholding presented in the original paper [9].

SINDy has also been extended to estimate the parameters of stochastic differential equations. In [6], it was demonstrated that we can use the SINDy algorithm to estimate both the drift and diffusion functions in an SDE. The drift and diffusion are estimated from the data of the state using the Kramer-Moyal formulas. This method was expanded on in [16]; solution methods based on binning and cross validation were introduced to reduce the effects of noise. Callaham et. al [11] expand upon this method by adapting it to applications for which the random forcing cannot be considered white noise.

In the section, we conduct a numerical analysis for using SINDy for stochastic system and introduce improved methods which give higher order convergence. As mentioned, in [6] the drift and diffusion are approximated using the Kramer-Moyal formulas. We demonstrate the convergence rates of the algorithm with respect to the sampling period and the length of the trajectory. The approximations given in [6] only give first order convergence with respect to the sampling frequency. A similar analysis of the Kramer-Moyal estimates based on binning can be found in [13]. Additionally, since they only converge in expectation, we may require a long trajectory for the variance of the estimate to be tolerable. Combined, these can make the data requirements to use SINDy for an SDE very demanding. To help remedy this, we demonstrate how we can develop higher order approximations of the drift and diffusion functions for use in SINDy.

This section is organized as follows: First, we will review the SINDy algorithm and some concepts from SDEs which we will be using in this paper. We will then conduct a numerical analysis of the algorithms presented in [6], including bounds on the error of the estimates. Next, we will present new, higher order methods and show the convergence rates of these methods. Finally, we will test all of these methods on several numerical examples to demonstrate how the new methods allow us to compute far more accurate approximations of the system for a given sampling frequency and trajectory length.

3.2 Sparse Identification of Nonlinear Dynamics (SINDy)

In the previous section, our dynamic mode ecomposition algorithms were defined for systems in discrete time. For this section, we will be considering the SINDy algorithm for continuous time systems, namely ordinary differential equations and stochastic differential equations. Specifically, we will be using SINDy to find approximations of the drift and diffusion functions of an SDE.

3.2.1 Overview of SINDy

Consider a system governed by the ordinary differential equation

$$\dot{x} = f(x), \qquad x \in \mathbb{R}^d. \tag{3.1}$$

If the dynamics of the system, f, are unknown, we would like to be able to estimate the function f using only data from the system. The SINDy algorithm [9] estimates f by choosing a dictionary of functions, $\theta = [\theta_1, \theta_2, ..., \theta_k]$ and assuming f can be expressed (or approximated) as a linear combination of these functions. The i^{th} component of f, f_i, can then be expressed as

$$f_i(x) = \sum_{j=1}^{k} \theta_j(x)\alpha_{i,j} = \theta(x)\alpha_i,$$

where $\theta = \begin{bmatrix} \theta_1 & \dots & \theta_k \end{bmatrix}$ is a row vector containing the dictionary functions and $\alpha^i = \begin{bmatrix} \alpha_1^i & \dots & \alpha_k^i \end{bmatrix}^T$ is the column vector of coefficients. Given data for $f(x_j)$ and $\theta(x_j)$ for

$j = 1, ..., n$, we can find the coefficients α_i by solving the minimization

$$\alpha_i = \underset{v}{argmin} \sum_{j=1}^{n} |f_i(x_j) - \theta(x_j)v|^2. \tag{3.2}$$

This optimization can be solved by letting

$$\Theta = \begin{bmatrix} \theta(x_1) \\ \theta(x_2) \\ \vdots \\ \theta(x_n) \end{bmatrix}, \quad F = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix}, \quad \text{and} \quad \alpha = \begin{bmatrix} \alpha^1 & \alpha^2 & \dots & \alpha^d \end{bmatrix},$$

and computing $\alpha = \Theta^+ F$.

3.2.2 Approximating $f(x)$

Typically, data for $f(x)$ cannot be measured directly. Instead, it is usually approximated using finite differences. The forward difference gives us a simple, first order approximation to f:

$$f(x(t)) = \frac{x(t + \Delta t) - x(t)}{\Delta t} + O(\Delta t). \tag{3.3}$$

The approximation (3.3) is derived from the Taylor expansion of x,

$$x(t+\Delta t) = x(t)+\dot{x}(t)\Delta t+\ddot{x}(t)\frac{\Delta t^2}{2}+... = x(t)+f(x(t))\Delta t+\frac{\partial f}{\partial x}\Big|_{x(t)} f(x(t))\frac{\Delta t^2}{2}+..., \tag{3.4}$$

for f sufficiently smooth. The Taylor expansion (3.4) is also used to derive higher order methods, such as the central difference,

$$f(x) = \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t} + O(\Delta t^2). \tag{3.5}$$

We can use these finite difference to populate the matrix F used in the optimization (3.2), knowing that we can control the error with a small enough step size.

3.2.3 Sparse Solutions

Since we are choosing an arbitrary dictionary of functions, $\{\theta_1, \ldots, \theta_k\}$, the conditioning of the minimization (3.2) can become very poor. Additionally, if the the dictionary is large and contains many redundant functions, having a solution which contains only a few nonzero entries would help to provide a simple interpretable result. The SINDy algorithm addresses these by using a sparse solution to (3.2). There are multiple methods for obtaining a sparse solution such as the least absolute shrinkage and selection operator (LASSO) or the sequentially thresholded least squares algorithm [9]. Using a sparse solution will give us a simpler identified system and improves the performance over the least squares solution.

3.3 Review of SDEs

Consider the Ito stochastic differential equation

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t \tag{3.6}$$

where $X_t \in \mathbb{R}^d$ and W_t is d-dimensional Brownian motion. The function $\mu : \mathbb{R}^d \to \mathbb{R}^d$ is the drift, a vector field which determines the average motion of system, while $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is the diffusion function, which governs the stochastic forcing. The diffusion, σ, is also assumed to be positive definite. Motivated by SINDy, we wish to estimate μ and σ^2 from data. We note that we are estimating $\Sigma = \frac{1}{2}\sigma^2$ and not σ directly. However, if σ is positive definite, which is assumed, σ^2 uniquely determines σ.

3.3.1 Ergodicity

As with the DMD algorithms in the previous section, SINDy represents functions using the data vectors evaluated along the trajectory. In order to relate the the data vectors to the functions, we will assume that the process X_t has an ergodic measure ρ (we note that in this section we are using ρ as the ergodic measure, since μ represents the drift). For this system, we will assume that both (1.4) and its continuous time analogue hold.

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T f(X_t)dt = \int_{\mathbb{R}^d} f(x)d\rho(x) \quad \text{and} \quad \lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1} f(X_{t_i}) = \int_{\mathbb{R}^d} f(x)d\rho(x) \tag{3.7}$$

hold almost surely. Some sufficient conditions that ensure that the SDE (3.6) generates a process with a stationary or an ergodic measure are given in e.g. [30].

With this ergodic measure, the natural function space to consider is the Hilbert space $L^2(\rho)$. For any two functions $f, g \in L^2(\rho)$, we can use time averages to evaluate inner products.

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T g^*(X_t)f(X_t)dt = \lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1} g^*(X_{t_n})f(X_{t_n}) = \int_{\mathbb{R}^d} g^* f\, d\rho = \langle f, g\rangle. \tag{3.8}$$

For notational simplicity, we will also use the brackets $\langle\cdot,\cdot\rangle$ to denote the matrix of inner products for two row vector-valued functions: if $f = \begin{bmatrix} f_1 & \dots & f_k \end{bmatrix}$ and $g = \begin{bmatrix} g_1 & \dots & g_l \end{bmatrix}$,

$$\langle f, g\rangle^{i,j} = \langle f^j, g^i\rangle, \qquad \text{or equivelently,} \qquad \langle f, g\rangle = \int_{\mathbb{R}^d} g^* f\, d\rho.$$

3.3.2 Ito-Taylor Expansion

In order to evaluate the performance of different SINDy methods on SDEs, we will need to use the Ito-Taylor expansion of the solution. Let $\Sigma = \frac{1}{2}\sigma^2$. Following the notation of [32], let

$$L^0 = \sum_{j=1}^{d} \mu^j \frac{\partial}{\partial x^j} + \sum_{j,l}^{d} (\Sigma)^{j,l} \frac{\partial^2}{\partial x^j \partial x^l}$$

be the operator for the Ito equation (3.6) and define the operators

$$L^j = \sum_{i=1}^{d} \sigma^{i,j} \frac{\partial}{\partial x^i}.$$

These operators will give us the coefficients for the Ito-Taylor expansion of a function f. Denoting $\Delta W_t^i = W_{t+\Delta t}^i - W_t^i$, the first couple of terms are

$$\begin{aligned} f(X_{t+\Delta t}) =& f(X_t) + L^0 f(X_t)\Delta t + \sum_{i=1}^{d} L^i f(X_t) \Delta W_t^i + (L^0)^2 f(X_t)\Delta t + \\ & \sum_{i=1}^{d} L^i L^0 f(X_t) \int_t^{t+\Delta t} \int_t^{s_1} dW_{s_2}^i ds_1 + \sum_{i=1}^{d} L^0 L^i f(X_t) \int_t^{t+\Delta t} \int_t^{s_1} ds_2 dW_{s_1}^i + \dots \end{aligned}$$

The general Ito-Taylor expansions can be found in Theorem 5.5.1 of [32]. We will use the Ito-Taylor expansion to develop estimates for μ^i and $\sigma^{i,j}$. For the purposes of this paper, we will be able to specialize to a few cases, which will allow us to quantify the error in our estimates while also being simpler to manipulate than the larger expansion.

Weak Expansion

The first specialization of the Ito-Taylor expansion will be a weak expansion, which will allow us to estimate the expected error in our estimate.

$$\mathbb{E}(f(X_{t+\Delta t})|X_t) = f(X_t) + \sum_{m=1}^{k} (L^0)^m f(X_t)\frac{\Delta t^m}{m!} + R(X_t). \tag{3.9}$$

with $R(X_t) = O(\Delta t^{m+1})$.

This expansion follows from the Proposition 5.5.1 and Lemma 5.7.1 of [32]. Theorem 5.5.1 gives the general Ito-Taylor expansion, while Lemma 5.7.1 shows that all multiple Ito integrals which contain integration with respect to a component of the Weiner process have zero first moment. The remainder term is then a standard integral.

We will consider the expansion (3.9) with the functions $f(x) = x^i$ to get

$$\mathbb{E}(X^i_{t+\Delta t}|X_t) = X^i_t + \mu^i(X_t)\Delta t + \sum_{m=2}^{k}(L^0)^{m-1}\mu^i(X_t)\frac{\Delta t^m}{m!} + O(\Delta t^{k+1}) \tag{3.10}$$

to estimate the drift. To estimate the diffusion, we will let $f(x) = (x^i - X^i_t)(x^j - X^j_t)$, with X_t held constant at the value at the beginning of the time step, to get

$$\mathbb{E}(f(X_{t+\Delta t})\,|\,X_t) = 2\Sigma^{i,j}(X_t)\Delta t + g(X_t)\Delta t^2 + O(\Delta t^3) \tag{3.11}$$

where

$$g = \left(L^0\Sigma^{i,j} + \mu^i\mu^j + \sum_{k=1}^{d}\Sigma^{i,k}\frac{\partial\mu^j}{\partial x^k} + \Sigma^{j,k}\frac{\partial\mu^i}{\partial x^k}\right).$$

Strong Expansions

We will also use the strong Ito-Taylor expansion, which will give a bound on the variance of our estimates. These immediately follow from Proposition 5.9.1 of [32]. First, if we apply it to $f(x) = x^i$, we have

$$X^i_{t+\Delta t} - X^i_t = \mu^i(X_t)\Delta t + \sum_{m=1}^{d} \sigma^{i,m}(X_t)\Delta W^m_t + R_t, \tag{3.12}$$

where $\mathbb{E}(|R_t|^2|X_t)d\rho = O(\Delta t^2)$.

Similarly, we can apply the same proposition to $f(x) = (x^i - X^i_t)(x^j - X^j_t)$, and which gives us (after moving around some of the terms)

$$(X^i_{t+\Delta t} - X^i_t)(X^j_{t+\Delta t} - X^j_t) = 2\Sigma^{i,j}(X_t)\Delta t + \sum_{k,l=1}^{d} (\sigma^{k,i}\sigma^{l,j}(X_t) + \sigma^{k,j}\sigma^{l,i}(X_t))I_{(i,j)} + R_t, \tag{3.13}$$

where $\mathbb{E}(|R_t|^2|X_t) = O(\Delta t^3)$ and $I_{(i,j)} = \int_0^{\Delta t}\int_0^{s_1} dW^i_{s_2}dW^j_{s_1}$. When we create estimates of $\mu^i(X_t)$ and $\Sigma^{i,j}(X_t)$, the expansions (3.12) and (3.13) will be useful in bounding the variance of these two estimates.

Remark 6 *For the expansions, it is implicit that we must assume that all (up to the necessary order) of the coefficient functions, $L^{a_1}L^{a_2}...L^{a_n}f$, satisfy the integrability requirements with respect to the multiple Ito integrals set forth in chapter five of [32]. Additionally, we will also assume that the remainder terms will be square integrable with respect to the ergodic measure. In particular, we will assume*

$$\int_{\mathbb{R}^d} |R(x)|^2 d\rho(x) = O(\Delta t^{m+1})$$

in the weak expansion and

$$\int_{\mathbb{R}^d} R_2(x)^2 d\rho(x) = O(\Delta t) \qquad \big(or \quad O(\Delta t^2)\big)$$

in the strong expansions, where $R_2(x) = \mathbb{E}(|R_t|^2 \mid X_t = x)$. This assumption will allow *us to take time averages and expect them to be finite. Following the proofs in [32], it can be seen that these can be guaranteed imposing similar conditions on the coefficient functions.*

3.4 SINDy for Stochastic Systems

Given data for the drift and diffusion matrix of (3.6), we can set up an optimization problem similar to (3.2). Similar to the deterministic case, we can also approximate μ and Σ using finite differences. As before, we assume we have a dictionary $\theta = [\theta_1, \theta_2, ..., \theta_k]$ and that each of the components of μ and Σ lie in the span of the components of θ:

$$\mu^i = \theta\alpha^i \qquad \text{and} \qquad \Sigma^{i,j} = \theta\beta^{i,j}.$$

Suppose we have the data from a trajectory of length T with sampling period Δt. If we let $\Delta X^i_{t_n} = X^i_{t_{n+1}} - X^i_{t_n}$, we can approximate the drift using

$$\mu^i(X_{t_m}) \approx \frac{X^i_{t_{m+1}} - X^i_{t_m}}{\Delta t} = \frac{\Delta X^i_t}{\Delta t}. \tag{3.14}$$

Similarly, we can approximate the diffusion with

$$\Sigma^{i,j}(X_{t_m}) \approx \frac{(X^i_{t_{m+1}} - X^i_{t_m})(X^j_{t_{m+1}} - X^j_{t_m})}{2\Delta t} = \frac{\Delta X^i_{t_m}\Delta X^j_{t_m}}{2\Delta t}. \tag{3.15}$$

It was shown in [6] that we can use the approximations (3.14) and (3.15) to set up the minimization problems

$$\tilde{\alpha}^i = \underset{v}{argmin} \sum_{m=0}^{N-1} \left| \frac{\Delta X^i_{t_m}}{\Delta t} - \theta(X_{t_m})v \right|^2. \tag{3.16}$$

and

$$\tilde{\beta}^{i,j} = \underset{v}{argmin} \sum_{m=0}^{N-1} \left| \frac{\Delta X^i_{t_m} \Delta X^j_{t_m}}{2\Delta t} - \theta(X_{t_m})v \right|^2 . \tag{3.17}$$

Under the assumptions set forth in Remark 6, we can show that as $\Delta t \to 0$ and $T \to \infty$, the coefficients given by (3.16) and (3.17) converge to the true coefficients; $\tilde{\alpha}^i \to \alpha^i$ and $\tilde{\beta}^{i,j} \to \beta^{i,j}$.

If we define the matrices

$$\Theta = \begin{bmatrix} \theta(X_{t_0}) \\ \theta(X_{t_1}) \\ \vdots \\ \theta(X_{t_{N-1}}) \end{bmatrix}, \qquad \text{and} \qquad D^i = \begin{bmatrix} \Delta X^i_{t_0} \\ \Delta X^i_{t_1} \\ \vdots \\ \Delta X^i_{t_{N-1}} \end{bmatrix}, \tag{3.18}$$

We can express (3.16) and (3.17) concisely as

$$\tilde{\alpha}^i = \underset{v}{argmin} \left\| \frac{D^i}{\Delta t} - \Theta v \right\| \qquad \text{and} \qquad \beta^{i,j} = \underset{v}{argmin} \left\| \frac{D^i \odot D^j}{2\Delta t} - \Theta v \right\| .$$

(Here $D^i \odot D^j$ represents the Hadamard, or element-wise, product.) These equations are solved by $\tilde{\alpha}_i = \Delta t^{-1} \Theta^+ D^i$ and $\tilde{\beta}_{i,j} = (2\Delta t)^{-1} \Theta^+ (D^i \odot D^j)$, respectively.

Theorem 3 *Let X_t be an ergodic drift-diffusion process generated by the SDE (3.6). Consider the optimization problems (3.16) and (3.17) using data from a trajectory of length T sampled with frequency Δt. Suppose the components of θ are linearly independent and span the subspace $\mathcal{F}$, and that the assumptions on the Ito-Taylor expansions outlined in Remark 6 are met. If μ^i or $\Sigma^{i,j}$ lie in $\mathcal{F}$, then the vectors given by corresponding optimization converges in probability to the true coefficients as $T \to \infty$ and $\Delta t \to 0$. That is, $\tilde{\alpha}^i \to \alpha^i$ or $\tilde{\beta}^{i,j} \to \beta^{i,j}$.*

The formal proof of Theorem 3 will be subsumed into the stronger Theorems 4 and

5, which give rates for the convergence. However, to demonstrate the idea of the proof, by the assumptions we have Θ has full rank and $\mu = \theta\alpha^i$, $\Sigma^{i,j} = \theta\beta^{i,j}$.

$$\tilde{\alpha}^i = (\Theta^*\Theta)^{-1}\Theta^*\frac{D^i}{\Delta t} = \left(\frac{1}{N}\Theta^*\Theta\right)^{-1}\left(\frac{1}{N\Delta t}\Theta^* D^i\right),$$

where $N = T/\Delta t$ is the number of data samples. The first quantity can be evaluated using ergodicity, as $N \to \infty$

$$\frac{1}{N}\Theta^*\Theta = \frac{1}{N}\sum_{m=0}^{N-1}\theta^*(X_{t_m})\theta(X_{t_m}) \xrightarrow{N} \langle\theta,\theta\rangle.$$

For the second expression, the definition of the stochastic integral gives us

$$\Theta^* D^i = \sum_{m=0}^{N-1}\theta^*(X_m)(X^i_{t_{m+1}} - X^i_{t_m}) \xrightarrow{\Delta t} \int_{t_0}^{t_0+T}\theta^* dX^i$$

as $\Delta t \to 0$. Finally, using (3.6) and (3.8), we can show

$$\frac{1}{N\Delta t}\Theta^* D^i \xrightarrow{\Delta t} \frac{1}{T}\int_{t_0}^{t_0+T}\theta^* dX^i \xrightarrow{T} \langle\mu,\theta\rangle = \langle\theta,\theta\rangle\alpha^i \tag{3.19}$$

as $\Delta t \to 0$ and $T \to \infty$. The limit as $\Delta t \to 0$ gives the convergence of the sum to the stochastic integral and the limit as $T \to \infty$ allows us to sample almost everywhere on the stationary measure for the ergodic convergence. Similarly, we can use the convergence

$$\sum_{m=0}^{N-1}\theta^*(X_{t_m})(X^i_{t_{m+1}} - X^i_{t_m})(X^j_{t_{m+1}} - X^j_{t_m}) \xrightarrow{\Delta t} \int_{t_0}^{t_0+T}\theta^* d[X^i, X^j], \qquad \Delta t \to 0$$

to show that $\frac{1}{2N\Delta t}\Theta^*(D^i \odot D^j) \to \langle\Sigma^{i,j},\theta\rangle = \langle\theta,\theta\rangle\beta^{i,j}$. (Here $[X,Y]_t$ is the quadratic covariation process of X_t, and Y_t.) This would establish the result, except that we used the iterated limits $\Delta t \to 0$ and $T \to \infty$ in (3.19) without showing the double limit exists.

This is where we would use the integrability assumptions in Remark 6, which are used in the proofs of Theorems 4 and 5.

Theorem 3 demonstrates how the least squares solutions converge to the true coefficients of the SDE. However, the SINDy algorithm finds a sparse solution, which can greatly improve the accuracy of the results over the least squares solution. To set this up, the two optimizations (3.16) and (3.17) can be summarized using the normal equations,

$$\Theta^*\Theta\tilde{\alpha}^i = \frac{1}{\Delta t}\Theta^* D^i \tag{3.20}$$

and

$$\Theta^*\Theta\tilde{\beta}^{i,j} = \frac{1}{2\Delta t}\Theta^*(D^i \odot D^j). \tag{3.21}$$

We can then solve equations (3.20) and (3.21) using a sparse solver, such as the one proposed in [9] to obtain a sparse solution.

3.5 Numerical Analysis of Stochastic SINDy

Theorem 3 claims that as $\Delta t \to 0$ and $T \to \infty$, the coefficients given by the (3.16) and (3.17) converge to the true parameters of the SDE (3.6) as $\Delta t \to 0$ and $T \to \infty$. In this section, we will look at the accuracy and variation of the approximations for finite Δt and T. In this setting, we will be using "big 'O'" notation to denote convergence as $\Delta t \to 0$, and we will be using "little 'o'" notation for the convergence as $T \to \infty$.

The SINDy algorithm will give us vectors of coefficients, $\tilde{\alpha}^i$ and $\tilde{\beta}^{i,j}$, for the system. We will be interested in the error of these vectors relative to the true coefficients α^i and $\beta^{i,j}$,

$$err = \tilde{\alpha}^i - \alpha^i \quad \text{or} \quad err = \tilde{\beta}^{i,j} - \beta^{i,j}.$$

(We note that this error is specifically for the vector α^i or $\beta^{i,j}$ being estimated, even though it is not indexed. Since each vector is estimated separately, there should be no confusion.) This error will be a random variable depending on the realization of the system. To evaluate the performance of the algorithms, we will use the mean and variance of this error:

$$err_{mean} = \|\mathbb{E}(err)\|_2 \qquad \text{and} \qquad err_{var} = Var(err) = \mathbb{E}(\|err - \mathbb{E}(err)\|_2^2).$$

The mean error and variance measure the bias and spread in the estimates $\tilde{\alpha}^i$ and $\tilde{\beta}^{i,j}$. These errors in the coefficients can be quantified using the errors in the estimates of μ^i and $\Sigma^{i,j}$ given in (3.14) and (3.15) at each step. We will present the analysis for the drift coefficients, α^i, noting that analysis for the diffusion follows the same path.

3.5.1 Drift

As mentioned, the error in $\tilde{\alpha}^i$ stems from the error in the approximation in (3.14)

$$\mu^i(X_{t_n}) \approx \frac{X_{t_{n+1}} - X_{t_n}}{\Delta t}.$$

We can define the error

$$e_{t_n} = \frac{X^i_{t_{n+1}} - X^i_{t_n}}{\Delta t} - \mu^i(X_{t_n}).$$

The order of the error, e_t, at each time step will directly determine the error in the coefficients $\tilde{\alpha^i}$. We can use Ito-Taylor expansions for X_t to bound both $\mathbb{E}(|e_t|)$ and $\mathbb{E}(|e_t|^2)$. The weak Ito-Taylor expansion (3.9) gives us

$$\mathbb{E}(e_t \,|\, X_t) = \frac{1}{\Delta t}\left(\mu^i(X_t)\Delta t + L^0\mu^i(X_t)\frac{\Delta t^2}{2} + O(\Delta t^3)\right) - \mu^i(X_t) = L^0\mu^i(X_t)\frac{\Delta t}{2} + O(\Delta t^2). \tag{3.22}$$

Similarly, we can use the strong truncation (3.12) to obtain

$$e_t = \sum_{m=1}^{d} \sigma^{i,m}(X_t) \frac{\Delta W_t^m}{\Delta t} + \frac{R_t}{\Delta t},$$

where $\mathbb{E}(|R_t|^2 | X_t) = O(\Delta t^2)$. Then, taking the expectance of e_t^2, we get

$$\mathbb{E}(|e_t|^2 \mid X_t) = \sum_{m=1}^{d} \frac{\sigma^{i,m}(X_t)^2}{\Delta t} + O\left(\Delta t^{\frac{-1}{2}}\right). \tag{3.23}$$

Now, let E be the matrix containing the time samples of e_t,

$$E = \begin{bmatrix} e_{t_0} & e_{t_1} & \dots & e_{t_{N-1}} \end{bmatrix}^T = \frac{D^i}{\Delta t} - \Theta\alpha^i,$$

using $\theta(X_t)\alpha^i = \mu^i(X_t)$. Then we have

$$err = \tilde{\alpha}^i - \alpha^i = \Theta^+ \frac{D^i}{\Delta t} - \Theta^+\Theta\alpha = (\Theta^*\Theta)^{-1}\Theta^* E. \tag{3.24}$$

Using ergodicity, we have

$$\left(\frac{1}{N}\Theta^*\Theta\right)^{-1} = (\langle\theta,\theta\rangle + o(1))^{-1} = \langle\theta,\theta\rangle^{-1} + o(1), \tag{3.25}$$

which allows us to evaluate the first term in (3.24):

$$err = (\langle\theta,\theta\rangle^{-1} + o(1)) \left(\frac{1}{N}\Theta^* E\right). \tag{3.26}$$

Bounding the mean and variance will follow from bounds on the mean and variance of $\frac{1}{N}\Theta^* E$.

Theorem 4 *Consider the optimization problem given by (3.14) and (3.16). Then the* *bias is bounded by*

$$err_{mean} \leq \frac{C_1}{2}\left(\|L^0\mu^i\|_2 + O(\Delta t) + o(1)\right)\Delta t$$

and

$$err_{var} \leq \frac{C_2}{T}\left(\sum_{m=1}^{d}\|\sigma^{i,m}\|_4^2 + O\left(\Delta t^{\frac{1}{2}}\right) + o(1)\right),$$

where

$$C_1 = \|\langle\theta,\theta\rangle^{-1}\|_2\|\theta\|_2 \quad \textit{and} \quad C_2 = \|\langle\theta,\theta\rangle^{-1}\|_2^2\|\theta\|_4^2 \tag{3.27}$$

depend only on the choice of θ*.*

Proof: For the mean error, we will need to bound the quantity $\frac{1}{N}\left\|\mathbb{E}\left(\Theta^* E\right)\right\|$. We have

$$\mathbb{E}\left(\frac{1}{N}\Theta^* E\right) = \mathbb{E}\left(\frac{1}{N}\sum_{n=0}^{N-1}\theta^*(X_{t_n})e_{t_n}\right) = \mathbb{E}\left(\frac{1}{N}\sum_{n=0}^{N-1}\theta^*(X_{t_n})\mathbb{E}(e_{t_n}\mid X_{t_n})\right).$$

Then, using ergodicity and (3.22), we obtain

$$\begin{aligned}\mathbb{E}\left(\frac{1}{N}\Theta^* E\right) &= \mathbb{E}\left(\frac{1}{N}\sum_{n=0}^{N-1}\theta^*(X_{t_n})\left(\frac{\Delta t}{2}L^0\mu^i(X_{t_n}) + O(\Delta t^2)\right)\right)\\ &= \frac{\Delta t}{2}\left(\langle L^0\mu^i,\theta\rangle + o(1)\right) + O(\Delta t^2).\end{aligned}$$

Finally, using (3.26), we get

$$\begin{aligned}\|\mathbb{E}(err)\| &= \left\|\left(\langle\theta,\theta\rangle^{-1} + o(1)\right)\right\|_2\left(\frac{\Delta t}{2}\left(\langle L^0\mu^i,\theta\rangle + o(1)\right) + O(\Delta t^2)\right)\\ &\leq \|\langle\theta,\theta\rangle^{-1}\|_2\left(\|\theta\|_2\|L^0\mu^i\|_2 + O(\Delta t) + o(1)\right)\frac{\Delta t}{2} = C_1\left(\|L^0\mu^i\|_2 + O(\Delta t) + o(1)\right)\frac{\Delta t}{2}\end{aligned}$$

This bounds the mean error. To find the variance, we have

$$
\begin{aligned}
Var\left(\frac{1}{N}\Theta^* E\right) &\leq \mathbb{E}\left(\left\|\frac{1}{N}\Theta^* E\right\|_2^2\right) = \mathbb{E}\left(\left\|\sum_{n=0}^{N-1}\theta^*(X_{t_n})e_{t_n}\right\|_2^2\right) \leq \mathbb{E}\left(\sum_{n=0}^{N-1}\|\theta^*(X_{t_n})\|_2^2|e_{t_n}|^2\|\right)\\
&= \mathbb{E}\left(\sum_{n=0}^{N_1}\|\theta(X_{t_n})\|_2^2\,\mathbb{E}\left(|e_{t_n}|^2 \mid X_{t_n}\right)\right)
\end{aligned}
$$

Now, using (3.23) with this equation, we have

$$
\begin{aligned}
Var\left(\frac{1}{N}\Theta^* E\right) &\leq \mathbb{E}\left(\frac{1}{N^2}\sum_{n=0}^{N-1}\|\theta(X_{t_n})\|_2^2\left(\sum_{m=1}^{d}\frac{|\sigma^{i,m}|^2}{\Delta t} + O\left(\Delta t^{\frac{-1}{2}}\right)\right)\right)\\
&= \frac{1}{N\Delta t}\left(\sum_{m=1}^{d}\langle(\sigma^{i,m})^2, \|\theta\|_2^2\rangle + O\left(\Delta t^{\frac{1}{2}}\right) + o(1)\right)\\
&\leq \frac{1}{T}\|\theta\|_4^2\left(\sum_{m=1}^{d}\|\sigma^{i,m}\|_4^2 + O\left(\Delta t^{\frac{1}{2}}\right) + o(1)\right).
\end{aligned}
$$

Then

$$
\begin{aligned}
Var(err) &= \left(\|\langle\theta,\theta\rangle^{-1}\|_2^2 + o(1)\right)\|\theta\|_4^2\left(\frac{1}{T}\left(\sum_{m=1}^{d}\|\sigma^{i,m}\|_4^2 + O\left(\Delta t^{\frac{1}{2}}\right) + o(1)\right)\right)\\
&= \frac{\|\langle\theta,\theta\rangle^{-1}\|_2^2\|\theta\|_4^2}{T}\left(\sum_{m=1}^{d}\|\sigma^{i,m}\|_4^2 + O\left(\Delta t^{\frac{1}{2}}\right) + o(1)\right)\\
&= \frac{C_2}{T}\left(\sum_{m=1}^{d}\|\sigma^{i,m}\|_4^2 + O\left(\Delta t^{\frac{1}{2}}\right) + o(1)\right)
\end{aligned}
$$

■

As shown in Theorem 4, in expectation, the accuracy of our estimate depends primarily on the sampling period Δt, and not on the length of the trajectory. The length of the trajectory instead controls the variance of the estimate, which is proportional to $1/T$. Up to the leading term, the variance does not depend on the sampling period. This pattern will persist as we develop higher order methods for estimating the drift, where the sampling frequency determines the bias and the length of the trajectory determines the variance.

3.5.2 Diffusion

The analysis of the diffusion coefficients follows the same argument. The approximation for $\Sigma^{i,j}$ given in (3.15) is

$$\Sigma^{i,j}(X_{t_m}) \approx \frac{(X^i_{t_{m+1}} - X^i_{t_m})(X^j_{t_{m+1}} - X^j_{t_m})}{2\Delta t} = \frac{\Delta X^i_{t_m} \Delta X^j_{t_m}}{2\Delta t}.$$

Then we can define the error

$$e_t = \frac{(X^i_{t+\Delta t} - X^i_t)(X^j_{t+\Delta t} - X^j_t)}{2\Delta t} - \Sigma^{i,j}(X_t).$$

We can use the weak Ito-Taylor expansion (3.11) to bound $\mathbb{E}(e_t \,|\, X_t)$:

$$\mathbb{E}(e_t \,|\, X_t) = g(X_t)\frac{\Delta t}{2} + O(\Delta t^2), \qquad g = \left(L^0\Sigma^{i,j} + \mu^i\mu^j + \sum_{k=1}^{d} \Sigma^{i,k}\frac{\partial \mu^j}{\partial x^k} + \Sigma^{j,k}\frac{\partial \mu^i}{\partial x^k} \right). \tag{3.28}$$

To calculate the squared error, $\mathbb{E}(|e_t|^2|X_t)$, we will use the strong expansion (3.13). This gives us

$$e_t = \frac{1}{2\Delta t}\left(\sum_{k,l=1}^{d} (\sigma^{k,i}\sigma^{l,j}(X_t) + \sigma^{k,j}\sigma^{l,i}(X_t))I_{i,j} + R_t \right) \tag{3.29}$$

with $\mathbb{E}(|R_t|^2|X_t) = O(\Delta t^3)$. From Lemma 5.7.2 of [32], we have

$$\mathbb{E}(I_{(k,l)}I_{(m,n)}) = \begin{cases} 0, & (k,l) \neq (m,n) \\ \frac{\Delta t^2}{2}, & k = m, l = n. \end{cases}$$

Then, squaring (3.29), we get

$$\begin{aligned}\mathbb{E}(|e_t|^2 \mid X_t) &= \frac{1}{4\Delta t^2}\sum_{k,l=1}^{d}(\sigma^{k,i}\sigma^{l,j}(X_t) + \sigma^{k,j}\sigma^{l,i}(X_t))^2\frac{\Delta t^2}{2} + O(\Delta t^{\frac{1}{2}}) \\ &= \frac{1}{8}\sum_{k,l=1}^{d} 2\left((\sigma^{k,i}\sigma^{l,j}(X_t))^2 + \sigma^{k,i}\sigma^{l,j}\sigma^{k,j}\sigma^{l,i}(X_t)\right) + O(\Delta t^{\frac{1}{2}}) \\ &= \Sigma^{i,i}(X_t)\Sigma^{j,j}(X_t) + \Sigma^{i,j}(X_t)^2 + O(\Delta t^{\frac{1}{2}}).\end{aligned}$$

Then we have

$$\mathbb{E}(|e_t|^2 \mid X_t) = \Sigma^{i,i}(X_t)\Sigma^{j,j}(X_t) + \Sigma^{i,j}(X_t)^2 + O(\Delta t^{\frac{1}{2}}). \tag{3.30}$$

Theorem 5 *Consider the optimization problem given by (3.15) and (3.17). Then the mean error is bounded by*

$$err_{mean} = \frac{C_1}{2}(\|g\| + O(\Delta t) + o(1))\Delta t,$$

where

$$g = \left(L^0\Sigma^{i,j} + \mu^i\mu^j + \sum_{k=1}^{d}\Sigma^{i,k}\frac{\partial\mu^j}{\partial x^k} + \Sigma^{j,k}\frac{\partial\mu^i}{\partial x^k}\right).$$

The variance is bounded by

$$err_{var} = \frac{C_2}{4}\left(\left\|\Sigma^{i,i}\Sigma^{j,j} + (\Sigma^{i,j})^2\right\| + O(\Delta t^{\frac{1}{2}}) + o(1)\right)\frac{\Delta t}{T}.$$

The constants C_1 and C_2 are the same as those given in (3.27).

Proof: The proof follows that of Theorem 4, except using equations (3.28) and (3.30) to bound $|\mathbb{E}(e_t \mid X_t)|$ and $\mathbb{E}\left(|e_t|^2 \mid X_t\right)$, respectively. ■

Similar to Theorem 4, the proof above shows that the mean error converges with order

Δt. However, unlike the estimate for the drift, when estimating the diffusion the variance is proportional to both Δt and $1/T$. This will also hold true for higher order estimates of the diffusion.

3.6 Higher Order Methods

From Theorems 4 and 5 we can see that the quantities Δt, T, C_1, and C_2 will control the magnitude of the error. The constants, C_1 and C_2, depend only on the choice of the dictionary θ, which determines the conditioning of the problem. The SINDy algorithm also uses a sparsity promoting algorithms which can improve the conditioning of the problem and force many of the coefficients to zero, which can reduce the error [9],[6]. However, even if the sparsity promoting algorithm chooses all of the correct coefficients, we have just shown that there is still a limit to the accuracy of the estimation determined by the sampling frequency and trajectory. The primary purpose of this section is to analyze alternate methods of approximating μ^i and $\Sigma^{i,j}$ which can improve the performance of SINDy (with respect to Δt).

The methods above resulted from first order approximations (3.14) and (3.15) of $\mu^i(X_t)$ and $\Sigma^{i,j}(X_t)$, respectively. Higher order approximations of these data points can in turn lead more accurate approximations of the functions in the output of SINDy. We can generate better approximations for the drift using multistep difference method. The use of linear multistep methods (LMMs) to estimate dynamics is investigated in [29] for deterministic systems. While the estimates for the diffusion will be similar, they can not be achieved strictly using LMMs.

In order to achieve a higher order approximation, we will need to use more data points in the approximation at each time step. As such, we will define

$$\Theta_n = \begin{bmatrix} \theta(X_{t_n}) \\ \theta(X_{t_{n+1}}) \\ \vdots \\ \theta(X_{t_{N+n-1}}) \end{bmatrix} \quad \text{and} \quad D_n^i = \begin{bmatrix} X_{t_n}^i - X_{t_0}^i \\ X_{t_{n+1}}^i - X_{t_1}^i \\ \vdots \\ X_{t_{N+n-1}}^i - X_{t_{N-1}}^i \end{bmatrix}. \tag{3.31}$$

With this definition, Θ_n contains the data of θ time delayed by n steps. With the earlier definition of Θ, we have $\Theta = \Theta_0$. Similarly, D_n^i contains the data for the change in X over n time steps, with $D_1^i = D^i$ using the earlier definition of D^i.

3.6.1 Drift

First, we will look to make improvements on estimating the drift. These estimates will be simpler than those for the diffusion. As mentioned, these approximations are directly analogous to the linear multistep methods used in the simulation of deterministic systems.

Second Order Forward difference

The first order forward difference, which is used to approximate μ^i in Theorem 4, is also commonly used to approximate the derivative $f(x)$ in the differential equation $\dot{x} = f(x)$. In fact, if we compare the weak Ito-Taylor expansion (3.9) with the deterministic Taylor series for an ODE, (3.4), we see that they are almost identical. There are many higher order methods which are used to approximate f in the simulation of ODEs. By analogy, can expect that these methods would give an approximation of the same order for μ^i (in expectation). One of the simplest of these would be the second order forward difference,

$$\mu^i(X_{t_n}) \approx \frac{4(X_{t_{n+1}} - X_t) - (X_{t_{n+2}} - X_t)}{2\Delta t} = \frac{-3X^i_{t_n} + 4X^i_{t_{n+1}} - X^i_{t_{n+2}}}{2\Delta t}. \tag{3.32}$$

Similar to before we can define the error in this approximation to be

$$e_t = \frac{-3X^i_t + 4X^i_{t+\Delta t} - X^i_{t+2\Delta t}}{2\Delta t} - \mu^i(X_t).$$

Using the weak Ito-Taylor expansion (3.9), it is easy to see that

$$\mathbb{E}(e_{t_n} \mid X_{t_n}) = -\frac{(L^0)^2\mu^i(X_{t_n})}{3}\Delta t^2 + O(\Delta t^3), \tag{3.33}$$

which shows that this method does indeed give a second order approximation of μ. Using this approximation, we can set up a matrix formulation of (3.32):

$$\Theta_0\alpha^i \approx \frac{1}{2\Delta t}\left(4D^i_1 - D^i_2\right),$$

If we set up the normal equations, this becomes

$$\Theta_0^*\Theta_0\tilde{\alpha}^i = \frac{1}{2\Delta t}\Theta_0^*\left(4D^i_1 - D^i_2\right). \tag{3.34}$$

Theorem 6 *Consider the approximation $\tilde{\alpha}^i$ obtained from (3.34). The mean error is bounded by*

$$\|\mathbb{E}(err)\|_2 = \frac{C_1}{3}(\|(L^0)^2\mu^i\| + O(\Delta t) + o(1))\Delta t^2$$

and the mean squared error by

$$\mathbb{E}\left(\|(err)\|_2^2\right) = \frac{C_2}{T}\left(\sum_j^d \|\sigma^{i,j}\|_4^2 + O(\Delta t^{\frac{1}{2}}) + o(1)\right).$$

The constants C_1 and C_2 are the same as those given in (3.27).

The proof of Theorem 6 is similar to that of Theorem 4, but requires some extra algebraic manipulation to bound the mean squared error.

Proof: The proof of the estimate on the mean error follows from (3.33) and the proof of Theorem 4. Now, to let Θ_0 be defined as in (3.31) and

$$E = \begin{bmatrix} e_0 & e_1 & \dots & e_{N-1}\end{bmatrix}^T.$$

To estimate the variance, we need to find $\mathbb{E}(\|\frac{1}{N}\Theta_0^* E\|_2^2)$. To do this, we will use the strong expansion (3.12) and obtain

$$e_t = \frac{1}{2\Delta t}\left(\sum_{m=1}^d \sigma^{i,m}(3\Delta W_t^m - \Delta W_{t+1}^m) + R_t\right)$$

with $\mathbb{E}(|R_t|^2) = O(\Delta t^2)$. Then, using the

$$\begin{aligned}\frac{1}{N}\Theta_0^* E = \frac{1}{N}\sum_{n=0}^{N-1}\theta^*(X_{t_n})e_{t_n} &= \frac{1}{N}\sum_{n=0}^{N-1}\theta^*(X_{t_n})\left(\sum_{m=1}^d \sigma^{i,m}(X_{t_n})\frac{3\Delta W_{t_n}^m - \Delta W_{t_{n+1}}^m}{2\Delta t} + \frac{R_{t_n}}{2\Delta t}\right)\\ &= \frac{1}{2T}\sum_{n=0}^{N-1}\theta^*(X_{t_n})\left(\sum_{m=1}^d (3\sigma^{i,m}(X_{t_n}) - \sigma^{i,m}(X_{t_{n-1}}))\Delta W_{t_n}^m + R_{t_n}\right) + R_1\\ &= \frac{1}{T}\sum_{n=0}^{N-1}\theta^*(X_{t_n})\left(\sum_{m=1}^d (\sigma^{i,m}(X_{t_n}) + R_{t_n}^m)\Delta W_{t_n}^m + R_{t_n}\right) + R_1\end{aligned}$$

where

$$R_1 = \frac{1}{T}\sum_{m=1}^d \left(\theta^*(X_{t_0})\sigma^{i,m}(X_{t_0})\Delta W_{t_0}^m - \theta^*(X_{t_N})\sigma^{i,m}(X_{t_N})\Delta W_{t_N}^m\right)$$

and $\mathbb{E}(|R^m_{t_n}|^2) = O(\Delta t)$. The second line comes from rearranging the indices of the sum which gives the remainder R_1 and the last line uses the Ito-Taylor expansion of $\sigma^{i,m}$, which gives the remainder $R^m_{t_n}$. Combining all of the errors gives us

$$\frac{1}{N}\Theta_0^* E = \frac{1}{T}\sum_{n=0}^{N-1}\sum_{m=1}^{d} \theta^*(X_{t_n})\sigma^{i,m}(X_{t_n})\Delta W^m_{t_n} + R$$

with $\mathbb{E}(R^2) = O(\Delta t^2)$. Taking the expectance of the square of this last equation gives us

$$\mathbb{E}\left(\left\|\frac{1}{N}\Theta_0^* E\right\|_2^2\right) \leq \frac{1}{T^2}\sum_{n=0}^{N}\sum_{m=1}^{d} \|\theta^*(X_{t_n})\|_2^2 \sigma^{i,m}(X_{t_n})^2\Delta t + O(\Delta t^{\frac{3}{2}}).$$

Using this, the rest of the proof follows that of Theorem 4. ■

Remark 7 *These methods can easily be generalized to higher order methods using higher order finite differences, as will be done in section 3.6.1. However, the least squares solution only yields correct results for forward differences. Other finite difference methods can cause certain sums to converge to the wrong stochastic integral. For example, a central difference approximation for μ^i,*

$$\mu^i_t \approx \frac{X^i_{t+\Delta t} - X^i_{t-\Delta t}}{2\Delta t},$$

gives us $\Theta_1\alpha^i \approx \frac{1}{2\Delta t}D^i_2$. The normal equations for the least squares solution

$$\Theta_1^*\Theta_1\tilde{\alpha}^i = \frac{1}{2\Delta t}\Theta_1^* D^i_2 \tag{3.35}$$

gives the wrong results, because as $\Delta t \to 0$, $\frac{1}{2}\Theta_1^* D^i_2$ converges to the Stratonovich integral *instead of the Ito integral,*

$$\frac{1}{2}\Theta_1^* D_2^i \to \int_0^T \theta^*(X_t) \circ dX_t^i \neq \int_0^T \theta^*(X_t)\, dX_t^i,$$

and $\tilde{\alpha}^i$ will not converge to the correct value. To prevent this, (3.35) can instead be solved using

$$\Theta_0^* \Theta_1 \tilde{\alpha}^i = \frac{1}{2\Delta t} \Theta_0^* D_2^i,$$

which gives the proper convergence. This amounts to using Θ_0 as a set of instrumental variables for the regression.

Trapezoidal Method

The second order method above uses additional measurements of X_t^i to provide a more accurate estimate of μ^i. Alternatively, we can use multiple measurements of μ^i to better approximate the difference $X_{t+\Delta t}^i - X_t^i$. Consider the first order forward difference given by (3.14).

$$\mu^i(X_{t_n}) \approx \frac{X_{t_{n+1}}^i - X_{t_n}^i}{\Delta t}.$$

Theorem 4 used this difference to give an order Δt approximation of μ^i. However, it turns out that $\frac{1}{2}(\mu^i(X_t) + \mu^i(X_{t+\Delta t}))$ gives a much better approximation of this difference:

$$\frac{1}{2}\left(\mu^i(X_{t_n}) + \mu^i(X_{t_{n+1}})\right) \approx \frac{X_{t_{n+1}}^i - X_{t_n}^i}{\Delta t}. \tag{3.36}$$

We will call this approximation the trapezoidal approximation, since this is exactly the trapezoidal method used in the numerical simulation of ODEs. If we consider the error in this equation,

$$e_t = \frac{X_{t_{n+1}}^i - X_{t_n}^i}{\Delta t} - \frac{1}{2}\left(\mu^i(X_{t_n}) + \mu^i(X_{t_{n+1}})\right),$$

we can use the weak Ito-Taylor approximations of X_t and $\mu^i(X_t)$ to show that

$$\mathbb{E}(e_t \,|\, X_t) = -(L^0)^2\mu^i(X_t)\frac{\Delta t^2}{12} + O(\Delta t^3). \tag{3.37}$$

This not only gives us a second order method, with respect to Δt, but the leading coefficient for the error is much smaller (by a factor of 1/8) than the second order forward difference.

To set up the matrix formulation of (3.36), we have

$$\frac{1}{2}\left(\Theta_0 + \Theta_1\right)\alpha^i \approx \frac{1}{\Delta t}D_1^i. \tag{3.38}$$

We can multiply (3.38) by Θ_0^* on each side to obtain

$$\frac{1}{2}\Theta_0^*(\Theta_0 + \Theta_1)\tilde{\alpha}^i = \frac{1}{\Delta t}\Theta_0^* D_1^i. \tag{3.39}$$

We can use this equation analogously to the normal equation; we will solve for $\tilde{\alpha}^i$ either directly using matrix inversion or by using a sparse solver.

Remark 8 *We note that we cannot solve (3.38) using least squares,*

$$\tilde{\alpha}^i \neq \frac{2}{\Delta t}(\Theta_0 + \Theta_1)^+ D_1^i.$$

Similar to Remark 7, this leads to sums converging to the wrong stochastic integral.

Theorem 7 *Consider the estimation $\tilde{\alpha}^i$ given by solving (3.39). The mean error is bounded by*

$$err_{mean} \leq C_1\frac{\Delta t^2}{12}(\|(L^0)^2\mu^i\|_2 + O(\Delta t) + o(1))$$

and

$$err_{var} \leq \frac{C_2}{T}\left(\sum_{j=1}^{d} \|\sigma^{i,j}\|_2^2 + O(\Delta t^{\frac{1}{2}}) + o(1)\right).$$

Proof: Letting E be the matrix containing the samples of e_t. We have

$$\frac{1}{\Delta t} D_1^i = \frac{1}{2}(\Theta_0 + \Theta_1)\alpha^i + E.$$

Using this in (3.39) gives us

$$\frac{1}{2}\Theta_0^*(\Theta_0 + \Theta_1)\tilde{\alpha^i} = \frac{1}{2}\Theta_0^*(\Theta_0 + \Theta_1)\alpha^i + \Theta_0^* E,$$

so the error is

$$err = \tilde{\alpha}^i - \alpha^i = \left(\frac{1}{2}\Theta_0^*(\Theta_0 + \Theta_1)\right)^{-1}\Theta_0^* E.$$

Since $\mathbb{E}(\theta(X_{t+\Delta t})|X_t) = \theta(X_t) + O(\Delta t)$, we can use ergodicity to evaluate

$$\frac{1}{2N}\Theta_0^*(\Theta_0 + \Theta_1) \to \langle \theta, \theta \rangle + O(\Delta t) + o(1).$$

The proof of first inequality then follows the proof of Theorem 4 and (3.37). The second inequality also follows using

$$\mathbb{E}\left(\|e_t\|_2^2 \mid X_t = x\right) \leq \frac{1}{\Delta t}\sum_{m=1}^{d} |\sigma^{i,m}(x)|^2 + O(\Delta t^{\frac{-1}{2}}),$$

which can easily be derived using the Ito-Taylor expansions. ■

General Method for Estimating Drift

We have given methods which give second order estimates of α^i. To generate methods which give even higher order approximations, we note the similarities of the above methods to linear multi-step methods used in the numerical simulation of ODEs. Using the general LMM as a guide, we set up a general method for approximating μ^i:

$$\sum_{l=0}^{k} a_l \, \mu^i(X_{t_{n+l}}) \approx \sum_{l=1}^{p} b_l \, (X^i_{t_{n+l}} - X^i_{t_n}), \tag{3.40}$$

or

$$\left(\sum_{l=0}^{k} a_l \Theta_l \right) \alpha^i \approx \sum_{l=1}^{p} b_l D^i_l.$$

Keeping Remark 7 in mind, we can solve this using

$$\left(\sum_{l=0}^{k} a_l \Theta_0^* \Theta_l \right) \tilde{\alpha^i} = b_l \sum_{l=1}^{p} \Theta_0^* D^i_l. \tag{3.41}$$

The coefficients in (3.40) can be chosen to develop higher order methods. However, due to the stochastic nature of the problem, large amounts of data may be required to achieve the order in practice. We will need enough data to average over the randomness in the SDE, and the higher order methods can be sensitive to noise. More detailed investigation into the convergence of certain classes of methods for dynamics discovery can be found in [29] for deterministic systems.

3.6.2 Diffusion

In this section we will discuss improvements to the estimate for the diffusion. For some systems, particularly when the drift is large relative the diffusion, the first order approximation given above may not be sufficient to obtain an accurate estimate of the diffusion coefficient. Using similar ideas to the previous section we can use the Ito-Taylor expansions to develop more accurate estimates of $\Sigma^{i,j}(X_t)$. However, these methods will be more complex; in addition to samples of X_t, some of these methods may also require data from the drift, $\mu^i(X_t)$ and $\mu^j(X_t)$.

Drift Subtraction

Before discussing the higher order methods, we can make an improvement upon the first order method. By correcting for the effects of the drift in the first order method, we can make significant improvements to the constant controlling the error. The Ito-Taylor expansion for X_t gives us

$$X^i_{t+\Delta t} - X^i_t = \mu(X_t)\Delta t + \sum_{m=1}^{d} \sigma(X_t)\Delta W^m_t + R_t,$$

where $\Delta W_t = W_{t+\Delta t} - W_t$ is the increment of a d-dimensional Wiener process and R_t is the remainder term. This equation, with the remainder term excluded, actually gives the Euler-Marayama method for simulating SDEs. In essence, the approximation (3.15) uses

$$X^i_{t+\Delta t} - X^i_t \approx \sum_{m=1}^{d} \sigma^{i,m}(X_t)\Delta W^m_t$$

to approximate the increment of the Wiener process. However, (3.15) tosses out the $\mu(X_t)\Delta t$ term because it is of a higher order. If we include it, we get the more accurate

$$\sum_{m=1}^{d} \sigma^{i,m}\Delta W^m_t = (X^i_{t+\Delta t} - X^i_t) - \mu(X_t)\Delta t - R_t. \tag{3.42}$$

We can use this to generated a better approximation of $\Sigma^{i,j}$,

$$\Sigma^{i,j}(X_t) \approx \frac{(X^i_{t+\Delta t} - X^i_t - \mu^i(X_t)\Delta t)(X^j_{t+\Delta t} - X^j_t - \mu^j(X_t)\Delta t)}{2\Delta t}. \tag{3.43}$$

This approximation will be more accurate than (3.15), but it will have the same order with respect to Δt. Letting e_t be the error in (3.43), we can use the weak Ito-Taylor expansion to show

$$\mathbb{E}(e_t \,|\, X_t) = f(X_t)\frac{\Delta t}{4} + O(\Delta t^2), \qquad f = L^0\Sigma^{i,j} + \sum_{m=1}^{d}\left(\Sigma^{i,m}\frac{\partial \mu^i}{\partial x^m} + \Sigma^{j,m}\frac{\partial \mu^j}{\partial x^m}\right).$$

This gives an improvement over (3.28) by removing the $\mu^i\mu^j\Delta t$ term in f (compared to Theorem 5). While this may not be an increase in order, if the contribution of the drift dominates the diffusion in an SDE, this term will give the main contributions to the error. As we will see in the numerical experiments, this leads to a drastic improvement in accuracy for some problems.

In order to implement this method, we will need an approximation of μ^i. However, we can use the methods above to represent the drift as $\mu^i(X_t) \approx \theta(X_t)\tilde{\alpha}^i$. We can use this to set up the matrix equations

$$\Theta_0^*\Theta_0\tilde{\beta}^{i,j} = \frac{1}{\Delta t}(D_1^i - \Theta_0\tilde{\alpha}^i) \odot (D_1^j - \Theta_0\tilde{\alpha}^j), \tag{3.44}$$

and solve for $\tilde{\beta}^{i,j}$.

Remark 9 *The equation (3.44) assumes that the same dictionary θ is used to estimate μ^i, μ^j and $\Sigma^{i,j}$. In general, we could used separate dictionaries to estimate each of the parameters, since all we need are the approximations of the samples of $\mu^i(X_t)$ and $\mu^j(X_t)$ to estimate $\beta^{i,j}$.*

3.6.3 Second Order Forward Difference

While subtracting the drift from the differences $X_{t+\Delta t}^i - X_t^i$ gives marked improvements, we can also generate a higher order method using a two step forward difference, similar to the drift. The analysis for the estimation of the diffusion constant using the two step forward difference is essentially identical to that of the drift, so we will go through it briefly. Define the approximation

$$\Sigma^{i,j} \approx \frac{4(X^i_{t+\Delta t} - X^i_t)(X^j_{t+\Delta t} - X^j_t) - (X^i_{t+2\Delta t} - X^i_t)(X^j_{t+2\Delta t} - X^j_t)}{4\Delta t}. \tag{3.45}$$

As usual, letting e_t be the error in this approximation, we can use the Ito-Taylor expansions (3.9) and (3.13) to show that

$$\mathbb{E}(e_t) = O(\Delta t^2) \qquad \text{and} \qquad \mathbb{E}(|e_t|^2) = O(\Delta t).$$

This will gives us a second order method for the diffusion coefficients. We did not include the constants for the order Δt^2 for brevity, since the number of terms in the expressions can get quite large. We can use the approximation (3.45) to set up the matrix equations

$$\Theta_0^* \Theta_0 \tilde{\beta}^{i,j} = \frac{1}{4\Delta t} \Theta_0^* \left(4D_1^i \odot D_1^j - D_2^i \odot D_2^j \right), \tag{3.46}$$

which we can solve for $\tilde{\beta}^{i,j}$.

Theorem 8 *Consider the estimate $\tilde{\beta}^{i,j}$ given by solving (3.46). Then we have*

$$err_{mean} = O(\Delta t^2) + o(1)$$

and

$$err_{var} = \frac{1}{T} O(\Delta t) + o(1/T).$$

The proof of Theorem 8 is similar to the previous proofs. Additionally, we only give the leading order of the error, so deriving the bounds for $\mathbb{E}(e_t|X_t)$ and $\mathbb{E}(|e_t|^2|X_t)$ is simpler than the previous methods.

Trapezoidal Method

Extending the trapezoidal approximation to estimating the diffusion coefficient is slightly trickier. Let $\Delta X_t^i = X_{t+\Delta t}^i - X_t^i$. If we attempt use the analogue to (3.36), we get

$$\Sigma^{i,j}(X_{t_{n+1}}) + \Sigma^{i,j}(X-t) = \frac{\Delta X_{t_n}^i \Delta X_{t_n}^j}{\Delta t} + R_{t_n},$$

with

$$\mathbb{E}(R_{t_n}) = \frac{\Delta t}{2} f(X_t) + o(\Delta t^2), \qquad f = 2\mu^i \mu^j + \sum_{k=1}^{d} \left(\Sigma^{i,k} \frac{\partial \mu^i}{\partial x^k} + \Sigma^{j,k} \frac{\partial \mu^j}{\partial x^k} \right),$$

which is still only an order Δt method. However, we already demonstrated in (3.42) that correct the difference ΔX_t^i for the drift can improve our approximation of $\sum_{m=1}^d \sigma^{i,m} \Delta W_t^m$. We will use the same trick here, except we will improve upon (3.42) by using the average values of μ^i and μ^j instead of the value at the left endpoint:

$$\sum_{m=1}^{d} \sigma^{i,m} \Delta W_t^m \approx (X_{t+\Delta t} - X_t) - \frac{\Delta t}{2}(\mu(X_t) + \mu(X_{T+\Delta t})).$$

If we use these differences to generate the trapezoidal method, we get

$$\Sigma^{i,j}(X_{t+\Delta t}) - \Sigma^{i,j}(X_t) \approx \frac{\left(\Delta X_t^i - \frac{\Delta t}{2}(\mu^i(X_t) + \mu^i(X_{t+\Delta t}))\right)\left(\Delta X_t^j - \frac{\Delta t}{2}(\mu^j(X_t) + \mu^j(X_{t+\Delta t}))\right)}{2\Delta t}. \tag{3.47}$$

If we consider the error in (3.47), using the appropriate Ito-Taylor expansions we can show

$$|\mathbb{E}(e_t \,|\, X_t)| = O(\Delta t^2) \qquad \text{and} \qquad \mathbb{E}(|e_t|^2) = O(\Delta t).$$

Then, using the usual matrix notation, we can set up the equation

$$\Theta_0^*(\Theta_0+\Theta_1)\tilde{\beta}^{i,j} = \frac{1}{\Delta t}\left(D_1^i - \frac{\Delta}{2}t(\Theta_0+\Theta_1)\alpha^i\right) \odot \left(D_1^j - \frac{\Delta t}{2}(\Theta_0+\Theta_1)\alpha^j\right). \quad (3.48)$$

We can solve this equation to get an order Δt^2 approximation of $\beta^{i,j}$.

Theorem 9 *Consider the estimate $\tilde{\beta}^{i,j}$ given by solving (3.48). Then we have*

$$err_{mean} = O(\Delta t^2) + o(1)$$

and

$$err_{var} = \frac{1}{T}O(\Delta t) + o(1/T).$$

The proof of Theorem 9 is similar to the previous proofs, using the appropriate error bounds. Although the order of the error is identical to that of Theorem (8), we will see that this method tends to have lower error. We did not include the constant terms for these errors for brevity, since the higher order Ito-Taylor expansions involve many terms.

	Drift		Diffusion	
Name	Equation	Leading Error Term	Equation	Error
FD-Ord 1	(3.20)	$\frac{C_1}{2}\|L^0\mu^i\|_2\Delta t$	(3.21)	$O(\Delta t)$
FD-Ord 2	(3.34)	$\frac{2C_1}{3}\|(L^0)^2\mu^i\|_2\Delta t^2$	(3.46)	$O(\Delta t^2)$
Trapezoidal	(3.39)	$\frac{C_1}{12}\|(L^0)^2\mu^i\|_2\Delta t^2$	(3.48)	$O(\Delta t^2)$
Drift-Sub	-	-	(3.44)	$O(\Delta t)$

Table 3.1: Summary of the methods for estimating the drift (μ^i) and the diffusion ($\Sigma^{i,j}$).

3.7 Numerical Examples

In this section, we demonstrate the performance of the methods presented above on numerical examples. For each example, we will generate approximations $\tilde{\alpha}^i \approx \alpha^i$ and $\tilde{\beta}^{i,j} \approx \beta^{i,j}$. However, to present the data more simply, instead of computing the mean and mean squared error for each vector $\tilde{\alpha}^i$ and $\tilde{\beta}^{i,j}$, we will be aggregating the errors across all the coefficients. We will compute the mean error, normalized for the norms of α^i and $\beta^{i,j}$ using

$$Err_m = \left(\frac{\sum_{i=1}^d \|\mathbb{E}(\tilde{\alpha}^i) - \alpha^i\|_2^2}{\sum_{i=1}^d \|\alpha^i\|_2^2} \right)^{\frac{1}{2}} \quad \text{or} \quad Err_m = \left(\frac{\sum_{i \geq j \geq 1}^d \|\mathbb{E}(\tilde{\beta}^{i,j}) - \beta^{i,j}\|_2^2}{\sum_{i \geq j \geq 1}^d \|\beta^{i,j}\|_2^2} \right)^{\frac{1}{2}}.$$

Similarly, we will calculate the normalized variance

$$Err_{var} = \frac{\sum_{i=1}^d Var\left(\tilde{\alpha}^i\right)}{\sum_{i=1}^d \|\alpha^i\|_2^2} \quad \text{or} \quad Err_{var} = \frac{\sum_{i \geq j \geq 1}^d Var\left(\tilde{\beta}^{i,j}\right)}{\sum_{i \geq j \geq 1}^d \|\beta^{i,j}\|_2^2}.$$

Since these errors are based on aggregating the errors for all of the components of α^i or $\beta^{i,j}$, they will demonstrate the same convergence rates as in Theorems 4-9. The constants, however, may be different.

For each example, we will estimate the drift and diffusion using each of the methods described. The drift will be estimated using the first and second order forward differences, as well as the trapezoidal approximation. For the diffusion, we will use the first and second order forward differences, the drift-subtracted first order difference, and the trapezoidal method. For the drift-subtracted estimation, we will use the estimation for μ generated by the first order forward difference. Similarly, for the trapezoidal approximation for Σ, we will use the estimate generated by the trapezoidal approximation for μ.

3.7.1 Double Well Potential

Consider the SDE

$$dX_t = \left(-X_t^3 + \frac{1}{2}X_t\right) dX_t + \left(1 + \frac{1}{4}X_t^2\right) dW_t \tag{3.49}$$

This equation represents a diffusion in the double well potential $U(x) = \frac{1}{4}x^4 - \frac{1}{2}x^2$. Without the diffusion, the trajectories of this system will settle towards one of two fixed points, depending on which basin of attraction it started in. With the stochastic forcing, the trajectories will move around in one basin of attraction until it gets sufficiently perturbed to move to the other basin. We also note that for the majority of the trajectory, the state will be near the point where the drift is zero, so the dynamics will be dominated by the diffusion. At these points, the trajectory will behave similarly to Brownian motion.

For the SINDy algorithm, we will use a dictionary of monomials in x up to degree 14:

$$\theta(x) = \begin{bmatrix} 1 & x & \dots & x^{14} \end{bmatrix}.$$

This basis will be used to estimate both the drift and diffusion. To generate the data for the algorithm, we simulated (3.49) using the Euler-Maruyama method 1,000 times with a time step of 2×10^{-4} seconds and a duration of 20,000 seconds. The initial condition was drawn randomly for each simulation from the standard normal distribution. The SINDy methods were then run on the data from each simulation for different sampling periods, Δt, and lengths of the trajectory, T. We use a minimum Δt of 0.002 so the simulation has a resolution of at least ten steps between each data sample. The truncation parameters for the sparse solver were set at $\lambda = 0.005$ for the drift and $\lambda = 0.001$ for the diffusion.

As can be seen from from figure 3.7.1, the expected errors in all three methods were converging to zero as $\Delta t \to 0$. For small Δt, the expected estimate was within 1% of the true value. Additionally, the two higher order methods showed that, in expectation, they produce more accurate results and appear to converge more quickly, in line with Theorems 4, 6, and 7. For these methods, the expected error was as much as an order of magnitude smaller, depending on the size of Δt.

The variance, however, is rather large relative to the size of the expected error for all three methods. This is likely due to the system tending to settle towards the points $x = \pm 1/\sqrt{2}$ where the drift is zero. Near these points, the dynamics are dominated by the diffusion, making it difficult to estimate the drift. As can be seen (noting the scale of the center plot), the variance does not change a great amount as Δt decreases, as is predicted for the estimates of the drift. As shown in the rightmost plot, the variance decreases as the length of the trajectory increases. In order to more fully benefit from using the higher order methods to the full extent, we would need a long enough trajectory

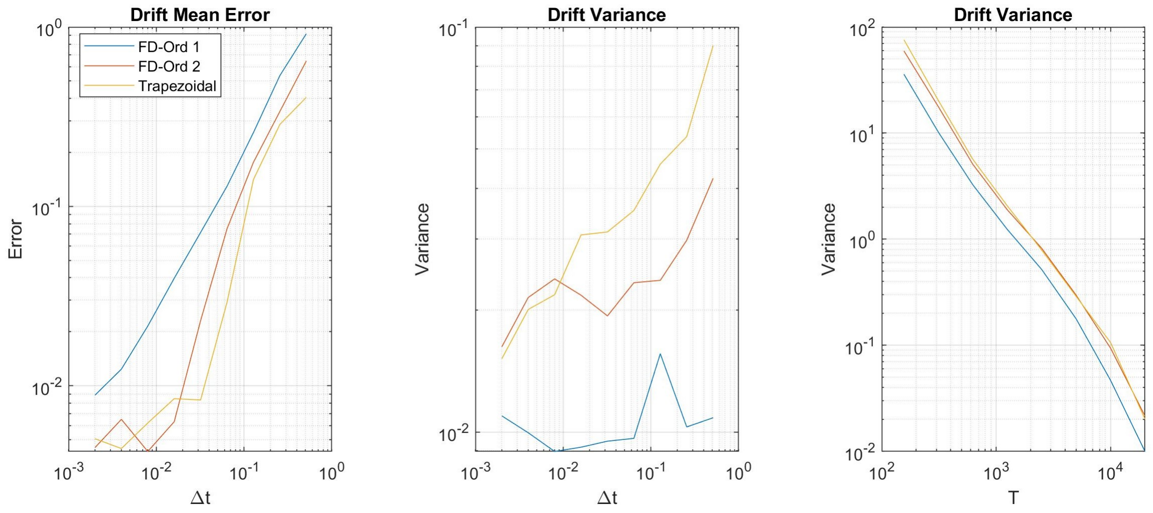

Figure 3.1: (Left) The mean error in the estimation of the drift coefficients for the double well system (3.49) is plotted as a function of Δt. The error is approximated using 1,000 trajectories of length $T = 20,000$ seconds.
(Center, Right) The variance for each method is plotted against the sampling period, Δt, and the trajectory length, T. For the trajectory length is fixed at $T = 20,000$ seconds for the center plot, while the sampling period was fixed at $\Delta t = 0.004 = 4 \times 10^{-3}$ for the rightmost plot.

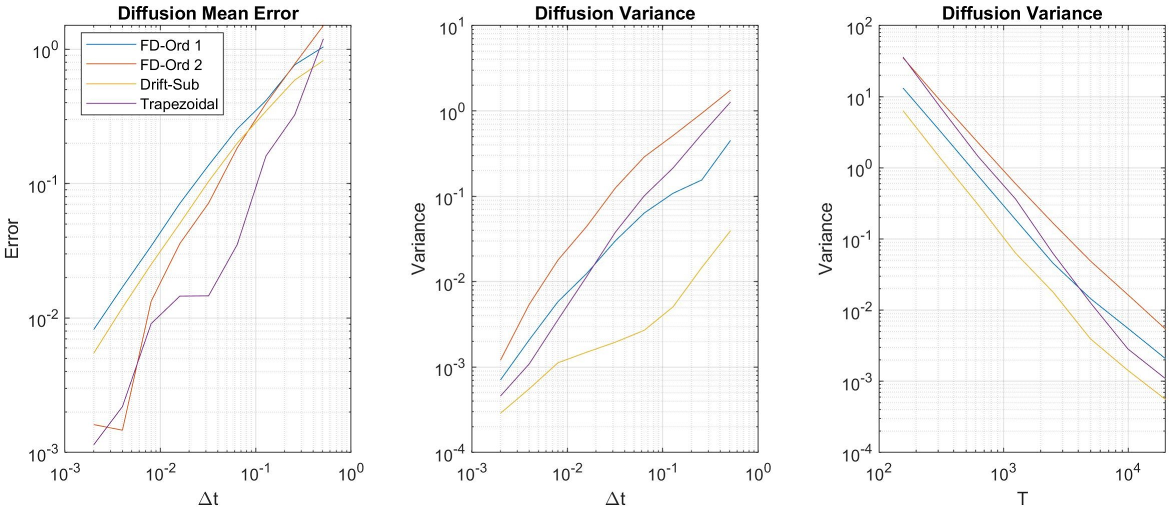

Figure 3.2: (Left) The mean error in the estimation of the diffusion coefficients for the double well system (3.49) is plotted as a function of Δt. The error is approximated using 1,000 trajectories of length $T = 20,000$ seconds.
(Center, Right) The variance for each method is plotted against the sampling period, Δt, and the trajectory length, T. For the trajectory length is fixed at $T = 20,000$ seconds for the center plot, while the sampling period was fixed at $\Delta t = 0.04 = 4\times 10^{-3}$ for the rightmost plot.

to control the variance.

For the diffusion, figure 3.7.1 shows again that, as $\Delta t \to 0$, all of the methods do indeed converge in expectation. The Drift-Sub method slightly outperforms FD-Ord 1, the error is typically reduced by about $20\% - 30\%$. Of the two higher order method, the trapezoidal method typically yields the best results, often an order of magnitude better than FD-Ord 1. FD-Ord 2 also gives substantial improvements for small Δt. Contrary to the drift, the variance in the estimate of the diffusion does decrease as Δt goes to zero. The decrease appears to be roughly proportional to both Δt and $1/T$, which is in line with the Theorems 5, 8, and 9.

3.7.2 Noisy Van-Der-Pol Oscillator

Consider the ODE

$$\begin{bmatrix} \dot{x}^1 \\ \dot{x}^2 \end{bmatrix} = \begin{bmatrix} x^2 \\ (1-(x^1)^2)x^2 - x^1 \end{bmatrix}.$$

This is the Van-Der-Pol equation, which describes a nonlinear oscillator. We can perturb this equation by adding noise, we get the SDE

$$\begin{bmatrix} dX_t^1 \\ dX_t^2 \end{bmatrix} = \begin{bmatrix} X_t^2 \\ (1-(X_t^1)^2)X_t^2 - X_t^1 \end{bmatrix} dt + \sigma(X_t)dW_t, \tag{3.50}$$

where W_t is a two dimensional Wiener process. For the simulations, we let

$$\sigma(x) = \frac{1}{2}\begin{bmatrix} 1+0.3x^2 & 0 \\ 0 & 0.5+0.2x^1 \end{bmatrix}.$$

We chose this system to represent a different type of limiting behavior. For this system, the dynamics settle around a limit cycle. While they will have a certain amount of randomness, the trajectories will demonstrate an approximately cyclic behavior. In particular, this also means that the drift will rarely be near zero, as opposed to the previous example where the drift was often small.

The dictionary we will use for the SINDy algorithm consists of all monomials in x^1 and x^2 up to degree 6:

$$\theta(x) = \begin{bmatrix} 1 & x^1 & x^2 & x^1x^2 & \dots & (x^1)^2(x^2)^4 & x^1(x^2)^5 & (x^2)^6 \end{bmatrix}.$$

This basis will be used to estimate both the drift and diffusion. To generate the data for the algorithm, we simulated (3.50) using the Euler-Maruyama method 1,000 times with a time step of 2×10^{-5} seconds and a duration of 1,000 seconds. Each component of the initial condition was drawn randomly for each simulation from the standard normal distribution. The SINDy methods were then run on the data from each simulation for different sampling periods, Δt, and lengths of the trajectory, T. As before, we use $\Delta t \geq 2 \times 10^{-4}$ to ensure that sampling period is at least 10 times the simulation time step. The truncation parameters for the sparse solver were set at $\lambda = 0.05$ for the drift and $\lambda = 0.02$ for the diffusion.

In figure 3.7.2, we first note that the variance very quickly drops to about 5×10^{-5} and stays roughly constant as Δt decreases. This falls very much in line with the Theorems 4, 6, and 7 which assert that the variance does not depend on the sample frequency, it only decreases with the trajectory length T. For the expected error, the FD-Ord 2 and trapezoidal methods show drastic improvements over FD-Ord 1, with the trapezoidal

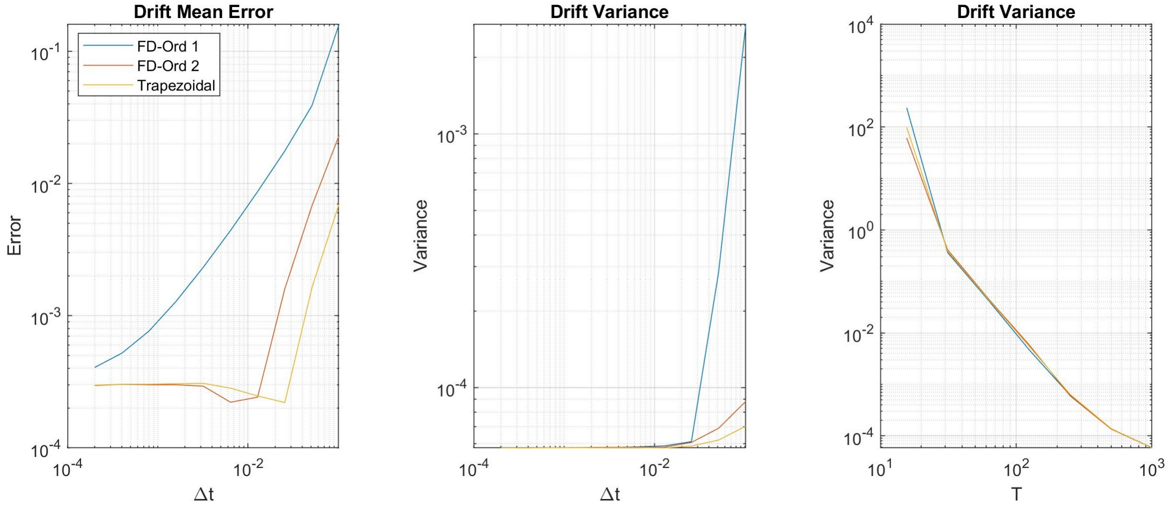

Figure 3.3: (Left) The mean error in the estimation of the drift coefficients for the Van-Der-Pol system (3.50) is plotted as a function of Δt. The error is approximated using 1,000 trajectories of length $T = 1,000$ seconds.
(Center, Right) The variance for each method is plotted against the sampling period, Δt, and the trajectory length, T. For the trajectory length is fixed at $T = 1,000$ seconds for the center plot, while the sampling period was fixed at $\Delta t = 0.008 = 8 \times 10^{-3}$ for the rightmost plot.

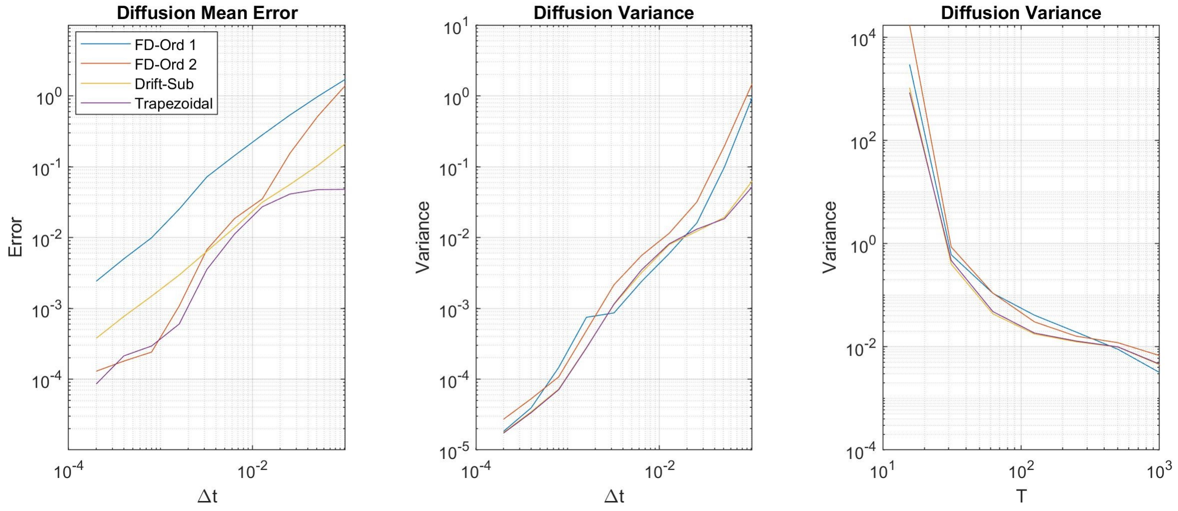

Figure 3.4: (Left) The mean error in the estimation of the diffusion coefficients for the Van-Der-Pol system (3.50) is plotted as a function of Δt. The error is approximated using 1,000 trajectories of length $T = 1,000$ seconds.
(Center, Right) The variance for each method is plotted against the sampling period, Δt, and the trajectory length, T. For the trajectory length is fixed at $T = 1,000$ seconds for the center plot, while the sampling period was fixed at $\Delta t = 0.008 = 8 \times 10^{-3}$ for the rightmost plot.

method reducing the error by almost two orders of magnitude on some values of Δt. For the larger Δt, the slopes of the graphs demonstrate that these methods are converging at twice the order of the first order forward difference, as predicted by Theorems 4, 6, and 7. However, both second order methods quickly reach a point where the performance remained constant at about 2×10^{-4}. This is due to the lack of data to average over the random variation to sufficient precision. With sufficient data, we would expect the performance to continue to improve proportionally to Δt^2.

For the diffusion, figure 3.7.2 demonstrates a greater separation in the performance of the different methods compared to the double well system. Here, the FD-Ord 1 and drift subtracted methods both demonstrate the same first order convergence, as predicted in Theorem 5, but the drift subtracted method demonstrates a substantially lower error, ranging from half an order to almost a full order of magnitude better. FD-Ord 2 begins at roughly the same error as FD-Ord 1 for large Δt, but convergences faster, as predicted by Theorem 8, until it gives over an order of magnitude improvement for small Δt. Finally, although it is difficult to judge the speed of convergence for the trapezoidal method, it gives the most accurate results across all Δt. The variance for all of the methods behave similarly to the Double Well example and as expected, decreasing as $\Delta t \to 0$ and $T \to \infty$.

3.7.3 Noisy Lorenz Attractor

Consider the ODE

$$\dot{x} = \begin{bmatrix} \dot{x}^1 \\ \dot{x}^2 \\ \dot{x}^3 \end{bmatrix} = \begin{bmatrix} 10(x^2 - x^1) \\ x^1(28 - x^3) - x^2 \\ x^1 x^2 - \frac{8}{3}x^3 \end{bmatrix} = f(x).$$

This is the Lorenz system, which is famously a chaotic system exhibiting a strange attractor. If we perturb this equation by adding noise, we get the SDE

$$dX_t = f(X_t)dt + \sigma(X_t)dW_t, \tag{3.51}$$

where W_t is a three dimensional Wiener process. For this example, we let

$$\sigma(x) = \begin{bmatrix} 1+\sin(x^2) & 0 & \sin(x^1) \\ 0 & 1+\sin(x^3) & 0 \\ \sin(x^1) & 0 & 1-\sin(x^2) \end{bmatrix}.$$

To generate the data for the algorithm, we simulated (3.50) using the Euler-Maruyama method 1,000 times with a time step of 2×10^{-5} seconds and a duration of 1,000 seconds. Each component of initial condition was drawn randomly for each simulation from the standard normal distribution. The SINDy methods were then run on the data from each simulation for different sampling periods, Δt, and lengths of the trajectory, T. The truncation parameters for the sparse solver were set at $\lambda = 0.05$ for the drift and $\lambda = 0.02$ for the diffusion.

We will use different dictionaries to estimate the drift and diffusion. For the drift, the dictionary consists of all monomials in x^1, x^2, and x^3 up to degree 4:

$$\theta(x) = \begin{bmatrix} 1 & x^1 & x^2 & \dots & x^1x^2(x^3)^3 & (x^2)^2(x^3)^3 & x^2(x^3)^4 & (x^3)^5 \end{bmatrix}.$$

As before, figure 3.7.3 shows that the variance of the estimate for the drift decreases steadily as $T \to \infty$, while it approaches a minimum value as Δt decreases and remains constant after reaching that minimum. In terms of the mean error, this example gives

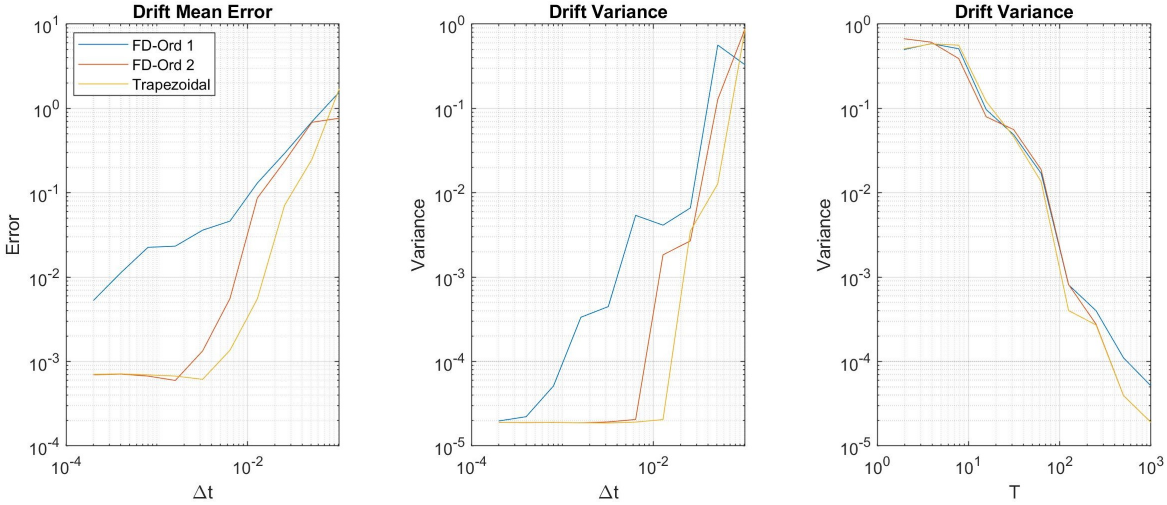

Figure 3.5: (Left) The mean error in the estimation of the drift coefficients for the Lorenz system (3.51) is plotted as a function of Δt. The error is approximated using 1,000 trajectories of length $T = 1,000$ seconds.
(Center, Right) The variance for each method is plotted against the sampling period, Δt, and the trajectory length, T. For the trajectory length is fixed at $T = 1,000$ seconds for the center plot, while the sampling period was fixed at $\Delta t = 0.08 = 8\times10^{-2}$ for the rightmost plot.

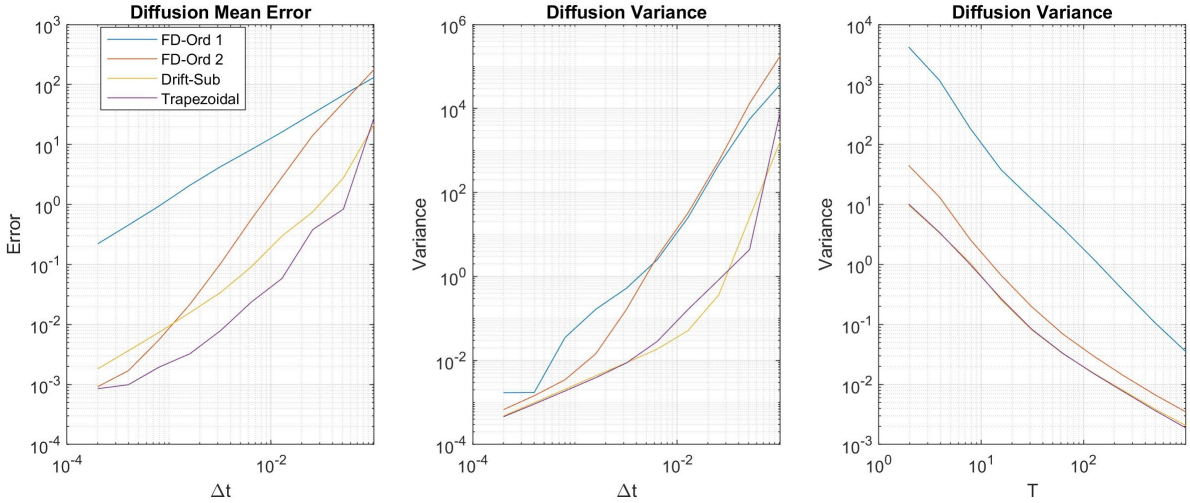

Figure 3.6: (Left) The mean error in the estimation of the diffusion coefficients for the Lorenz system (3.51) is plotted as a function of Δt. The error is approximated using 1,000 trajectories of length $T = 1,000$ seconds.
(Center, Right) The variance for each method is plotted against the sampling period, Δt, and the trajectory length, T. For the trajectory length is fixed at $T = 1,000$ seconds for the center plot, while the sampling period was fixed at $\Delta t = 0.02 = 2\times 10^{-2}$ for the rightmost plot.

the clearest confirmation of the convergence rates demonstrated in Theorems 4, 6, and 7. The slopes of the plots show that the error with FD-Ord 1 is roughly proportional to Δt, while the FD-Ord 2 and trapezoidal methods converge at double the rate. For small Δt, the second order methods do not seem to improve, due to the lack of sufficient data to compute the averages to high enough precision.

To estimate the diffusion, we used a dictionary consisting of all monomials in $\sin(x^1)$, $\sin(x^2)$, and $\sin(x^3)$ up to degree four:

$$\theta(x) = \begin{bmatrix} 1 & \sin(x^1) & \sin(x^2) & \dots & \sin(x^1)\sin(x^2)\sin^2(x^3) & \sin(x^2)\sin^3(x^3) & \sin^4(x^3) \end{bmatrix}.$$

The error plot in figure 3.7.3 provides the most compelling example of the improvements of the higher order methods for estimating the diffusion. FD-Ord 1 clearly demonstrates its order one convergence as $\Delta t \to 0$ (Theorem 5), but the error is quite large compared to the other methods. Even at our highest sampling frequency, $\Delta t = 2 \times 10^-4$, we only get slightly accurate results, with an error over 20%. For this system, the drift subtracted method, although still first order, provides great improvements over FD-Ord 1, nearly two orders of magnitude better for most Δt. FD-Ord 2 also demonstrates the second order convergence given in Theorem 8, giving very accurate results for small Δt. Finally, the best performance again comes from the Trapezoidal method, which gives the best performance across all Δt. As expected from Theorem 9, we can see that it converges faster than FD-Ord 1, but the convergence rate is not as clear as that of the other methods.

As for the variance, it decreased for all four methods as T increased and Δt decreased, as expected. However, the Trapezoidal and drift subtracted methods both showed a substantially lower variance. This is likely because the drift tends to dominate the diffusion in this system. Both the drift subtracted and trapezoidal methods correct for this, preventing the drift from having an effect on the estimate of the diffusion.

3.8 Conclusion

As was shown in this and previous papers ([6],[16],[11]), the SINDy algorithm can be used to accurately estimate the parameters of a stochastic differential equation. However, the significant amount of noise involved requires one to use either use great deal of data (i.e. a long time series) and/or methods which improve the robustness of SINDy to noise. Unfortunately, even if SINDy should identify all of the correct dictionary functions present in the dynamics, we showed that the sampling frequency limits the accuracy of the results when using the first order Kramer-Moyal formulas to estimate the drift and diffusion. The necessity for high sampling frequencies, combined with long trajectories, make SINDy a data hungry algorithm.

The higher order estimates presented in this paper allow us to overcome the $O(\Delta t)$ convergence given in [6]. With the higher order methods we can compute accurate estimations of the SDEs using far lower sampling frequencies. In addition to making SINDy a more accurate system identification tool, these improvements also greatly reduce the data requirements to feed the algorithm. By achieving accurate results at lower sampling frequencies we can reduce the data acquisition constraint, which makes SINDy a more feasible system identification method for SDEs.

Chapter 4

Conclusion

In this dissertation, we discussed the application of the stochastic Koopman operator to random dynamical systems. The Koopman operator allows us to represent nonlinear systems with a linear operator and give a spectral expansion of the evolution of observables. We demonstrate DMD algorithms which allow us to compute finite sections of the stochastic Koopman operator and approximations of the spectral expansion from data. We then continue onto algorithms which allow us to identify nonlinear representation SDEs. We improve the SINDy algorithm by introducing methods of approximating the drift and diffusion functions with higher order rates of convergence.

4.1 Summary of “Noise Resistant Dynamic Mode Decomposition”

In this section, we analyzed the convergence of DMD algorithms for random dynamical systems, culminating in the introduction of a new DMD algorithm that converges to the spectrum of the stochastic Koopman operator in the presence of both random dynamics and noisy observables. This allows us to avoid the bias in standard DMD algorithms that can come from “overfitting” to the noise. We then specialized the algorithm to handle observables with i.i.d. measurement noise and time delayed observables and showed that measurements of a single set of observables was sufficient to generate an approximation of the stochastic Koopman operator. In particular, we demonstrated that a single trajectory of a single observable could be used to generate a Krylov subspace of the operator, which allows us to use DMD without needing to choose a basis of observables.

This algorithm provides a method for modeling complex systems where a deterministic model is unfeasible. This could be because a full state model would be to complex, observables of the full state are unavailable, or measurements come with uncertainty. A possible extension of this algorithm could adapt it to handle data from systems with control inputs, which could be used to develop control algorithms for random dynamical systems.

4.2 Summary of "Numerical Methods for Stochastic SINDy"

In the section, we analyzed the performance of SINDy for stochastic differential equations. As was shown in this and previous papers ([6],[16],[11]), the SINDy algorithm can be used to accurately estimate the parameters of a stochastic differential equation. However, the significant amount of noise involved requires one to use either use great deal of data (i.e. a long time series) and/or methods which improve the robustness of SINDy to noise. Unfortunately, even if SINDy should identify all of the correct dictionary functions present in the dynamics, we showed that the sampling frequency limits the accuracy of the results when using the first order Kramer-Moyal formulas to estimate the drift and diffusion. The necessity for high sampling frequencies, combined with long trajectories, make SINDy a data hungry algorithm.

The higher order estimates presented in this paper allow us to overcome the $O(\Delta t)$ convergence given in [6]. With the higher order methods we can compute accurate estimations of the SDEs using far lower sampling frequencies. In addition to making SINDy a more accurate system identification tool, these improvements also greatly reduce the data requirements to feed the algorithm. By achieving accurate results at lower sampling frequencies we can reduce the data acquisition constraint, which makes SINDy a more feasible system identification method for SDEs.

Bibliography

[1] Hassan Arbabi and Igor Mezic. Computation of transient koopman spectrum using hankel-dynamic mode decompoisition. In *APS Division of Fluid Dynamics Meeting Abstracts*, pages G1–009, 2017.

[2] Hassan Arbabi and Igor Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.

[3] Ludwig Arnold. *Random Dynamical Systems.* Springer-Verlag Berlin Heidelberg, 1998.

[4] Ludwig Arnold and Hans Crauel. Random dynamical systems. In *Lyapunov exponents*, pages 1–22. Springer, 1991.

[5] Erik Berger, Mark Sastuba, David Vogt, Bernhard Jung, and Heni Ben Amor. Estimation of perturbations in robotic behavior using dynamic mode decomposition. *Advanced Robotics*, 29(5):331–343, 2015.

[6] Lorenzo Boninsegna, Feliks Nüske, and Cecilia Clementi. Sparse learning of stochastic dynamical equations. *The Journal of chemical physics*, 148(24):241723, 2018.

[7] Daniel Bruder, Xun Fu, and Ram Vasudevan. Advantages of bilinear koopman realizations for the modeling and control of systems with unknown dynamics. *IEEE Robotics and Automation Letters*, 6(3):4369–4376, 2021.

[8] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eurika Kaiser, and J Nathan Kutz. Chaos as an intermittently forced linear system. *Nature communications*, 8(1):1–9, 2017.

[9] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[10] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Sparse identification of nonlinear dynamics with control (sindyc). *IFAC-PapersOnLine*, 49(18):710–715, 2016.

[11] Jared L Callaham, J-C Loiseau, Georgios Rigas, and Steven L Brunton. Nonlinear stochastic modelling with langevin regression. *Proceedings of the Royal Society A*, 477(2250):20210092, 2021.

[12] Torsten Carleman. Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. *Acta Mathematica*, 1932.

[13] Xi Chen and Ilya Timofeyev. Non-parametric estimation of stochastic differential equations from stationary time-series. *Journal of Statistical Physics*, 186:1–31, 2022.

[14] Hans Crauel. Markov measures for random dynamical systems. *Stochastics: An International Journal of Probability and Stochastic Processes*, 37(3):153–173, 1991.

[15] Nelida Črnjarić-Žic, Senka Maćešić, and Igor Mezić. Koopman operator spectrum for random dynamical systems. *Journal of Nonlinear Science*, pages 1–50, 2019.

[16] Min Dai, Ting Gao, Yubin Lu, Yayun Zheng, and Jinqiao Duan. Detecting the maximum likelihood transition path from data of stochastic dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(11):113124, 2020.

[17] Scott TM Dawson, Maziar S Hemati, Matthew O Williams, and Clarence W Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):42, 2016.

[18] Evgeniĭ Borisovich Dynkin and Evgenij Borisovič Dynkin. *Markov processes*. Springer, 1965.

[19] Urban Fasel, J Nathan Kutz, Bingni W Brunton, and Steven L Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A*, 478(2260):20210904, 2022.

[20] I.I. Gikhman, A.V. Skorokhod, and S. Kotz. *The Theory of Stochastic Processes: I*. Classics in Mathematics. Springer Berlin Heidelberg, 2004.

[21] Mircea Grigoriu. *Stochastic calculus: applications in science and engineering*. Springer, 2002.

[22] David A Haggerty, Michael J Banks, Patrick C Curtis, Igor Mezić, and Elliot W Hawkes. Modeling, reduction, and control of a helically actuated inertial soft robotic arm via the koopman operator. *arXiv preprint arXiv:2011.07939*, 2020.

[23] Maziar S Hemati, Clarence W Rowley, Eric A Deem, and Louis N Cattafesta. De-biasing the dynamic mode decomposition for applied koopman spectral analysis of noisy datasets. *Theoretical and Computational Fluid Dynamics*, 31(4):349–368, 2017.

[24] Mihailo R Jovanović, Peter J Schmid, and Joseph W Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26(2), 2014.

[25] Oliver Junge, Jerrold E Marsden, and Igor Mezic. Uncertainty in the dynamics of conservative maps. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 2, pages 2225–2230. IEEE, 2004.

[26] Kadierdan Kaheman, J Nathan Kutz, and Steven L Brunton. Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A*, 476(2242):20200279, 2020.

[27] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A*, 474(2219):20180335, 2018.

[28] Yuzuru Kato, Jinjie Zhu, Wataru Kurebayashi, and Hiroya Nakao. Asymptotic phase and amplitude for classical and semiclassical stochastic oscillators via koopman operator theory. *Mathematics*, 9(18):2188, 2021.

[29] Rachael T Keller and Qiang Du. Discovery of dynamics using linear multistep methods. *SIAM Journal on Numerical Analysis*, 59(1):429–455, 2021.

[30] Rafail Khasminskii. *Stochastic stability of differential equations*, volume 66. Springer Science & Business Media, 2011.

[31] Yuri Kifer. *Ergodic Theory of Random Transformations.* Birkhäuser Boston, Inc, 1986.

[32] Peter E Kloeden and Eckhard Platen. Numerical solution of stochastic differential equations. In *Numerical solution of stochastic differential equations*, pages 103–160. Springer, 1992.

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

[34] Bernard O Koopman and J v Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263, 1932.

[35] Elias B Kosmatopoulos, Marios M Polycarpou, Manolis A Christodoulou, and Petros A Ioannou. High-order neural network structures for identification of dynamical systems. *IEEE transactions on Neural Networks*, 6(2):422–431, 1995.

[36] S Narendra Kumpati, Parthasarathy Kannan, et al. Identification and control of dynamical systems using neural networks. *IEEE Transactions on neural networks*, 1(1):4–27, 1990.

[37] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Other titles in applied mathematics. Society for Industrial and Applied Mathematics SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104, Philadelphia, Pennsylvania, 2016.

[38] Yueheng Lan and Igor Mezić. Linearization in the large of nonlinear systems and koopman operator spectrum. *Physica D: Nonlinear Phenomena*, 242(1):42–53, 2013.

[39] Todd K Leen, Robert Friel, and David Nielsen. Eigenfunctions of the multidimensional linear noise fokker-planck operator via ladder operators. *arXiv preprint arXiv:1609.01194*, 2016.

[40] L. Ljung. *System Identification: Theory for the User*. Pearson Education, 1998.

[41] Niall M Mangan, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):52–63, 2016.

[42] Alexandre Mauroy and Igor Mezić. On the use of fourier averages to compute the global isochrons of (quasi) periodic dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(3), 2012.

[43] Alexandre Mauroy, Igor Mezić, and Jeff Moehlis. Isostables, isochrons, and koopman spectrum for the action–angle representation of stable fixed point dynamics. *Physica D: Nonlinear Phenomena*, 261:19–30, 2013.

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

[45] Daniel A Messenger and David M Bortz. Weak sindy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021.

[46] Daniel A Messenger and David M Bortz. Weak sindy: Galerkin-based data-driven model selection. *Multiscale Modeling & Simulation*, 19(3):1474–1497, 2021.

[47] I Mezic and Andrzej Banaszuk. Comparison of systems with complex behavior: Spectral methods. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, volume 2, pages 1224–1231. IEEE, 2000.

[48] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.

[49] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.

[50] Igor Mezić. Analysis of fluid flows via spectral properties of the koopman operator. *Annual review of fluid mechanics*, 45:357–378, 2013.

[51] Igor Mezic. Koopman operator, geometry, and learning. *arXiv preprint arXiv:2010.05377*, 2020.

[52] Igor Mezić. Spectrum of the koopman operator, spectral expansions in functional spaces, and state-space geometry. *Journal of Nonlinear Science*, 30(5):2091–2145, 2020.

[53] Igor Mezić. On numerical approximations of the koopman operator. *Mathematics*, 10(7):1180, 2022.

[54] Igor Mezić and Andrzej Banaszuk. Comparison of systems with complex behavior. *Physica D: Nonlinear Phenomena*, 197(1-2):101–133, 2004.

[55] Taijiro Ohno. Asymptotic behaviors of dynamical systems with random parameters. *Publications of the research Institute for Mathematical Sciences*, 19(1):83–98, 1983.

[56] Samuel E Otto and Clarence W Rowley. Koopman operators for estimation and control of dynamical systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:59–87, 2021.

[57] Grigorios A Pavliotis. *Stochastic processes and applications*. Springer, 2016.

[58] Karl Endel Petersen. *Ergodic Theory*. Cambridge studies in advanced mathematics; 2. Cambridge University Press, Cambridge [Cambridgeshire];, 1983.

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

[60] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018.

[61] Joshua L Proctor and Philip A Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International health*, 7(2):139–145, 2015.

[62] Olav Reiersøl. *Confluence analysis by means of instrumental sets of variables*. PhD thesis, Almqvist & Wiksell, 1945.

[63] Olav Reiersøl. Identifiability of a linear relation between variables which are subject to error. *Econometrica: Journal of the Econometric Society*, pages 375–389, 1950.

[64] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, Dans Henningson, et al. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641(1):115–127, 2009.

[65] Walter Rudin. *Functional analysis 2nd ed.* McGraw Hill, 1991.

[66] Walter Rudin. *Fourier analysis on groups.* Courier Dover Publications, 2017.

[67] Peter Schmid and Joern Sesterhenn. Dynamic mode decomposition of numerical and experimental data. *APS*, 61:MR–007, 2008.

[68] Peter J Schmid. Dynamic mode decomposition and its variants. *Annual Review of Fluid Mechanics*, 54:225–254, 2022.

[69] Torsten Söderström and Petre Stoica. Instrumental variable methods for system identification. *Circuits, Systems and Signal Processing*, 21(1):1–9, 2002.

[70] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Subspace dynamic mode decomposition for stochastic koopman analysis. *Physical Review E*, 96(3):033310, 2017.

[71] A Tantet, MD Chekroun, HA Dijkstra, and JD Neelin. Mixing spectrum in reduced phase spaces of stochastic differential equations. *Part II: Stochastic Hopf Bifurcation. ArXiv e-prints*, 2017.

[72] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[73] Mathias Wanner and Igor Mezic. Robust approximation of the stochastic koopman operator. *SIAM Journal on Applied Dynamical Systems*, 21(3):1930–1951, 2022.

[74] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[75] Kōsaku Yosida. *Functional analysis.* Springer Science & Business Media, 2012.

[76] Linan Zhang and Hayden Schaeffer. On the convergence of the sindy algorithm. *Multiscale Modeling & Simulation*, 17(3):948–972, 2019.