Control Theory for *C. elegans* Calcium Imaging Dynamics

Charles Fieseler

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

Doctor of Philosophy

University of Washington

2020

Reading Committee:

J. Nathan Kutz, Chair

Subhadeep Gupta

Eric Shea-Brown

Program Authorized to Offer Degree:
Physics

University of Washington

Abstract

Control Theory for *C. elegans* Calcium Imaging Dynamics

Charles Fieseler

Chair of the Supervisory Committee:
Yasuko Endo Professor J. Nathan Kutz
Applied Mathematics

This thesis contains several projects under two main headings: data-driven modeling, and *C. elegans* applications. Most of these works build to the main core of this thesis: applying novel control theoretic methods to *C. elegans* Calcium imaging data. Sparse optimization techniques are used to provide a unsupervised solution to a pervasive problem in complex systems modeling: some features of the data lie outside the modeling assumptions, which can distort the entire model and interpretations thereof. The method divides the time series into regions that are explainable within a given modeling framework (intrinsic dynamics) and those that are not (control signals), and mathematically characterizes how they affect each other. In addition, previously uncharacterized neurons are identified that encode these control signals, which can inform future experimental work. Other projects include advances to the underlying mathematical methods, as well as attempts to reconstruct network structure from data, along with significant caveats.

TABLE OF CONTENTS

Page

List of Figures iii

Glossary x

Chapter 1: Introduction 1
1.1 Why Data-Driven? 1
1.2 Low Dimensionality 2
1.3 Control Theory 4
1.4 *C. elegans* as a model organism 9

Chapter 2: Background 12
2.1 Data-driven methods 12
2.2 Biology 24
2.3 Biomechanical modeling 27

Chapter 3: Methodological extensions 41
3.1 Infinite series DMD 41
3.2 Sparse optimization for control signals 45
3.3 Subsampling to learn control signals 47
3.4 Using DMD as a network reconstruction method 51

Chapter 4: Biological Results 70
4.1 Omega turns in a biomechanical model 71
4.2 Neuron population activity as a controlled dynamical system 74
4.3 Learning control signals: Optimization 78
4.4 Neural encoding of control signals 79

Chapter 5: Discussion 83
5.1 Overview: "Complexity-scale" Separation 83
5.2 Biomechanical Modeling 84
5.3 Data-Driven Methods 85
5.4 Scale separation in *C. elegans* 88

Bibliography 92

Appendix A: Appendix A: Information theory for control signals 119
A.1 Biological systems as feedback-controlled systems 119
A.2 Learning control signals from data 120
A.3 Variable selection 128
A.4 Reconstructions for additional individuals 129

LIST OF FIGURES

Figure Number Page

1.1 a) A simple example system: a bouncing ball. The intrinsic dynamics is the acceleration due to gravity, which is linear in the velocity-acceleration basis. The external spatially-dependent forcing is provided by the ground, and in this example there is an additional time-dependent forcing, e.g. a kick. b) The observed data in this simple case have obvious discontinuities, and the two types are easy to distinguish. c) The control signal provided by the ground, which is actually a function of space, and the external kick, which is purely a function of time. 7

2.1 Figure from [1]. PCA returns orthogonal vectors that capture the largest variance, in order. 13

2.2 Figure reproduced from [47]. SINDy produces a set of governing equations (ODEs) from data and a hand-picked library of terms. 16

2.3 A graphical explanation of DMD and DMDc in discrete time. 20

2.4 A 3-step framework for modeling neurosensory integration. 1) Transition signals are learned from data with an assumption of linear dynamics. 2) A DMDc model is learned which uses dynamics, transition signals, and actuation. These are global models, and are capable of reconstructing much of the data dynamically from an initial state. 3) Where and at what timescales control signals are encoded in the neural activity is studied using sparse linear models. Bottom: the sensory-computation-behavior pathway. Each term in the above equations can be freely translated into this biological process. Transitions (Green) actuate neurons via their own connectivity (Yellow). Neuron traces (Blue) evolve according to intrinsic dynamics (Red), and also encode (Light Blue) the transition signals (Green). 36

2.5 Figure from the website www.wormwiring.org [3], an excellent compilation of many connectome-related resources. 37

2.6 Biomechanical model of C elegans. Based off of [38]. a) The body has 12 segments. (b) and (c) Each segment has two rigid vertical components and four damped spring components. The diagonal (blue) elements are passive; the horizontal (red) elements are active and controlled by the neuron voltages. d,e) Each segment also has a simplified connectome model, with four pairs of ventral and dorsal motor neurons, and a pair of excitatory B-class neurons and inhibitory D-class neurons. These are activated by a toy "AVB-like" command neuron, for forward motion. f) Proprioception produces oscillation and more complex behavior. A small curvature will produce almost no proprioceptive signaling, but a stretched segment will. 38

2.7 Transition signals in *C. elegans*: Top: A calcium imaging trace of a neuron connected with the discrete reversal behavior. Behavioral labels are determined by experimentalists, as described in [152]. Green=Forward; Yellow=Reversal; Dark Blue=Ventral Turn; Light Blue=Dorsal Turn. Below: These labels can be reframed as "onset" signals, and are characteristically sparse in time. 39

2.8 Figure from [257], showing the experimental setup for imaging immobilized worms. 40

3.1 Comparisons of iDMD with four leading DMD methods in a small system ($n = r = r_0 = 2$). Overall, iDMD performs similarly to exact DMD (blue). a) The correlation of data reconstructions for different methods at different noise levels. b) The same methods and color schemes, but for one of the eigenvalues of the dynamical system as well as the unit circle (dashed black line). fb-DMD is particularly accurate, similar totls-DMD. However, the latter has a single outlier eigenvalue that drastically increase the variance of the eigenvalues. opt-DMD is generally the most accurate method available. 56

3.2 Comparisons with four leading DMD methods in a large system ($r = r_0 = 20, n = 70$). Overall, iDMD provides a performance boost over exact DMD. a) The correlation of data reconstructions for different methods at different noise levels. Even though the eigenvalues are relatively precise, if they are outside the unit circle then the reconstruction will quickly diverge. b) One of the eigenvalues and the unit circle (dashed black line). Forward-backward DMD (fb-DMD) is no longer very accurate and has unstable eigenvalues, i.e. outside the unit circle. iDMD is the most precise for both noise scenarios. . 57

3.3 Comparison of correlations between data and reconstructions, as well as eigenvalues in the actuated setting for a small system ($n = r = r_0 = 3; r_A = 2, r_B = 1$). Both DMD and iDMD methods have similar performance for a system of this size, though the eigenvalues for DMDc (b) are more biased. 58

3.4 Comparison of DMDc and iDMDc methods when the truncation rank is known ($r_0 = r = 25; r_A = 20, r_B = 5; n = 70$). a) In this relatively unrealistic scenario the methods perform similarly, although iDMDc is still more accurate particularly for higher noise levels. b) For $\sigma = 0.10$, the reconstructions are of similar quality. 59

3.5 Comparison of methods when the truncation rank is unknown ($r_0 = 25; r_A = 20, r_B = 5; r =$ Adaptively determined$; n = 70$). a) In this more realistic scenario, iDMDc is significantly more accurate for all noise levels. b) For $\sigma = 0.10$, the reconstruction of iDMDc (red) is clearly better. 60

3.6 Sensitivity of algorithms to choice of parameters for a small system ($r_0 = 5; r_A = 4, r_B = 1; n = 10$). a) DMDc has good performance only for the exact choice of the underlying dimension. b) For iDMDc, as the truncation parameter is increased, γ can be chosen so that the truncation error is lower than a tolerance value in this case 10^{-8}. The performance of the algorithm is slightly better than the best DMDc value, and is not very sensitive to the truncation value. 61

3.7 a) Example dataset with very low noise level ($\sigma = 0.01$). b) True and recovered control signal. 62

3.8 a) Example dataset with large noise level ($\sigma = 0.3$). The true dynamics is the same as Fig. 3.7 b) True and recovered control signal. 63

3.9 a) Example waveform for an acoustic emission from a mortar sample as it undergoes drying. b) Learned control signal 64

3.10 Beginning with data (in this case the Lorenz attractor with time-dependent external forcing), there are six steps to the model: 1. Fit a naive ODE. When integrated, this reconstruction will be very poor. 2. Find the posterior distribution of residuals of this naive ODE to numerically calculated derivatives. Note that this uses a collocation method, not integration. 3. Subsample the data, choosing the data points with small residual in the naive model. 4. Fit a "partial" model on the smaller sample of data. The control signals will not be captured, but the intrinsic dynamics may be. If they are not fit well, then looping back to step 2 will increase the quality of the subsample. 5. Using the residual of the final "partial" model, determine the control signals. 6. Fit the full control model, using control signals and data. This example is a chaotic system, so the individual trajectory will never be well reconstructed. Rather, the goal is to reconstruct the attractor. 65

3.11 a) Voltage data from a spiking neuron model. The membrane recovery variable (u) is not shown, but is provided to the algorithm. b) The "control signal" in this case is then the fast nonlinearity, instead of a truly external input. The location is learned very accurately but because it is modeled as a true discontinuity in the equations, the exact amplitude of the derivative will depend sensitively on the sampling rate and the exact method used to numerically differentiate. c) However, if a varying input current is also applied then this will show up as an additional control signal. d) The reset nonlinearity and and external voltage are learned as a single control signal. Importantly, the learned control signals are of very different orders of magnitude. 66

3.12 The algorithm works for a wide variety of classic models. 67

3.13 As noise increases, the performance of both this algorithm and the original SINDy algorithm can deteriorate. As the number of perturbations increase, this algorithm becomes progressively more important. 67

3.14 The true spatially-discretized matrix propagator that describes diffusive dynamics. 68

3.15 The true and best L2 fit spatially-discretized matrix propagator that describes diffusive dynamics. 68

3.16 Two matrices produced for different sparsity levels by sequential least squares thresholding, both failing to recover the true structure. 69

3.17 A matrix produced by the LASSO algorithm for various values of λ. 69

3.18 The eigenvalues of the true diffusion dynamics, along with their multiplicity. Also shown are the found eigenvalues of the best L2 fit and a rank-truncated fit. 69

3.19 When the adjacency matrix is known and enforced, then simple L2 fitting will produce the correct signs and weights 69

4.1 A wave of suppression on the stretch receptors produces an omega turn. The percent suppressed is shown, which travels along lasting approximately 5 seconds. The snapshots on the right are at the same times as the bold cross sections of the figure on the left. 70

4.2 Asymmetries needed to produce backwards and forward motion. a) The nueromuscular junctions (NMJs) decrease in strength as you travel posteriorly (anteriorly) along the body for forward (backward) motion and B- (A-) class neurons. The head (tail) is weakened in the original model in order to produce straight forward motion, and there is recent experimental evidence that the head circuit is fine tuned in a similar way [258]. b) Partially to compensate for the decrease in NMJ strength, the stretch receptor sensitivity is increased as you travel posteriorly (anteriorly) along the body for forward (backward) motion. 71

4.3 Average Center Of Mass velocity for regular forward motion as a function of the length of the stretch receptors, measured in body segments. 72

4.4 Angle change as a function of various body parameters. Displayed is the mean and standard deviation when the simulation is run for individuals with up to 10% variation in these parameters: D_*= damping coefficient; k_*= spring constant; $*_{PE}$= horizontal (passive) element; $*_{DE}$= diagonal element; L_{SEG}= segment length; T_{MSC}= muscle time constant. 74

4.5 2d PCA projections of a) data, b) an uncontrolled "null" model, c) a "supervised" model using expert-determined control signals, and d) an "unsupervised" model that uses control signals learned via algorithm 3. The governing equations matrices in are all learned from data, either uncontrolled (b, Equation 2.3) or controlled (c-d, Eq. 2.12). These data are color-coded by state: Black for Unknown, Yellow for Reversal neurons, Green for Forward, and Light (Dark) Blue for Dorsal (Ventral) turns. e-f) Example neuron datasets with reconstructions from the supervised model. A reversal-active (AVAL) and a Dorsal-Turn-active neuron (SMDDL) are shown. g-h) The same neurons shown with reconstructions from the unsupervised model. i-k) Correlations across datasets between data and reconstructions, split up into 4 different neuron groupings for interpretability. i) Baseline null models. The left-hand side is a straight line fit to a neural trace, i.e. how well pure drift can explain the signal. The right-hand side corresponds to the uncontrolled model in panel (b), and is generally worse than a straight-line fit. j) Full models with either expert/supervised control signals, (c) above, or learned/unsupervised control signals, (d) above. For each neuron grouping the expert signals produce significantly better fits. k) Partial supervised models, as more signals are added. For the Reversal (left-hand side) set, a "baseline" of a straight-line fit is subtracted. Shown are additive improvements, i.e. how much better each partial model is than the one immediately to the left. 80

4.6 Control signals can be learned from data via algorithm 3. a-d) The onset of well-known states as determined by experts (above) and as learned (below). All signals are normalized to have a maximum of 1.0. e) Correlation between expert and learned signals across 15 individual datasets. Reversals (Rev), Dorsal (DT) and Ventral Turns (VT) are consistently learned, but Forward state (Fwd) onsets are never significant. 81

4.7 a-b) Control signal reconstructions via linear encoding on the data including time delays, with all neurons (a) or 4 neurons removed (b). The removed neurons are: SMDDR, 81, SMDVL, and SMDVR as shown on the x axis of (c). Event detection is determined via a hard threshold for each signal (dotted line). See supplement for more discussion of this threshold. c) Neurons are eliminated in order of the largest magnitude given to them by the linear model. The number of false detections increases significantly only after 8 neurons have been removed. d-e) The weights given to the top 10 most important neurons for different iterations. 82

A.1 Increasing sparsity of an example well-reconstructed signal, onset of Reversal. 120

A.2 The entire time series of a well-reconstructed signal, onset of Reversal. . . . 121

A.3 Increasing sparsity of an example well-reconstructed signal, onset of Ventral Turn. There are more spurious spikes identified in this signal as compared to the Reversal signal. 122

A.4 The entire time series of a well-reconstructed signal, onset of Ventral Turn. 123

A.5 A signal that does not correspond to any known behavior and is of low quality. The neuron actuated most strongly by this signal does not have a name, and thus cannot be compared to literature studies or other datasets. 124

A.6 A signal that does not correspond to any known behavior and is of low quality. The neuron actuated most strongly by this signal is identified as 'RIML' and is known to be important in reversals. However, this signal is almost entirely noise and cannot be interpreted as real. 125

A.7 a. A heatmap of the autocorrelations of all signals across iterations. Each of the ten signals has a different maximum correlation, and the top two high quality signals (which actuate SMDVR and AVAL/R) are shown in more detail in other figures. b. As the free parameter 'r' (number of control signals) is increased, identifiable signals appear. Most have a similar quality (maximum autocorrelation) across different ranks. 130

A.8 The elimination path of the Ventral turn signal for one individual dataset. . 131

A.9 a) The encoding variables when all neurons are allowed in the reconstruction. The well-known neuron Left/Right neuron pair, SMDVL/R, are highly predictive. Additionally, other unidentified neurons appear and may be promising for future study. b) The reconstruction across the entire time series using the variables selected above. 132
A.10 a) The encoding variables when some neurons are removed from the reconstruction. Additional neurons appear. b) The reconstruction across the entire time series using the variables selected above. 133
A.11 a) The encoding variables when some neurons are removed from the reconstruction. Additional neurons appear. b) The reconstruction across the entire time series using the variables selected above. 134
A.12 The elimination path of the Reversal signal for one individual dataset. Several neuron pairs are well-known to be associated with this behavior, particularly AIBL/R, AVAL/R, AVEL/R, and RIML/R. These neurons do not show up as much in the elimination path because their activity only increases significantly after the onset of the reversal behavior, which is much slower than the turns. 135
A.13 a) The encoding variables when all neurons are allowed in the reconstruction. Many well-known neurons appear, as well as some unidentified ones. b) The reconstruction across the entire time series using the variables selected above. 136
A.14 a) The encoding variables when some neurons are removed from the reconstruction. Additional neurons appear. b) The reconstruction across the entire time series using the variables selected above. 137
A.15 a) The encoding variables when some neurons are removed from the reconstruction. Additional neurons appear. b) The reconstruction across the entire time series using the variables selected above. 138
A.16 PCA projections for individual 1. 139
A.17 PCA projections for individual 2. 140
A.18 PCA projections for individual 3. 141
A.19 PCA projections for individual 4. 142

GLOSSARY

DMD: Dynamic Mode Decomposition, a data-driven method for regressing to linear dynamics on a low-dimensional manifold.

SINDY: Sparse Identification of Nonlinear Dynamics, a complimentary data-driven method for regressing to sparse non-linear dynamics within a given library of candidates.

DMDC: Dynamic Mode Decomposition with control, DMD that incorporates a control signal.

SINDYC: Sparse Identification of Nonlinear Dynamics with control, SINDy that incorporates a control signal

LASSO: An algorithm that uses a common regularizer in order to promote sparsity in an optimization problem

C. ELEGANS: A soil-dwelling nematode, about 1mm in length, that is a major model organism in biology. Importantly, it is transparent and has non-spiking neurons.

CALCIUM IMAGING: A technique that involves changing gene expression in neurons so that when the Calcium they release fluoresces under application of low-level laser stimulation. Note that Calcium is released when neurons increase their membrane voltage, and is thus a crude measure of the neuron's activity level.

CONNECTOME: The entire network of neurons and synaptic connections for a real brain. The *C. elegans* connectome has been almost entirely mapped since 1986 [294]

OMEGA TURN: A deep turning behavior performed by *C. elegans* in order to reorient, so-called because the body makes the shape of an Ω.

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to the interconnected network of people who have made this possible. Personally, my wife Ariana who has been supportive through all the ups and downs of research life, as well as someone I admire and am made better by learning from. Professionally, my committee members all deserve special thanks: Nathan, for being a wonderfully supportive boss and mentor; Eric for introducing me to Nathan in the first place and for energetically tackling the science that comes his way; Deep for hosting me as an REU student and for graciously helping me find my way as an early graduate student, even though my path lead out of his lab; and Paula and Suzanne for being inspiring teachers, researchers, and helpers of an early, confused graduate student!

DEDICATION

to my wonderful and supportive wife, Ariana

Chapter 1
INTRODUCTION

This thesis has two main components: first, several recently developed data-driven algorithms are extended and newly developed, and second, the methods are used on data from the *C. elegans* model organism, often affectionately referred to as "the worm". In the following sections, the required background knowledge for each of these components is introduced.

1.1 Why Data-Driven?

There are two broad approaches to modeling emergent or complex phenomena in a natural system: first-principles or data-driven modeling. The first-principles approach seeks to understand the system using small-scall "known" mechanics as a starting point. For example, in the *C. elegans* community, many groups seek to build biomechanical models using rod-spring or other approximations of the biophysical forces the body experiences [30, 38, 37, 51, 70, 138, 137, 140, 139, 273]. This strategy requires a large number of small-scale parameters to be known at least approximately, and generally proposes putative mechanisms for an observed large-scale behavior. More specifically, this strategy generally produces answers of the form: "if this mechanism were present, these would be the effects/limits/requirements."

On the other hand, data-driven modeling strategies have a different, and unsurprising starting point: data. That is, a particular microscopic mechanism for how changes in a system happen are not proposed, but rather the relationships between measured quantities are characterized. This strategy generally produces answers like: "These statistics describe a baseline of activity, from which deviations can be measured" and/or "The variation in activity

can be explained by a small number of degrees of freedom or hidden states." This last point is connected to a theme of this thesis, and something that has generated much excitement in the computational neuroscience field: observations that the "true" dimensionality and degrees of freedom of a population of neurons are much lower than the full number of individual cells. That is, their activity produces a "low dimensional manifold," showing a way forward for large-scale understanding of the brain. These methods have become very popular recently due to a suite of technologies that allow real-time recording from large numbers of neurons, which will be discussed in more detail in following sections.

This thesis is composed of several projects, spanning both modeling strategies. The chronological trajectory is one of beginning in a biomechanical modeling space in order to gain domain knowledge, and moving to data driven methods because of the many exciting opportunities that are opening up.

1.2 Low Dimensionality

1.2.1 Space

Given a dataset with many channels, for example a continuous measure of neuron activity, how can one understand what is going on? A priori, each of these neurons might do computations of unknown complexity and might be largely independent of each other. Alternatively, entire populations of neurons could coordinate their activity in order to perform common tasks. The second appears to be true across a wide variety of model organisms and environments.

This has lead to a general principle for analyzing neural data: try and understand the underlying dimensionality of the data, using one of a various number of "dimensionality reduction" techniques [152]. There are a multiplicity of techniques that quantify this dimensionality, and most also discover a set of "basis functions" that span this new space, which may or may not be simply related to the original dimensions. By far the most popular of these spatial dimensionality reduction techniques is Principle Component Analysis (PCA)

[284], a covariance-based method.

PCA is attractive for many reasons, including a) it can be applied to any dataset; b) it is computationally very efficient; c) it produces both a basis in "space" (i.e. across recorded channels) and modes in time; d) these modes are clearly interpretable and ranked in order of "importance" (variance explained); and e) it often works in practice. Many datasets have been analyzed using PCA and variants [152], although it is not without its criticisms. In particular, as PCA is based purely on covariance within a time slice, it is not aware of dynamics. That is, if the data time slices are shuffled, then the algorithm will give the exact same modes in space, regardless of how strange or uninterpretable the attached time series are.

1.2.2 Time

At this point, we must consider an older and even more popular technique, the Fourier Series decomposition. This is a dimensionality reduction in time, as it takes a very large number of measurements and expresses them as a function of a small number of, hopefully interprettable, oscillations. Indeed, Fourier Series and power spectra are commonly used in many neuroscience applications, like EEG [262, 288].

However, Fourier series do not apply a spatial dimensionality reduction. Thus, although they simplify a single time series, they do not reduce the number of units under study. Enter the recent dimensionality reduction method that combines the strengths of both of the above methods, Dynamic Mode Decomposition (DMD).

1.2.3 DMD

There are several different perspectives from which one can view this method, and it will be explained more mathematical detail in the Background section below. It has excited many theorists due to its mathematical connections to Koopman theory [45, 64, 124, 158, 213, 270, 295, 22, 248]. However, in this manuscript, two different perspectives are more emphasized. First, a time and space dimensionality reduction can be accomplished simultaneously.

Second, the reduction produces a model of the dynamics on this low dimensional manifold.

The first point above describes one of the two main mathematical formulations of DMD:

$$\mathbf{X} = \sum_{j}^{N} b_j \exp i\omega_j t \tag{1.1}$$

where $\mathbf{X}$ is the data matrix, b_j are the spatial modes akin to the PCA spatial basis modes, and ω_j are the complex temporal frequencies. In words, each term in this sum describes a coherent spatial structure (b_j) that has a well defined oscillatory component (real part of ω_j) and an envelope that is either exponentially decaying, growing, or constant (imaginary part of ω_j). This perspective grew out of the fluid dynamics community [248], and has been used to characterize coherent structures in multi-million dimensional hydrodynamic simulations.

DMD also produces a dynamical systems model, of the form:

$$\dot{\mathbf{x_k}} = \mathbf{A}\mathbf{x_k} \tag{1.2}$$

where $\mathbf{x_k}$ is a single time slice of the data ($n \times 1$), $\dot{\mathbf{x_k}}$ is the time derivative, and $\mathbf{A}$ ($n \times n$) describes the flow field at a single point. Of course, Equation 1.2 describes a system of linear ordinary differential equations whose solution is given by Equation 2.5, thus the equivalence of the perspectives. However, this dynamical systems perspective allows for the easy addition of terms to describe the evolution of the dynamics, which, under certain circumstance, will not destroy the dimensionality reduction benefits of the original algorithm. In particular, the addition of "control signals" is a well developed and interpretable to extend this framework.

This forms the starting point for the novel data-driven methods that will be introduced in later sections, as well as the applications to biological data.

1.3 Control Theory

1.3.1 Control in Engineering

Control theory starts from a simple mathematical framework:

$$\dot{\mathbf{x_k}} = \mathbf{A}\mathbf{x_k} + \mathbf{B}\mathbf{u_k} \tag{1.3}$$

where $\mathbf{u_k}$ is the $r \times 1$ external input called the "control signal" and $\mathbf{B}$ is the $n \times r$ maps the control signal to the phase space of the intrinsic dynamics. This equation also describes the formalism of Dynamic Mode Decomposition with Control (DMDc), which discovers both $\mathbf{A}$ and $\mathbf{B}$ from data. Note that many formalisms also explicitly model a measurement function [291, 121], because it is often the case that measured quantities can be noisy or otherwise different from the quantities undergoing simple dynamics. We do not focus on that aspect of control theory in this manuscript.

There are two fundamentally different types of control: closed loop, where $\mathbf{u} = f(\mathbf{x})$, and open loop, where $\mathbf{u} = f(t)$. A common use case for closed loop feedback is to stabilize an unstable fixed point. A classic example is that of an inverted pendulum (equivalently, balancing a broomstick) [17]; the upside-down position quickly collapses without intervention. However, such a state can be balanced via using a "controller" that, broadly speaking, tries to push the state back near the unstable fixed point as it attempts to fall off. Alternatively, this type of control can be used to stabilize a trajectory [17].

This paper instead focuses on the open loop feedback scenario, in which control signals are specified externally. This is often used in partnership with offline optimization [309], in which an "ideal" input is calculated before the system begins to evolve using an accurate model, and then it is applied during simulation. A separate scenario is that of perturbations to a system that are not a function of the internal state. Effectively, this perturbation scenario creates a boundary for "the system" under study, dividing the model-able world into the intrinsic dynamics of the system, denoted by the matrix $\mathbf{A}$, and something external to that, the control signal $\mathbf{u}$. This interpretation forms a backbone of a framework to apply control theory to biological systems.

1.3.2 Control in biology

Many biological systems display a curious duality, which can be easily seen in *C. elegans*: 1) Long-lived behaviors which are relatively stable to changes in environment, like oscillatory body bends that comprise forward crawling; 2) System-wide transitions that can

often happen both spontaneously or due to outside stimuli, like an escape response when touched. Although it is largely outside the scope of this work, it is worth noting that the first phenomenon is akin to closed-loop feedback control. That is, even in a wide variety of environments and possibly resistive forces, a highly specific coordination of every single muscle in the body robustly produced by the system.

This manuscript concerns the second phenomenon, and uses both first-principles and data-driven modeling methodologies to explore it. For example, as part of an escape response in *C. elegans* to a touch stimulus, deep "Omega turns" are performed (so called because of the shape that the body makes). It has been a challenge for biomechanical models that can capture forward swimming motion to also produce such a behavior [70], and it has been unclear what sort of mechanism would produce such a transient change in behavior. The first result of this thesis reports on a model that does just that, which proposes a mechanism of open loop control. This perspective considers the forward-motion-producing elements of the animal to be the "intrinsic" dynamics, and the turn to be produced by a subsystem eternal to those dynamics though still within the animal itself.

This perspective is further extended to data recorded from neurons, taking transitions between distinct states to be similar "internally generated control signals" and the rest of the dynamics to be explained by the intrinsic dynamics of the neuronal network. However, in order to fully model this situation, the established methods required extensions.

1.3.3 An Open Problem: Learning Control Signals

In classic open-loop control theory [309], including DMDc, the control signals must be known ahead of time. This is often acheivable in an engineering context, as the control signals describe input from the scientist or engineer. However, in a biological context they are often produced by an unknown mechanism, and must be discovered along with the dynamics themselves. Attempts to solve this problem form the bulk of the mathematical methods discussed in this thesis.

Data-driven methods for system identification (system ID) are of growing interest across

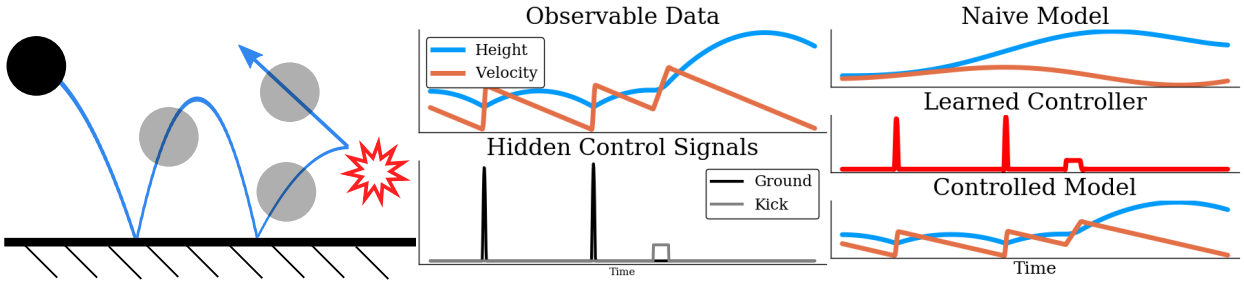

Figure 1.1: a) A simple example system: a bouncing ball. The intrinsic dynamics is the acceleration due to gravity, which is linear in the velocity-acceleration basis. The external spatially-dependent forcing is provided by the ground, and in this example there is an additional time-dependent forcing, e.g. a kick. b) The observed data in this simple case have obvious discontinuities, and the two types are easy to distinguish. c) The control signal provided by the ground, which is actually a function of space, and the external kick, which is purely a function of time.

the physical, biological and engineering sciences. In addition to the discovery of interpretable models and/or governing equations, the primarily goal is to identify control laws and/or protocols that can be used for understanding how to actuate a given dynamical system. Recent data-driven system ID methods include the *dynamic mode decomposition* (DMD) [248, 171], which provides a least-square regression to a best-fit linear model $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, and the *sparse identification of nonlinear dynamics* (SINDY) [48], which provides a sparse regression to a parsimonious nonlinear dynamical system $\dot{\mathbf{x}} = f(\mathbf{x})$. Both DMD and SINDy have counterparts, DMDc [227] where $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ and SINDYc [49] where $\dot{\mathbf{x}} = f(\mathbf{x}) + g(t)$ respectively, that explicitly disambiguate between the dynamics $\mathbf{x}$ and actuation $g(t) = \mathbf{u}$ (control signal) in the system. Such disambiguation relies on knowing the control signal $\mathbf{u}(t)$, and current methods are unable to effectively perform system ID without such knowledge. We propose a data-driven framework that leverages a Bayesian framework for learning sparsely active control signals from data while simultaneous extracting interpretable model of the intrinsic dynamics. Thus, we extend the data-driven DMDc and SINDYc frameworks by allowing for accurate system ID without knowledge of the control signal.

In the absence of actuation, learning dynamical models from data alone is amenable to a

diverse set of mathematical methods. DMD is attractive in that it produces a best-fit linear model whose solutions are simple exponentials. The original DMD algorithm [248] was proposed by Schmid for approximating autonomous, low-dimensional spatio-temporal dynamics in high-dimensional fluids. It was subsequently used in a wide variety of application areas including computer vision [117, 93], neuroscience [42], disease modeling [225], finance [203], and fluid dynamics [23, 248, 250, 252]. Accounting for control allowed for both a drastically lower number of relevant dimensions and increased accuracy of the recovered autonomous dynamics [227]. Nonlinear models can be constructed through regression techniques on libraries of potential nonlinear dynamic terms, including sparsity promoting techniques [49]. Such parsimonious methods have shown to be successful on spatio-temporal [241], parametric [239], networked [199], control [49], and multiscale [58] systems. There are numerous alternative approaches to fitting the data with models, including non-sparsity promoting regressions to polynomial and/or special function bases [302, 260], as well as symbolic regression to identify directly the structure of a nonlinear dynamical system from data [35, 254, 73]. Similar benefits were realized in adding control to nonlinear systems [49]. However, as with DMDc these control signals must be known in advance, which is often not the case for natural systems.

An alternate method for learning discrepancies

One approach for simultaneously learning a model and unknown external forcing is the *discrepancy modeling* framework [52, 154, 279]. This fully Bayesian approach posits a model where the actuation $g(t)$ is treated as a *discrepancy* and is generally modeled as a Gaussian process. This and similar frameworks have been applied in many different real-world settings, including ecology [11], robotics [160], and control [54]. Gaussian processes are very powerful in that they can model nearly any smooth function, but this contributes to a major difficulty with this framework: identifiability [6, 299]. That is, unless you have many different data sets [9] or have more knowledge of the exact functional form of the discrepancy [147], there is no clear way to separate out what is the external signal and what is the intrinsic dynamics.

1.3.4 How to learn control signals

We propose a partially Bayesian framework for learning sparsely active control signals purely from data simultaneous with an interpretable model of the intrinsic dynamics, allowing for accurate reconstructions both of the underlying autonomous system and the effects of the external signals. We do this by building the posterior distribution of a set of uncontrolled models, $f(\mathbf{x})$, either linear (DMD) or nonlinear (SINDy), as described in Section 2 (Background Methods). Sampling from the posterior and comparing to data produces an initial guess for dynamics that are outside of the initial model structure, which forms a basis for approximating $g(t)$, as detailed in Section 3. We test our method on linear, nonlinear, and chaotic systems with external control signals, successfully learning the intrinsic simple system as well as the control signals. We demonstrate our method on a number of example problems where we show that both the control signal and dynamics can be extracted from data alone, as described in Section 4 (Results). The code for this framework is freely available on GitHub in the programming language Julia, heavily using advanced features from the DifferentialEquations [231] and the Turing probabalistic programming packages [109].

1.4 C. elegans *as a model organism*

The nematode *Caenorhabditis elegans* (*C. elegans*) is an ideal model organism for probing the relationship between structure and function in neuronal networks as it is comprised of only 302 sensory, motor, and inter-neurons whose stereotyped synaptic connections (i.e. its connectome) are known from serial section electron microscopy [294, 72, 141]. This stereotyped connectivity has allowed the assignment of a unique name to each neuron in the animal. Through painstaking study, many of the individual neurons that are required for important functions have been identified. For example, ablation of a left/right pair of neurons denoted AVBL/R prevents the individual from moving forwards, while the ablation of the AVAL/R pair completely disrupts backwards motion [294]. However, such studies have not been able to uncover the role of many other neurons, whose ablation or mutation

do not lead to large-scale phenotypic changes although there are databases cataloging such changes [304]. In light of this, whole-network modeling strategies are seen as a promising avenue for understanding the animal more completely

Fortunately, *C. elegans* is perhaps the simplest organism to display many of the hallmark features of high-dimensional networked biological systems, including the manifestation of low-dimensional patterns of activity associated with functional behavioral responses [152]. Thus the nervous system must reduce the high-dimensional representation of environmental stimuli into much lower dimensional representations of motor commands [236, 186, 171, 168, 99, 152, 153]. Low dimensional representations have been separately considered in posture (behavioral) analysis [266, 265] as well as in previous analysis of calcium imaging data [152, 214]. These representations can be used to characterize the evolution of both postures [265] and neuron population dynamics [40, 185]. Efforts to characterize the dynamics on this manifold have used piecewise linear models to describe the entire state space, but it is unknown how a single, global dynamical model can generate the observed dynamics.

1.4.1 Whole-brain imaging

Although *C. elegans* has been studied for decades, the field has recently undergone a revolution due to several new genetic technologies, most notably Calcium imaging [152]. This technology consists of several steps: 1) Mutate the individuals such that they express fluorescent markers in their neurons; 2) Characterize how the markers are expressed as a function of the underlying quantity of interest, for example neuron membrane voltage potential; and 3) Build a hardware and software setup for imaging and tracking. The output of this technology is traces of neural activity in real time throughout a large portion of the brain; in *C. elegans* the majority of active neurons are captured.

Most experimental setups use immobilized animals to collect data, which makes the tracking problem significantly easier. However, it also opens up an important potential criticism: any lessons from modeling efforts on immobilized data may not be generalizable to freely behaving animals. As this thesis exclusively uses data from immobilized animals,

a strong focus will be on the overall mathematical framework, which can be easily rerun on new datasets.

Chapter 2

BACKGROUND

2.1 Data-driven methods

2.1.1 Principle Component Analysis (PCA)

PCA is a very popular technique for reducing the spatial dimension of a signal [259]. As with several of the methods discussed in this thesis, there are multiple mathematically equivalent yet conceptually different perspectives from which to view this algorithm. Two shall be described here, both requiring mean-subtracted data: 1) PCA is a linear auto-encoder; and 2) PCA diagonalizes the covariance matrix of the data.

PCA as an autoencoder

The first view has connections to several popular autoencoder methods in machine learning, and can be framed as an optimization problem [77]. That is, PCA seeks to find the best linear transformation matrix $\mathbf{M^T}$ such that:

$$\mathbf{Y} = \mathbf{M^T X} \tag{2.1}$$

where $\mathbf{X}$ is the $d \times n$ data matrix and $\mathbf{Y}$ is the $r \times n$ data in the new basis. In addition, the columns of $\mathbf{M}$ are constrained to be orthogonal. The new dimensionality $r \leq d$ is a hyper-parameter which must be chosen by the user, although automated methods like cross validation can be used [162].

PCA as a diagonalization

Fig. 2.1 shows a graphical explanation of the second perspective on PCA: the modes capture the largest possible variance of all orthogonal directions, ranked. Mathematically, this means

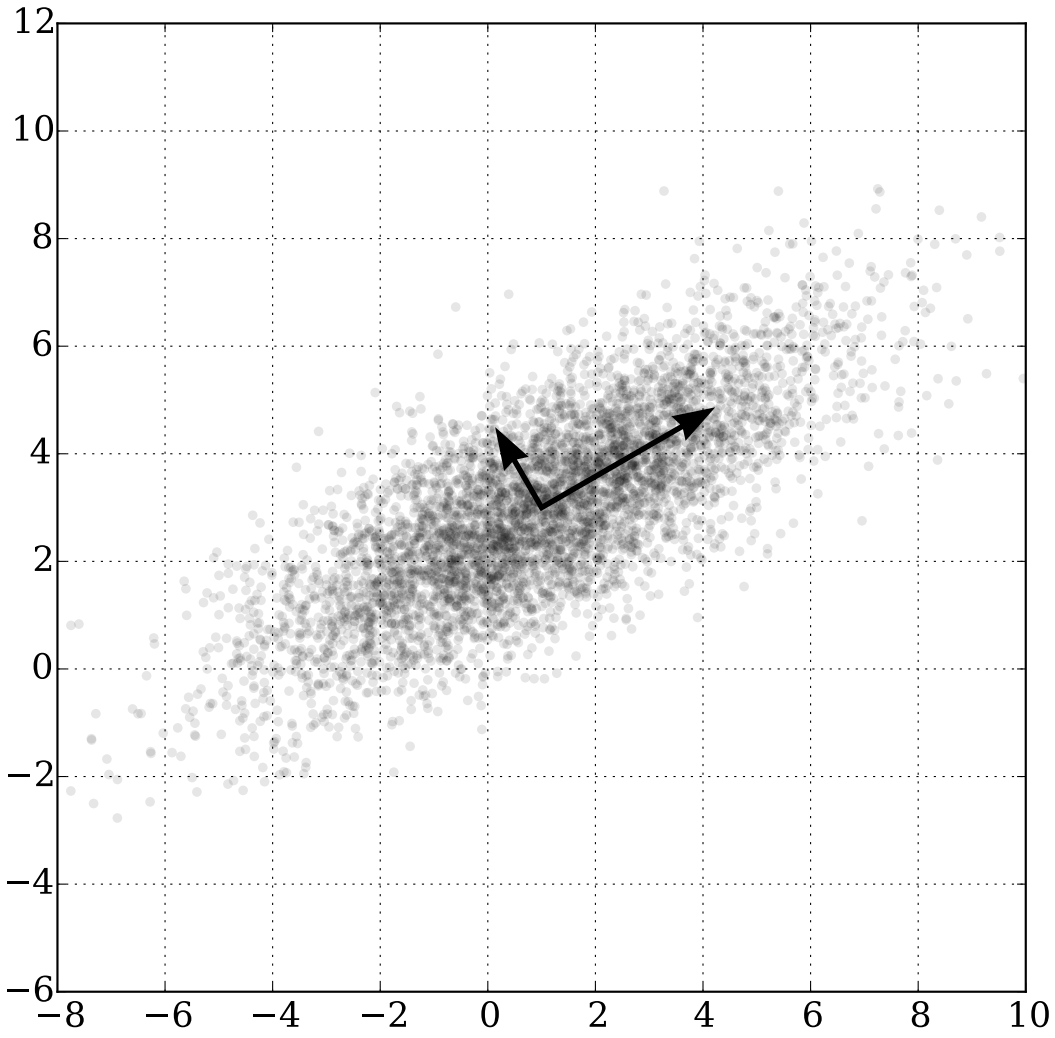

Figure 2.1: Figure from [1]. PCA returns orthogonal vectors that capture the largest variance, in order.

that the first mode satisfies:

$$\mathbf{w_1} = argmax_{\|\|w\|\|=1} \|\|\mathbf{Xw}\|\|^2 = argmax_{\|\|w\|\|=1} \|\|\mathbf{w^T X^T Xw}\|\| \tag{2.2}$$

This is then equivalent to the Rayleigh Quotient [218], and the maximal solution is of the largest eigenvalue of $\mathbf{X^T X}$ and occurs when $\mathbf{w}$ is the corresponding eigenvector. But of course, the quantity for mean-subtracted data $\mathbf{X^T X}$ is just the covariance! And so, the eigenvectors of the covariance matrix are the PCA modes, and the corresponding eigenvalues are the amount of variance captured.

2.1.2 DMD

DMD provides a linear model for data matrices constructed using temporal snapshots of the state space, $\mathbf{X}_1 = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_{m-1}]$ and $\mathbf{X}_2 = [\mathbf{x}_2\ \mathbf{x}_3\ \ldots\ \mathbf{x}_m]$ where $\mathbf{x}_j = \mathbf{x}(t_j)$. Specifically, it finds the best fit linear dynamical system

$$\mathbf{X}_2 = \mathbf{A}\mathbf{X}_1 \tag{2.3}$$

passing through the m snapshots of the statespace. There are a number of variants for computing $\mathbf{A}$ [248, 282, 144, 79, 125, 13, 20], with the *exact DMD* [282] simply constructing

$$\mathbf{A} = \mathbf{X}_2\mathbf{X}_1^\dagger \tag{2.4}$$

where † denotes the Moore-Penrose pseudo-inverse, which is a least-squares fitting procedure. However, in practice due to the size of matrix $\mathbf{A}$ in (2.4), the data is first projected onto the dominant correlated modes via the singular value decomposition before an eigen-decomposition is computed [171], i.e. a low-rank approximation is first computed. The many variants of the DMD algorithm can be easily accessed with the package PyDMD [83]. The reader is encouraged to download this package to view the diversity of algorithms for approximating $\mathbf{A}$. Optimized DMD improves on a number of innovations for DMD by debiasing the regression [13].

Many of these formulations do not actually compute the matrix $\mathbf{A}$, but instead calculate some number of eigenvalues and eigenvectors, giving the data the reconstructed form:

$$\mathbf{X} = \sum_{j}^{N} b_j \exp i\omega_j t \tag{2.5}$$

where $\mathbf{X}$ is the data matrix, b_j are the spatial modes akin to the PCA spatial basis modes, and ω_j are the complex temporal frequencies. In words, each term in this sum describes a coherent spatial structure (b_j) that has a well defined oscillatory component (real part of ω_j) and an envelope that is either exponentially decaying, growing, or constant (imaginary part of ω_j).

2.1.3 SINDy

The SINDy algorithms have a parallel structure to the DMD equations above:

$$\dot{\mathbf{X}} = \mathbf{\Phi}(\mathbf{X})\mathbf{\Xi} \tag{2.6}$$

And the formulation with control is similar:

$$\dot{\mathbf{X}} = \mathbf{\Phi}(\mathbf{X})\mathbf{\Xi} + \mathbf{BU} \tag{2.7}$$

where $\mathbf{\Phi}(\mathbf{X})$ is a library of nonlinear functions of the original data rows and $\mathbf{\Xi}$ is the sparse matrix of coefficients. Sparsity is enforced via an L1 norm or sequential least squares thresholding [48, 49], and significant parameters are determined via information theory metrics like AIC, as described in previous work [201].

2.1.4 Sparse optimization

Imposing sparsity on a solution is done for many reasons: increase interpretability of the few nonzero elements [311]; recover solutions that are otherwise ill-defined [55]; and/or decrease overfitting [311]. A non-sparsified form of the SINDy equation written as an optimization problem is:

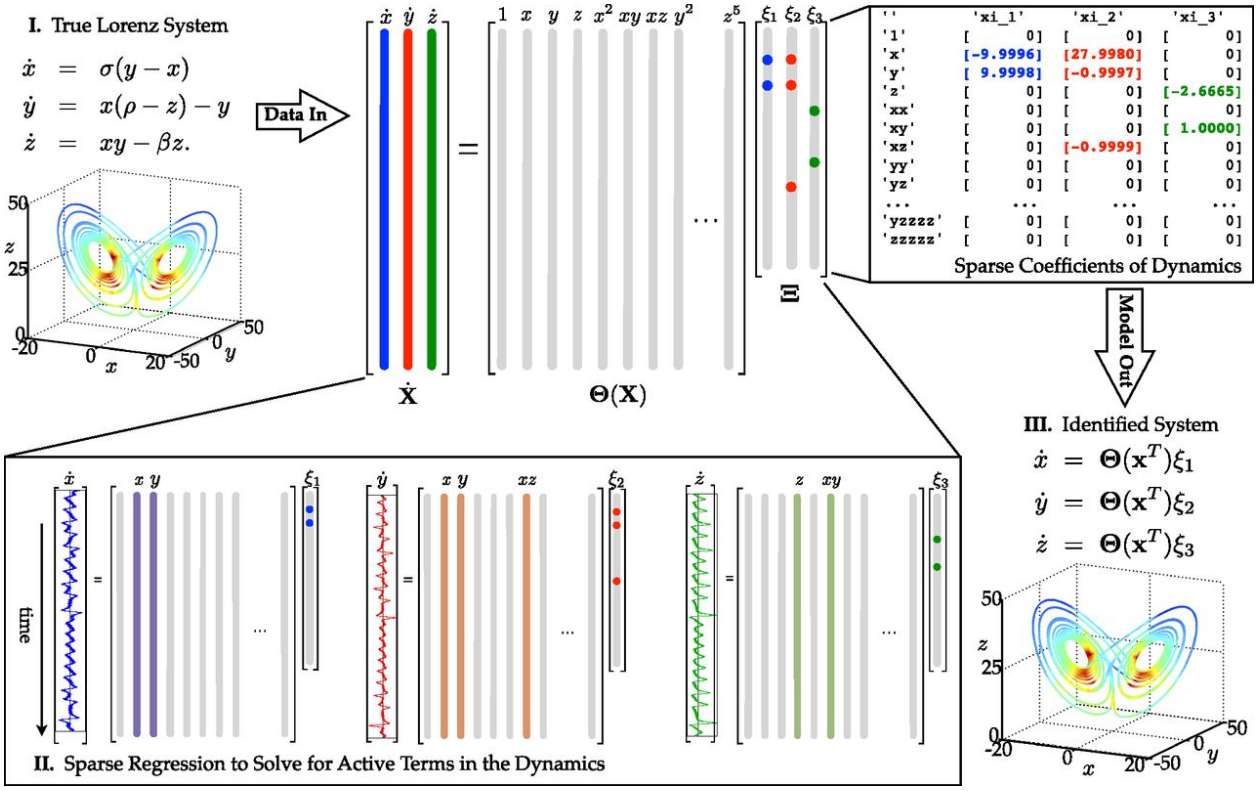

Figure 2.2: Figure reproduced from [47]. SINDy produces a set of governing equations (ODEs) from data and a hand-picked library of terms.

$$\min_{\boldsymbol{\Xi}} \|\|\dot{\mathbf{X}} - \boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Xi}\|\|_F \tag{2.8}$$

Where ℓ_F refers to the Frobenius norm, i.e. a sum of the squares of each individual matrix entry, analagous to the ℓ_2 norm for vectors. Solving this problem gives a dense solution. Pure sparsity is enforced via the ℓ_0 norm, which is a simple binary indicator of whether the entry is nonzero:

$$\min_{\boldsymbol{\Xi}} \|\|\dot{\mathbf{X}} - \boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Xi}\|\|_F + \lambda\|\|\boldsymbol{\Xi}\|\|_0 \tag{2.9}$$

However, this is an NP-hard, non-convex problem [55]! Fortunately, there is a convex relaxation of the problem which is much easier to solve [89]:

$$\min_{\boldsymbol{\Xi}} \|\|\dot{\mathbf{X}} - \boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Xi}\|\|_F + \lambda\|\|\boldsymbol{\Xi}\|\|_1 \tag{2.10}$$

where λ is a hyperparameter that is often chosen through cross validation. The only difference is that the regularization term (second term) has been changed from an ℓ_0 to an ℓ_1 norm. This trick is often used in machine learning [123], and has a special name: LASSO.

Instead of using the ℓ_1 norm, it is also possible to cast this as a greedy thresholding algorithm, also called Sequential Least Squares Thresholding. This algorithm is described in Alg.1, and has a free parameter, γ, related to λ in Eq. 2.10 that sets the threshold level. There are many variations of this thresholding algorithm that, for example, adaptively set the threshold [34].

Algorithm 1 Sequential Thresholding

procedure SEQUENTIALTHRESHOLDING($\gamma, MaxIter$)
    $zeroMask = initializeZeros()$
    for $i \leftarrow 1, MaxIter$ **do**
        $\boldsymbol{\Xi} = SolveConstrainedL2(\dot{\mathbf{X}} * inv(\boldsymbol{\Phi}(\mathbf{X}), zeroMask)$
        $zeroMask = setMask(\boldsymbol{\Xi}, \gamma)$
    end for
end procedure

There are also other, more complex, algorithms for solving the sparse optimization problem that sometimes have better convergence properties, but that are not the focus of this thesis [308].

Caveats with sparse optimization

Although under some conditions the ℓ_1 solution produces the optimal sparse solution [89], it can also make mistakes and remove important variables too early [269]. In some cases, it has been shown that a properly parametrized thresholding algorithm is a more accurate approximation of the ℓ_0 norm solution [305].

2.1.5 Interpretation: Lasso/thresholding as a variable selector

If internally generated control signals are present, then there are two possibilities: they are random and fundamentally unpredictable, or they are encoded in the network. We explore the degree to which these signals are encoded using sparse variable selection algorithms and *time-delay embedding,* (Eq. 2.1.7) where data from further in the past is utilized. Mathemaically, this is written as:

$$\mathbf{U} = \mathbf{K}_1\mathbf{X}_1 + \mathbf{K}_2\mathbf{X}_2 + ... \tag{2.11}$$

There are multiple methods that are often used to perform this variable selection task [281]. However, these methods may make mistakes in their selections [269], and in general it is unclear how unique the selection is. The behaviors of *C. elegans* have been well studied, and each onset is associated with well-known neurons. Variable selection methods will almost certainly discover these well-known neurons, but by exploring further in the "elimination path", less obvious encodings can be discovered. Algorithmically, this is the sequential removal or ablation of the most important neuron for all time delays, and then a re-fitting of the sparse model. If the quality of the reconstruction does not degrade along the elimination path, the signal ($\mathbf{U}$) must be distributed throughout the data ($\mathbf{X}$). The quality of signal reconstruction is defined here as the number of false positives and false negatives in the reconstructed signal. Event detection is defined as a minimum number of frames above a hard threshold, as shown in Fig. 4.7 and discussed in the supplement.

2.1.6 DMDc and SINDYc

DMDc [227] leverages the advantages of DMD and provides the additional innovation of being able to disambiguate between the underlying dynamics and the effects of a known actuation signal. For a control input matrix $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ \dots\ \mathbf{u}_{m-1}]$ where $\mathbf{u}_j = \mathbf{u}(t_j)$ is the actuation signal at time t_j, DMDc regresses instead to the linear control system

$$\mathbf{X}_2 = \mathbf{A}\mathbf{X}_1 + \mathbf{B}\mathbf{U}. \tag{2.12}$$

The DMDc method regresses to find the best matrices $\mathbf{A}$ and $\mathbf{B}$ in a least-squares sense given $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{U}$. If the matrix $\mathbf{B}$ is known, then DMDc needs only to find $\mathbf{A}$. Thus DMDc does not require knowledge of the underlying governing equations, only time snapshots of the state space and control input, making it compelling for systems whose governing equations are unknown. As with DMD, the DMDc algorithm capitalizes on underlying low-dimensional structure in the data by using the singular value decomposition to compute $\mathbf{A}$ and $\mathbf{B}$ in practice.

Note that DMDc uses only snapshots in time of the state space and control input, making

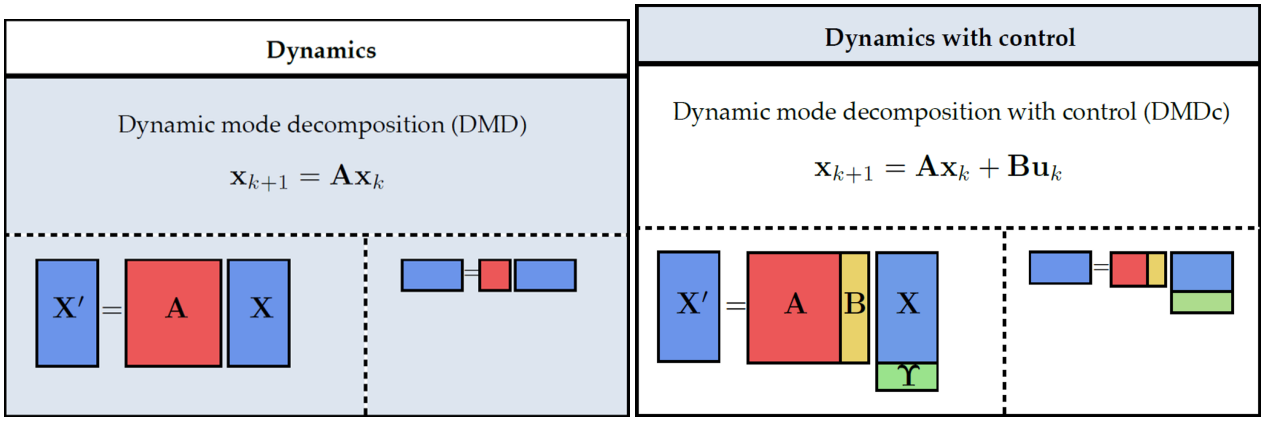

Figure 2.3: A graphical explanation of DMD and DMDc in discrete time.

it compelling for systems whose governing equations are unknown. The DMDc equation is graphically represented in Fig. 2.4. The governing matrices ($\mathbf{A}$ and $\mathbf{B}$) along with the control signal ($\mathbf{U}$) produce a predictive model, such that the state of the system far in the future can be predicted. For instance, the third time step can be estimated from the first via:

$$\mathbf{x}_3 = \mathbf{A}(\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{u}_1) + \mathbf{B}\mathbf{u}_2 \tag{2.13}$$

2.1.7 Time-delay embedding

A weakness of the DMD and SINDy methods we have introduced thus far is their single-step nature. That is, they produce a differential equation not by weighting effects across time but via the effect of one time-step on the next as in Eq. 1.3. Time-delay embedding is a technique for getting around this. Simply put, this technique stacks time series offset by increasing multiples of an offset Δt into a large matrix, and takes this as the new state. This is also known as the Hankel matrix, and with a set of n time slices (which are themselves data vectors, e.g. activity levels of neurons) $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ..., \mathbf{x_n}$, we can write this as:

$$X(d) = \begin{bmatrix} \mathbf{x_1} & \mathbf{x_2} & \dots & \mathbf{x_{n-i}} \\ \mathbf{x_2} & \mathbf{x_3} & \dots & \mathbf{x_{n-i+1}} \\ \vdots & & \ddots & \\ \mathbf{x_i} & \mathbf{x_{i+1}} & \dots & \mathbf{x_n} \end{bmatrix}$$

Note that the subscripts refer to times, and that each entry is a vector. In addition, this form is deeply related to the Takens Embedding Theorem. This theorem has two practical implications that connect real-world messy time series to simple models: Nonlinear dynamics can be better modeled by a linear dynamical system if using time-delay embedding [129, 44], and the effects of unobserved variables can be recovered [171].

2.1.8 Generating Test Data

The exact form of the dynamics matrix $\mathbf{A}$ depends on the basis of the measurements. In particular, the dimensionality of the underlying true dynamics $\mathbf{z}$ may be much smaller (or larger) than the state space $\mathbf{x}$ being measured. This is mathematically expressed by a measurement matrix, $\mathbf{H}$, of size $n \times r_0$ where n is the number of distinct measurements of the state space and r_0 is the dimensionality of the true state. Thus we find,

$$\mathbf{x}_j = \mathbf{H}\mathbf{z}_j + \mathbf{W}_j \tag{2.14}$$

where the components (w_k) of the vector $\mathbf{W}_j$ are a Gaussian noise parametrized by the variance σ such that $w_k = \mathcal{N}(0, \sigma)$. The level of measurement noise and the dimensionality mismatch between the measurements and the underlying dynamics can drastically change the effectiveness of the various DMD algorithms.

If control is added, then the true dimension of the system as a whole will be $r_0 = r_A + r_B$ where r_A is the rank of the matrix $\mathbf{A}$ and r_B is the rank of the matrix $\mathbf{B}$. Notationally, r without a subscript is the algorithmic rank truncation, which is typically different than r_0 in practice and difficult to accurately extract with noise and measurement mask $\mathbf{H}$.

When generating data for benchmarking the iDMD and iDMDc methods, a random matrix $\mathbf{A}$ was generated and then the eigenvalues were set to the unit circle for stable dynamics. This is consistent with previous comparison tests for the accuracy of DMD algorithms [79, 125, 13]. The true data is generated based on (2.3) or (2.12), for unactuated and actuated scenarios respectively. If $n > r_0$ then a random measurement matrix $\mathbf{H}$ is also generated, otherwise $\mathbf{H}$ is the identity. Finally, Gaussian white noise is added and the data is given by (2.14).

2.1.9 Learning Control signals

Discrepancy modeling

Many inverse problems, that is learning a model structure and parameter values from data, are ill-posed [279]. There are multiple possible sources, and a major one is systematic discrepancy between the model and data due to the data being produced by a process that does not conform to the assumptions of the model. Seminal work has demonstrated how direct modeling attempts can fail to recover accurate dynamics, but they can be recovered if a discrepancy term is added, as in Eq. 2.15 [52, 154, 279]. This is done by positing that $g(t)$ can be modeled by a Gaussian Process. Previous work also requires the form of the intrinsic dynamics $f(\mathbf{X})$ to be known, though parameters can be discovered:

$$\dot{\mathbf{X}} = f(\mathbf{X}) + g(t) \tag{2.15}$$

We take an opposite approach, positing sparsely active or spike-like control signals. In this view, although the intrinsic dynamics of the system could be modeled by Eqs. 2.6 or 2.3, a direct approach will not work because these external perturbations are unknown. A key assumption is that these external perturbations are sometimes weak or entirely absent, allowing a subsampling procedure somewhat similar to [299]. Our approach then does not posit an explicit function form for $g(t)$, but instead treats this control function as the statistically significant deviations between the intrinsic dynamics and the data, where statistically

significant refers to an explicit noise model made possible due to the Bayesian framework. This method is more fully explained in the next section and in algorithm 3.

2.1.10 Iteratively Reweighted Least Squares

There is a large body of work on the effects of outliers on regression problems. One body of research is the generalized linear model approach, which allows residuals to be non-normally distributed. A subset of this approach is the iteratively reweighted least-squares approach which has been used to create robust algorithms [127, 116, 297], as well as compressive sensing and sparse signal recovery [78] algorithms. This family of algorithms corresponds to a specific assumption on the errors: they are uncorrelated but may have different variances (i.e. are heteroscedastic) [268]. Indeed, if there are control signals that correspond to random forcing, then the residuals will be of this sort. The following algorithm leverages this work on outlier detection not to remove the "outliers" but to reinterpret them as input on the system.

Residual Analysis

Data generated from a linear process with external shocks of the form in equation 2.12 can be fit using an uncontrolled framework, for example via the least squares method described in 2.4. In this case the regression matrix, called $\hat{\mathbf{A}}$, may be very different from the true linear dynamics $\mathbf{A}$, because it is trying to account for the external input $\mathbf{BU}$. However, if we knew the dynamics, it would be very easy to discover the control signals via rearranging equation 2.12:

$$\dot{\mathbf{X}} - \mathbf{AX} = \mathbf{BU} \tag{2.16}$$

We do not know the true dynamics $\mathbf{A}$, but in many circumstances (TODO) $\hat{\mathbf{A}}$ can be used to approximate this residual and thus the control signals themselves. This extends to the nonlinear case, and if any parameters are known in advance these can be explicitly

specified in this step.

Probabilistic Programming

It is possible to analyze a single residual directly from equation 2.16. However, the residual of any single model realization will be very sensitive to the exact training data, measurement noise, and numerical differentiation errors. In some cases this sensitivity can be mitigated, as shown in Fig. S3. A more statistically sound alternative is to explore an ensemble of models, producing a distribution of residuals. The presence of outliers beyond the noise envelope is then very obvious, as the noise envelope is explicitly modeled and fit. These outliers are then the initial guess for the control signal, as shown in Fig. 3.10.

In addition, this Bayesian extension of the original SINDy algorithm automatically produces a posterior distribution for the model parameters as shown in the supplement.

2.2 Biology

2.2.1 Survey of Modeling efforts

Of general interest to the biology community is understanding how biomechanical systems process sensory input to produce behavioral outcomes and robust control strategies. Seemingly simple behavioral paradigms such as flying, crawling, and walking all involve complex interactions between neuronal networks of sensory neurons, propioceptive feedback, and muscle activation. Understanding how these various networks interact to produce a robust control strategy remains an open challenge. A model organism that can help elucidate the control laws arising from these complex dynamics is the *Caenorhabditis elegans*: a nematode with only 302 neurons, 95 muscles involved in locomotion, and a well-mapped and stereotyped connectome [294, 289]. Importantly, it has a limited behavioral repertoire that includes four primary motions: forward crawling, backward crawling, omega turns and head sweeps. In this manuscript, we explore a dynamic mechanism that can produce the full repertoire of turns in C elegans in a model optimized for forward motion.

Given its importance as a model organism, there has long been an interest in modeling the behavior and locomotion of the worm (see [139] for a recent review). Broadly, these efforts (i) attempt to model the generation of locomotion within the nervous system alone (e.g. [215, 245, 151, 163, 224, 165, 164, 167]), (ii) model the biomechanics of the musculature/body alone [70, 92, 237, 197, 198, 230, 21, 177], or (iii) build an integrated model for neural and bodily dynamics [216, 51, 37, 38, 85]. Most previous modeling efforts have focused on simulating the simple, sinusoidal bodily postures involved in forward locomotion. It is unclear if said models could be extended to include the more complex behaviors exhibited by the worm. Ultimately, the full complexity of the dynamics may be captured within future high-fidelity, fully three-dimensional particle-based models involving the collaboration of hundreds of scientists and modeling almost every aspect of the *C. elegans* geometry and anatomy [31, 273]. To our knowledge the only model previously shown to be capable of generating complex postures is a non-integrated model of the body alone [70]. This model stops short of considering the role of neuronal dynamics and proprioception in generating complex postures.

Nonetheless, integrated neuromechanical models have generated considerable insight into *C. elegans* locomotion. A notable recent example is the integrated neuromechanical model of Boyle, Berri and Cohen [38], a two-dimensional spring-rod model which uses proprioception to generate sinusoidal locomotion. The model incorporates proprioceptive feedback through specific stretch receptors, which have been long hypothesized [293], and for which there is experimental evidence [179, 293]. Via proprioceptive feedback, the model replicated the experimentally-observed continuous modulation of the worm's forward motion gait in response to its environment [30]. However, this work considered only forward motion, and the model is unable to reproduce other typical behaviors such as backward motion, head sweeps, or omega turns.

In this manuscript, we extend the model of Boyle, Berri and Cohen [38], discovering the necessary modifications for the model to produce the full range of complex *C.elegans* postures. Our modifications produce a single biomechanically realistic model that can produce the

full repertoire of behaviors, including the "omega turn" in which the animal makes a deep bend in order to reverse directions. We show that a traveling wave of suppression on the stretch receptors is necessary and sufficient for this complex behavior. This study suggests that transient, extra-synaptic modulation of the synaptic weights is necessary for complex behavior, which is a vital step for understanding the control paradigm of the animal.

2.2.2 Known connectome

Background

The connectome of the *C. elegans* neuronal network has been known for more than forty years [294], and has been recently updated [289, 141, 72]. All of these projects have used similar experimental techniques, focused on very fine slicing and careful tracking of physiological structures through the body. Each paper recovers part of the connectome from a single individual, and thus the final output is compiled across these individuals. This gives a good view of the adjacency matrix of the network, i.e. which connections exist.

Limitations

In order to go from the connectomic structure to function, several additional properties are needed: edge direction, edge weight, and sign. I will discuss the general consensus and some caveats for each of these properties.

There are two general types of connections in the animal: gap junctions, and chemical synaptic junctions. It is generally assumed that gap junctions are two-way and that chemical synapses are directed. However, there is some work that suggests the possibility of directed or at least asymmetrical gap junctions [153, 188]. In addition, many neurons have both types of edges, sometimes even between the same pair of neurons [188]; it is unknown what effect this might have.

Neurons in *C. elegans* are generally denoted as excitatory or inhibitory [294], based on their expression of Acetylcholine or Glutamate (excitatory) or GABA (inhibitory). However,

there are several complications. The effect of a neurotransmitter on a downstream neuron depends on the receptor that is expressed, and there is evidence that different receptors can change the sign of the interaction [221, 180, 222]

The physiological data shows that not all connections are created equal: some have many more processes connecting them, with a range of roughly 1-50 processes. Many models assume this to be proportional to the edge weight, e.g. [163], but this is not well established.

To truly uncover function from the connectome, the functional form of the connections must be known. This is generally unknown, though circuit-based models [163] have made some progress. However, there is strong evidence of several effects that these models do not account for, in particular changes in time constant for different neurotransmitters [189, 235], nonlinearities related to saturation and spill-over [143], and extra-synaptic connections [29, 25, 258, 223].

In summary, the connectome provides a very incomplete picture of the neuronal network of even this very simple animal. This is one reason why data-driven methodologies both have become very popular, and have the potential to contribute much that cannot be determined from the connectome alone.

2.3 Biomechanical modeling

We review the two-dimensional spring-rod model [38]. This model integrates our dynamic proprioception which generates the repertoire of observed behaviors.

2.3.1 Environmental properties

This model implements the drag coefficient of the body by separating the parallel and perpendicular components. In relatively low viscosity media similar to water, the drag coefficients can be analytically calculated [181], and in highly viscous media like agar, these coefficients have been experimentally estimated [30, 216]. In the model, the medium is a linearly tunable parameter that varies from 0 (water) to 1 (agar), as shown in table 1.

2.3.2 Model components

The two-dimensional model of the *C. elegans* has long been considered a compromise between feasibility and accuracy [138], i.e. it is a parsimonious model that balances complexity with accurate biomechanics. The two-dimensional structure is motivated by the laboratory environment where nearly the entire body moves only in two dimensions along a surface. The only truly three-dimensional behaviors are exploratory head motions, which are outside the scope of this study.

Body Shape and Segmentation

The *C. elegans* body, as shown in Fig. 2.6, is composed of 12 segments organized into 3 different layers of interaction. This approximates the known muscle structure: *C. elegans* has 48 dorsal and 47 ventral muscles, though the body itself is not segmented. A segment refers to 8 passive vertical and diagonal elements containing a set of 4 dorsal and 4 ventral muscles, a pair of stretch receptors, and a pair each of A- and B-class neurons. The body is further divided into 48 sub-segments, 4 per full segment, such that each has a pair of lateral, diagonal, and vertical elements and a single muscle.

The two-dimensional cross-section of the body is approximated by an ellipsoid, with the radius of each of the $M = 48$ sub-segments given by:

$$R_i = R_0 \left| \sin\left(\arccos\left(\frac{i - (M/2 + 1)}{M/2 + 0.2} \right) \right) \right| \tag{2.17}$$

where R_i is the radius of the ith body segment and R_0 is maximium radius.

Rod spring model

The first modeling component is a rod-spring system with passive vertical and diagonal elements, as well as active muscle-driven lateral elements. The vertical rod elements are of a fixed length $2R_i$, given by equation 2.17, and enforce the biological constraint that the radius of the body is nearly constant throughout normal behavior.

The diagonal elements are damped springs that model hydrostatic internal forces. The force from each diagonal element for the ith body segment is given by:

$$f^k_{D,i} = \kappa_D \left(L_{0D,i} - L^k_{D,i}\right) + \beta_D v^k_{D,i} \tag{2.18}$$

where β_D and κ_D are the spring and damping constants, and $L_{0D,i} = \sqrt{L^2_{seg} + (R_i + R_{i+1})^2}$ is the rest length. In addition, $v_i = \frac{d}{dt} L^k_i$ is the rate of change of the length of each element. The subscripts D and, in the next equation, L, refer to either the diagonal or lateral elements. The superscript k denotes which side of the animal (dorsal or ventral) is being considered and which subnetwork is characterized (A-class or B-class). It can thus take on 4 distinct values. Values are identical across the subnetworks unless otherwise noted. The lateral elements are also damped springs, but these are actively driven by the motor neurons,

$$f^k_{L,\ i} = \begin{cases} \kappa_L \left(L_{0L,i} - L^k_{L,i}\right) + \beta_L v^k_{L,i} \\ \qquad\qquad \text{for} \quad L^k_{L,i} < L_{0L,i} \\ \kappa_L \left[\left(L_{0L,i} - L^k_{L,i}\right) + 2\left(L_{0L,i} - L^k_{L,i}\right)^4\right] + \beta_L v^k_{L,i} \\ \qquad\qquad \text{otherwise} \end{cases} \tag{2.19}$$

The rest length for the lateral elements is slightly different: $L_{0L,i} = \sqrt{L^2_{seg} + (R_i - R_{i+1})^2}$. The force output of a given muscle segment is a function of the motor neuron voltage, the muscle length, and the rate of contraction. In addition, a gradient was imposed on the maximum output force of the muscles, reflecting experimentally observations

$$f^k_{M,i} = \kappa^k_{M,i} \left(L^k_{0M,i} - L^k_{L,i}\right) + \beta^k_{M,i} v^k_{L,i} \tag{2.20}$$

with

$$\kappa^k_{M,i} = \kappa^k_{0M} G^k_{NMJ,i}\ \sigma\left(A^k_{M,i}\right) \tag{2.21a}$$

$$L^k_{0M,i} = L_{0L,i} - G^k_{NMJ,i}\ \sigma\left(A^k_{M,i}\right)\left(L_{0L,i} - L_{min,i}\right) \tag{2.21b}$$

$$\beta^k_{M,i} = \beta_{0M}\ G^k_{NMJ,i}\ \sigma\left(A^k_{M,i}\right) \tag{2.21c}$$

and where L_{min} is a minimum muscle length for each subsegment, normalized to have the same maximum curvature. The function $G^k_{NMJ,i}$ is a linearly decreasing function from the initiation of the propagation that captures the experimental fact that curvature decreases as the wave propagates. Additionally, $\sigma(x)$ is a linearized sigmoidal function of the muscle activation:

$$\sigma(x) = \begin{cases} 0 & , \quad x \leq 0 \\ x & , \quad 0 < x < 1 \\ 1 & , \quad x \geq 1 \end{cases} \tag{2.22}$$

Parameter	Value
M	48
N	12
L	1mm
$\mathrm{L}_{\mathrm{seg}}$	L/M
$\mathrm{CL}_{\mathrm{water}}$	$1.65 \cdot 10^{-6}/\left(\mathrm{M}+1\right)$
$\mathrm{CN}_{\mathrm{water}}$	$2.6 \cdot 10^{-6}/\left(\mathrm{M}+1\right)$
$\mathrm{CL}_{\mathrm{agar}}$	$1.6 \cdot 10^{-3}/\left(\mathrm{M}+1\right)$
$\mathrm{CN}_{\mathrm{agar}}$	$64 \cdot 10^{-3}/\left(\mathrm{M}+1\right)$
κ_L	$0.02\ \mathrm{kg} \cdot \mathrm{s}^{-1}$
κ_D	$\kappa_L \cdot 350$
κ_{0M}	$\kappa_L \cdot 20$
β_L	$\kappa_L \cdot 0.025\mathrm{s}$
β_D	$\kappa_D \cdot 0.01\mathrm{s}$
β_{0M}	$\beta_L \cdot 100$
$\mathrm{L}_{0L,m}$	$\sqrt{\mathrm{L}_{seg}^2 + \left(\mathrm{R}_m - \mathrm{R}_{m+1}\right)^2}$
$\mathrm{L}_{0D,m}$	$\sqrt{\mathrm{L}_{seg}^2 + \left(\mathrm{R}_m + \mathrm{R}_{m+1}\right)^2}$
Δ_M	$0.65 \cdot \left(\mathrm{R}_m + \mathrm{R}_{m+1}\right)$
$\mathrm{L}_{\mathrm{min,m}}$	$\mathrm{L}_{0L,m}\frac{1-\Delta_M}{2R}$
R_0	$40\mu\mathrm{m}$
ϵ_{hyst}	0.5

Motor neurons

A second critical component of the model is a simplified connectomic structure. In each segment, the pair of 4 muscles receive input from two separate classes of excitatory neurons. These A- and B-class motor neurons form separate subnetworks that are experimentally well-known to be active in backwards and forwards motion, respectively. Each neuron is modeled

as bistable neuron which transitions instantanteously and is either "on" or "off," $S = \{0, 1\}$.

$$S_i^k = \begin{cases} 1 & \text{for } \; I_i^k > 0.5 + \epsilon_{hys}\left(0.5 - S_n^k\right) \\ 0 & \text{otherwise} \end{cases} \tag{2.23}$$

Although there is some evidence that muscles display graded transmission [190], there is also biological evidence for bistability in the worm [112, 208, 105]. Previous work addressing this issue explicitly [38] found no significant difference in behavior when the neurons were modeled using a continuous model of the membrane voltage. The current term is composed of three inputs into each of these motor neurons, given by cross-inhibition, a "command" neuron, and proprioception:

$$I_i^k = w_-^k S_i^{\bar{k}} + I_{AVA/AVB}^k + I_{SR,i}^k \tag{2.24}$$

The first two terms are explained in detail in the following paragraphs while the third, which contains the key contributions of this work, is detailed in the next section.

In the real worm the contralateral inhibitory GABA-ergic D-class neurons synchronize muscle contractions so that when one side is contracting the other is relaxing. These D-class neurons are connected to the A- and B-class neurons and their activity is highly correlated. Thus in this model, cross-inhibition is applied directly in proportion to the activity of the excitatory neurons on the opposite side, and is captured in the term $w_-^k S_i^{\bar{k}}$. The second superscript, $\bar{k}$, refers to the opposite side of the animal (dorsal or ventral).

In the full connectome, these motor neurons are part of larger locomotion circuits and this is modeled here as the second input, from a single *command neuron*. This approximation does not assume that there exists a CPG for the production of oscillatory behavior, but is not incompatible with a hybrid CPG and proprioceptive mechanism. Which (DC) current a neuron receives depends on which subnetwork it is part of, with A-class (backwards) neurons receiving current from the command "AVA," and B-class (forwards) neurons receiving current from command "AVB."

2.3.3 Proprioception

We now review the proprioceptive components of the model and introduce our modifications towards a more general dynamic model of proprioception.

Stretch receptor current

The remaining input (2.24) into the motor neurons, $I^k_{SR,i}$, is also the final component of the biomechanical model: proprioception. Stretch receptors have long been hypothesized to exist due to long, undifferentiated "arms" that extend from the A- and B-class motorneurons down the length of body [293]. Shown in Fig. 2.6 is how a stretched body segment produces a strong signal for the posterior body segments on the same side, and a weak to non-existent one on the opposite side. The number of segments to be summed over is given by a parameter $s = \min(M;\ N_{SR} + (n-1)N_{out})$, which is a constant determined by the number of remaining posterior body segments. The full sum is

$$I^k_{SR,i} = (1 - \alpha(t)) \cdot A_i \cdot G^k_{SR,i} \sum_{j=(n-1)N_{out}+1}^{s} h^k_j \tag{2.25}$$

where

$$A_i = \begin{cases} 1 & , \quad (n-1)N_{out} \leq M - N_{SR} \\ \sqrt{\frac{N_{SR}}{M-(n-1)N_{out}}} & , \quad (n-1)N_{out} > M - N_{SR} \end{cases}. \tag{2.26}$$

The term $(1 - \alpha(t))$ is the time dependent term that allows for dynamic suppression of this current, and will be explained in the next section. The parameter A_i compensates for the fact that for segments close to the posterior of the animal, there are fewer segment contributions. Additionally, the parameter

$$G^k_{SR,i} = \begin{cases} 0.65 \cdot \left(0.4 + 0.08 \cdot (N - i - 1) \cdot \frac{2N_{seg}}{12N_{seg\ per}}\right) \\ \qquad \text{for} \quad k = A \\ 0.65 \cdot \left(0.4 + 0.08 \cdot i \cdot \frac{2N_{seg}}{12N_{seg\ per}}\right) \\ \qquad \text{for} \quad k = B \end{cases} \tag{2.27}$$

is a gradient that increases posteriorly for forward motion (B class) and anteriorly for backward motion (A class), to make the receptors more sensitive to the decreased curvature of the body (shown in figure 4.2). Finally,

$$h_i^k = \lambda_i \gamma_m^k \frac{L_{L,m}^k - L_{0L,m}}{L_{0L,m}} \tag{2.28}$$

is a stretch receptor activation function with parameters:

$$\lambda_i = \frac{2R_0}{R_i + R_{i+1}} \tag{2.29}$$

which compensates for the variable radius of each segment with

$$\gamma_m^k = \begin{cases} 1 & , \quad k = V \\ 0.8 & , \quad k = D;\ L_{L,m}^k > L_{0L,m} \\ 1.2 & , \quad k = D;\ L_{L,m}^k < L_{0L,m} \end{cases} \tag{2.30}$$

which compensates for the previously mentioned asymmetry in the inhibitory circuit. The proprioceptive stretch sensors form the fundamental oscillatory mechanism of the model.

An important note is that proprioceptive feedback for forward motion in our model can be described as an *anteriorly* directed signal encouraging contraction from a *stretched* posterior segment, which is consistent with the physiology of the B motor neurons [294]. In contrast, Quen et al. in [293] provide experimental evidence that proprioception acts as a *posteriorly* directed signal for contraction from a *contracted* anterior segment. It is possible that both of these mechanisms are correct, and possible distinguishing experiments are discussed later in this manuscript.

Dynamics of proprioception

Unlike simple forward and backward locomotion, which are long-lived oscillations of the network, the omega turn is a transient behavior which only lasts on the order of a few seconds. We phenomenologically model this as a wave of modulation in neuron properties

that travels posteriorly along the body. Though the behavior is robust to these details, the functional form used is a two-sided sigmoidal function:

$$\alpha(t) = \frac{1}{2}\left[\tanh\left(s\left(t - t_{start}\right)\right) - \tanh\left(s\left(t - t_{end}\right)\right)\right] \tag{2.31}$$

where t_{start} and t_{end} are respectively the initiation and completion of wave, and s models the speed at which this suppression takes effect. This function can be used to smoothly tune a parameter to 0 and then back to its full value, as well as increasing or decreasing parameters by a percentage using $1 - \alpha$ and $1 + \alpha$, respectively. This addition to the original model is vital for complex and transient behaviors like the omega turn and other shallow turns, and is implemented in equation 2.25.

2.3.4 Numerical modeling

The original model used Sundials version 2.3.0 [38]; this paper uses version 2.6.1. The numerical simulation portion of the code is written in C++. Based on the original paper, a visualization package written in MATLAB 2016b is included. The model and dynamics proposed here are all fully reproducible, with the code and example datasets openly available at [101]

2.3.5 Whole-brain imaging (with immobilized caveats)

Calcium imaging is a now a well-established method for imaging many model organisims [257, 267]. This produces fluorescence data which is a convolution of the quantity of interest [267]. There are several easy to use packages for deconvolving Calcium fluorescence time traces [103], but most are focused on recovering spikes; *C. elegans* are non-spiking.

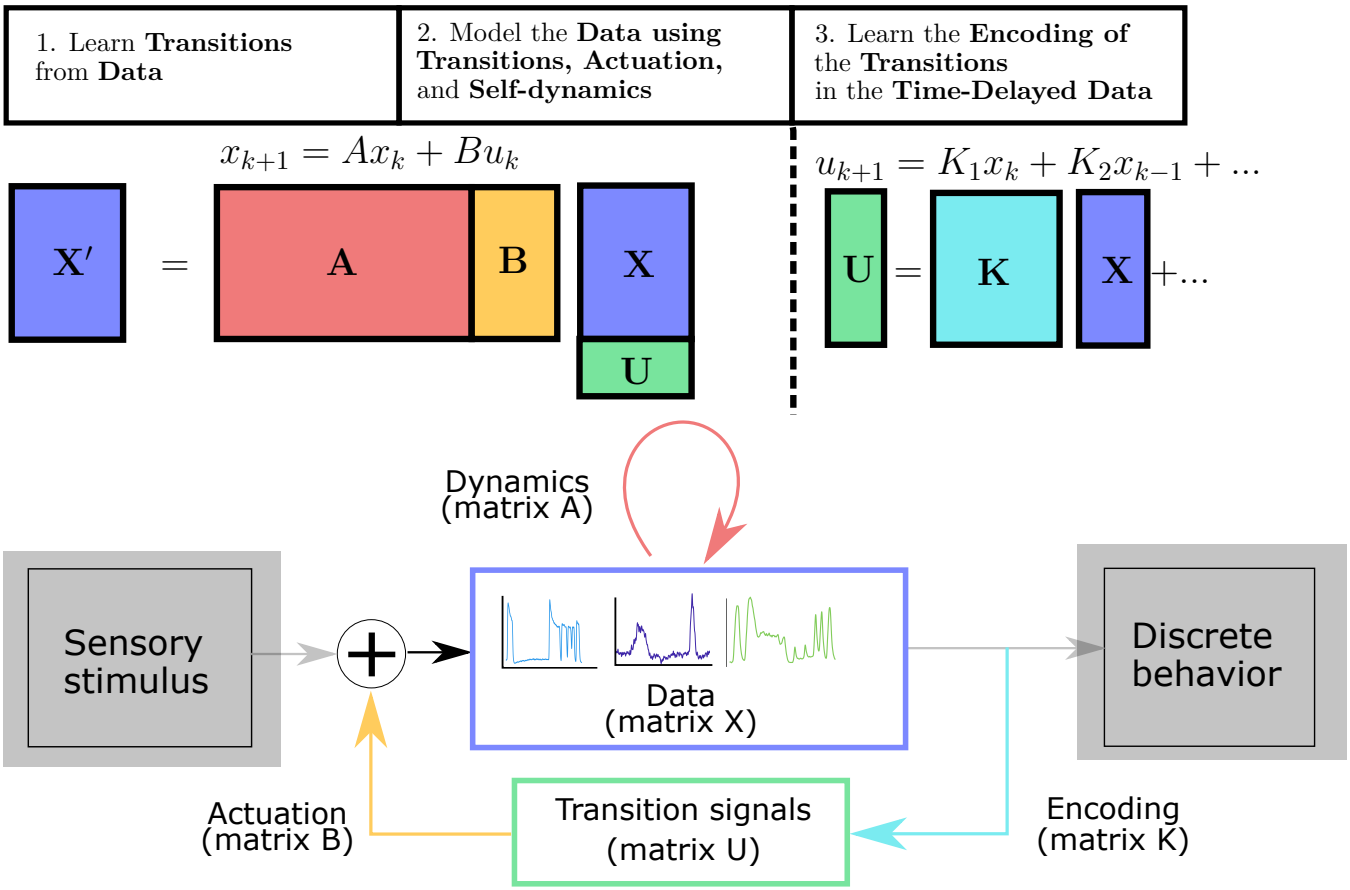

Figure 2.4: A 3-step framework for modeling neurosensory integration. 1) Transition signals are learned from data with an assumption of linear dynamics. 2) A DMDc model is learned which uses dynamics, transition signals, and actuation. These are global models, and are capable of reconstructing much of the data dynamically from an initial state. 3) Where and at what timescales control signals are encoded in the neural activity is studied using sparse linear models. Bottom: the sensory-computation-behavior pathway. Each term in the above equations can be freely translated into this biological process. Transitions (Green) actuate neurons via their own connectivity (Yellow). Neuron traces (Blue) evolve according to intrinsic dynamics (Red), and also encode (Light Blue) the transition signals (Green).

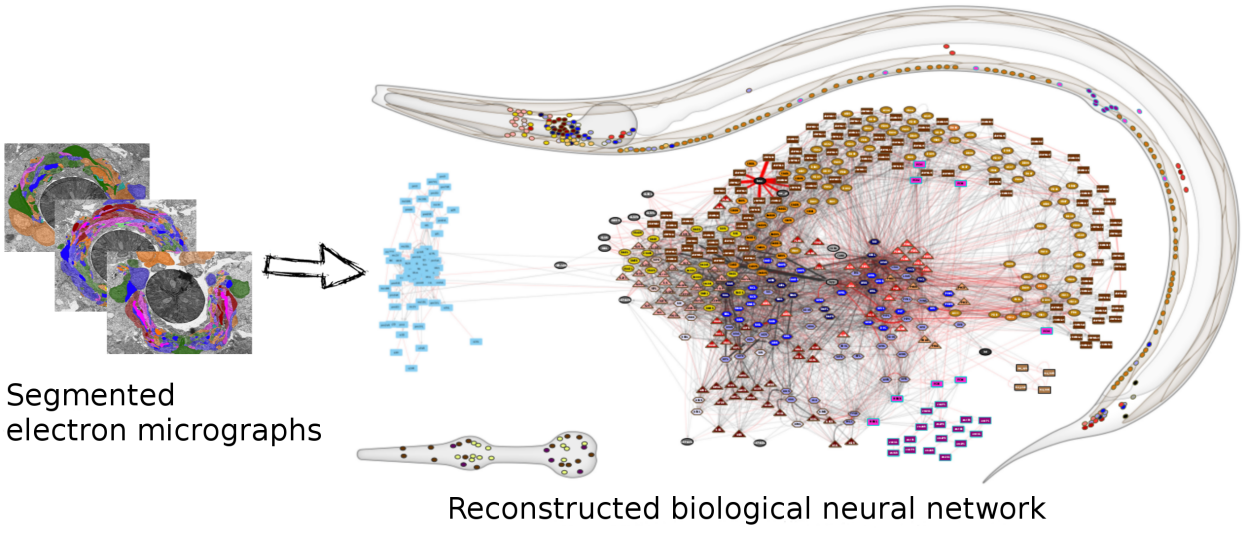

Figure 2.5: Figure from the website www.wormwiring.org [3], an excellent compilation of many connectome-related resources.

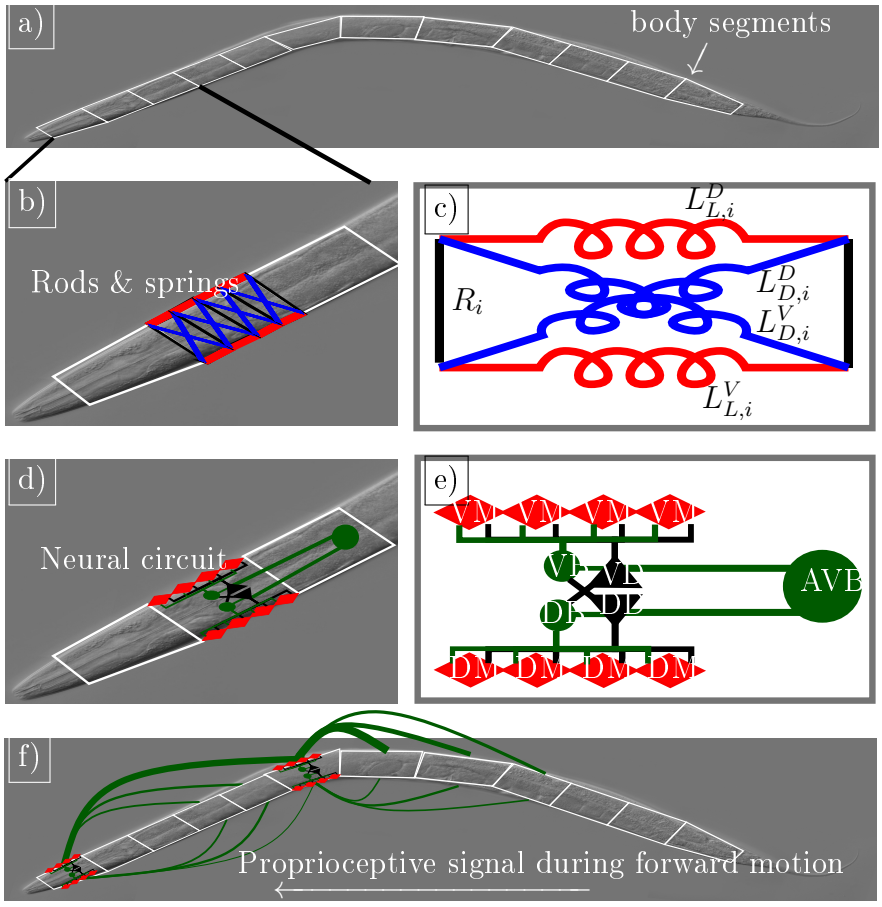

Figure 2.6: Biomechanical model of C elegans. Based off of [38]. a) The body has 12 segments. (b) and (c) Each segment has two rigid vertical components and four damped spring components. The diagonal (blue) elements are passive; the horizontal (red) elements are active and controlled by the neuron voltages. d,e) Each segment also has a simplified connectome model, with four pairs of ventral and dorsal motor neurons, and a pair of excitatory B-class neurons and inhibitory D-class neurons. These are activated by a toy "AVB-like" command neuron, for forward motion. f) Proprioception produces oscillation and more complex behavior. A small curvature will produce almost no proprioceptive signaling, but a stretched segment will.

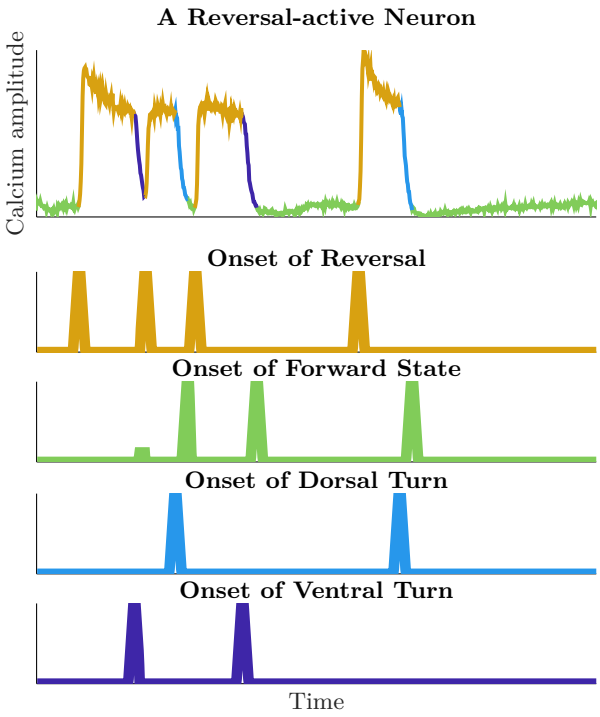

Figure 2.7: Transition signals in *C. elegans*: Top: A calcium imaging trace of a neuron connected with the discrete reversal behavior. Behavioral labels are determined by experimentalists, as described in [152]. Green=Forward; Yellow=Reversal; Dark Blue=Ventral Turn; Light Blue=Dorsal Turn. Below: These labels can be reframed as "onset" signals, and are characteristically sparse in time.

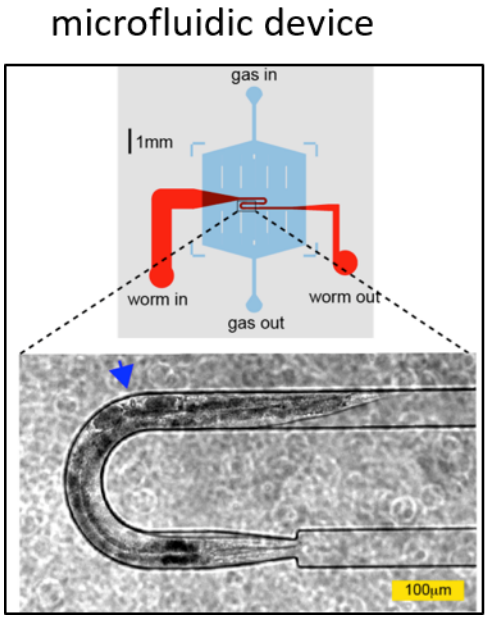

Figure 2.8: Figure from [257], showing the experimental setup for imaging immobilized worms.

Chapter 3

METHODOLOGICAL EXTENSIONS

3.1 Infinite series DMD

3.1.1 Method

Our methods aim to improve the DMD algorithm by minimizing the reconstruction error by using higher powers (multiple time steps) of $\mathbf{A}$. For example, $\min_{\mathbf{A}} ||\mathbf{A}^2\mathbf{X}_1 - \mathbf{X}_3||$ minimizes for $\mathbf{A}$ over two time steps. Via judicious choice of coefficients, a closed form solution can be found for an infinite number of such terms, or an infinite number of future steps. This is referred to as *infinite series DMD* (iDMD).

The iDMD optimization problem is given by

$$\min_{\mathbf{A}} \sum_{k=1}^{d} \left|\left|c_k\gamma^k \left(\mathbf{A}^k\mathbf{X}_1 - \mathbf{X}_{k+1}\right)\right|\right|_F \tag{3.1}$$

In this form, terms cannot be combined. However, due to the special form of the problem and the free parameter γ (see Appendix), the summation can be brought inside the Frobenius norm:

$$\min_{\mathbf{A}} \left|\left|\sum_{k=1}^{d} c_k\gamma^k \left(\mathbf{A}^k\mathbf{X}_1 - \mathbf{X}_{k+1}\right)\right|\right|_F. \tag{3.2}$$

Here d is the number of steps into the future for which we are optimizing, and the regularization terms γ^k, as will be shown, allow control of the convergence properties of the series. The number of terms retained in the series (d) is a free parameter that does not significantly affect performance. This is an upper bound of the full error term, and thus this modified problem is similar to recently popular variational inference methods in machine learning [32].

Geometric Series

A geometric series can be constructed by using $c_k = 1$ in (3.2):

$$\min_{\mathbf{A}} \|\gamma\left(\mathbf{A}\mathbf{X}_1 - \mathbf{X}_2\right) + \gamma^2\left(\mathbf{A}^2\mathbf{X}_1 - \mathbf{X}_3\right) + \gamma^3\left(\mathbf{A}^3\mathbf{X}_1 - \mathbf{X}_4\right) + ...\| \tag{3.3}$$

which can be regrouped as

$$\min_{\mathbf{A}} \| \left(\gamma\mathbf{A} + \gamma^2\mathbf{A}^2 + \gamma^3\mathbf{A}^3 + ...\right)\mathbf{X}_1 - \left(\gamma\mathbf{X}_2 + \gamma^2\mathbf{X}_3 + \gamma^3\mathbf{X}_4 + ...\right) \|. \tag{3.4}$$

Using the closed form representation of the geometric series, the minimization problem (3.2) becomes

$$\min_{\mathbf{A}} \| \left(\mathbf{\Gamma} - \mathbf{I}\right)\mathbf{X}_1 - \mathbf{F}\left(\mathbf{X}, \gamma, d\right) \| \tag{3.5}$$

where

$$\mathbf{\Gamma} = \left(\mathbf{I} - \gamma\mathbf{A}\right)^{-1} \tag{3.6}$$

is the closed form solution for the geometric series. Note that this matrix inverse always exists for convergent series, as shown in the appendix. The data itself is recast into the modified matrix:

$$\mathbf{F}\left(\mathbf{X}, \gamma, d\right) = \sum_{k=1}^{d} \gamma^k \mathbf{X}_{k+1} \tag{3.7}$$

Equation (3.5) can be directly solved using a least-squares regression

$$\mathbf{\Gamma} = \mathbf{F}\left(\mathbf{X}, \gamma, d\right)\mathbf{X}_1^{\dagger} + \mathbf{I} \tag{3.8}$$

where † represents the Moore-Penrose pseudo-inverse. The approximation error for iDMD depends strongly on the parameter γ, which specifies the weighting of snapshot matrices as shown in (3.7).

Transforming this matrix back into the original data space can be accomplished via a single matrix inverse:

$$\mathbf{A} = \frac{1}{\gamma}\mathbf{\Gamma}\left(\boldsymbol{I} + \mathbf{\Gamma}\right)^{-1}. \tag{3.9}$$

Formulation with Control

To penalize higher order powers of dynamics that include control, the signal must be propagated through multiple time steps and will accrue all possible powers of $\mathbf{A}$. In a geometric series this corresponds to simply shifting the series by one step and thus the form of the series is retained. In this case, we end up with two geometric series: the $\mathbf{A}$ matrix multiplying the data matrix $\mathbf{X}_1$, and the $\mathbf{B}$ matrix multiplying the control signal $\mathbf{U}$. The optimization problem, this time over both $\mathbf{A}$ and $\mathbf{B}$ in (2.12) is

$$\min_{\mathbf{A},\mathbf{B}} \|(\boldsymbol{\Gamma}-\mathbf{I})\mathbf{X}_1 - \mathbf{F}(\mathbf{X},\gamma,d) + \gamma\boldsymbol{\Gamma}\mathbf{B}\mathbf{F}(\mathbf{U},\gamma,d-1)\| \tag{3.10}$$

As before, this formulation modifies the future data matrix. The control signal is additionally modified in this formula.

3.1.2 Applications

Benchmark systems and datasets are generated according to equation (2.14), with varying levels of Gaussian noise and dimensionality as described in the titles of each figure. Both a low-dimensional and high-dimensional system are considered.

Comparative Performance

As shown in Fig. 3.1, the error in the eigenvalues for iDMD is comparable to the commonly used exact DMD method [282], though often worse than the debiasing state-of-the-art DMD extensions tls-DMD, fb-DMD, and opt-DMD for small systems. For larger systems, Fig. 3.2 shows that iDMD is comparable to the best other methods. As is expected, the opt-DMD [13] algorithm illustrates the best performance across test cases. For large systems, iDMD is remarkably competitive with state of the art methods.

iDMDc Improves Performance

For real-world datasets with actuation and unknown dimensionality, iDMDc performs significantly better than exact DMDc with heuristic methods for determining the truncation

rank. For very simple and small systems such as in Fig. 3.3, the performance of the two methods is similar. Figure 3.4 compares iDMDc to DMDc using a toy dataset where the true rank is known, and iDMDc is somewhat better. However, the true rank is rarely known, and Fig. 3.5 compares the methods using the Gavish and Donoho [107] truncation method for approximating the proper rank. In this more realistic scenario, iDMDc is much better than exact DMDc.

Not as much work has been done to extend DMD when control is present. fb-DMD no longer has a simple solution if $\mathbf{B}$ is unknown, and tls-DMD can destabilize if the space of these control signal is significantly different from the dynamics. In these figures, iDMDc is only compared to DMDc due to the lack of viable and/or competing extensions. The fb-DMD and tls-DMD variations of DMD cannot be stably extended to the actuated case as demonstrated in the appendix. In particular, opt-DMD solves the inverse differential equation version of the original DMD problem directly, and thus cannot be used with control. Thus iDMDc is competing with the original formulation of exact DMDc, the performance of which strongly depends on the truncation rank.

Robustness of iDMDc

The free parameter that must be chosen by the user in DMD and DMDc, typically via hyper-parameter tuning, is the rank truncation for the SVD. However, although there are some heuristics [107] for determining this truncation, they do not lead to ideal reconstructions, as Fig. 3.5 shows. This restricts the real-world usefulness of the DMDc algorithm, because although its performance can be comparable to iDMD when the underlying rank is known (Fig. 3.4), such knowledge is rare. Instead, the parameter to be chosen is truncation of a different sort, that of the number of terms of the infinite series in equation (3.7). Figure 3.6 shows that the performance of the iDMDc algorithm is largely insensitive to the number of terms kept in the series, and thus a difficult to know and highly sensitive parameter (underlying rank) has been replaced by one that is robust and insensitive. Indeed, hyper parameter tuning of the rank truncation in DMDc is known to be problematic in practice,

as demonstrated in Fig. 3.4.

3.2 Sparse optimization for control signals

3.2.1 Methods

The DMDc algorithm requires knowledge of the linear control signals $\mathbf{U}$. Expert-identified state labels and an example neuron that displays strong state-dependent behavior are shown in figure 2.7. However, these are only available because of the decades of *C. elegans* experimental work identifying 1) discrete behavioral states and 2) the command neurons for each activity. For new organisms, and in order to generate hypotheses about potential new states in *C. elegans*, the unsupervised problem, i.e. learning the signal directly from data, is of critical interest.

DMDc (Eq. 2.12) can be thought of as an error minimization problem

$$\min_{\mathbf{A},\mathbf{B}} ||\mathbf{A}\mathbf{X}_1 + \mathbf{B}\mathbf{U} - \mathbf{X}_2||_2 \tag{3.11}$$

over the dynamics and actuation matrices, $\mathbf{A}$ and $\mathbf{B}$. If the control signal is unknown, the minimization must be extended to the control signal $\mathbf{U}$ itself. However, there is now a trivial solution where the control signal dominates the model: $\mathbf{X}_2 = \mathbf{B}\mathbf{U}$ with $\mathbf{A} = 0$. For this reason, an assumption must be made about the control signals. In this case, the statement that these signals are sparse is directly biologically interpretable, and means that the transitions between states should be rare as a percentage of frames. This "sparsity constraint" can be expressed in a mathematically precise way using the ℓ_0 norm:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{U}} [||\mathbf{A}\mathbf{X}_1 + \mathbf{B}\mathbf{U} - \mathbf{X}_2||_2 + \lambda ||\mathbf{U}||_0] \tag{3.12}$$

Directly solving this optimization problem is extremely difficult, although there are efficient algorithms in certain cases [142]. More recently, a convex relaxation of the ℓ_0 to an ℓ_1 norm is often solved [89], though this has been recently shown to lead to errors in its selection pathway [269]. We use a different approximation, the sequential least squares thresholding algorithm [49], which has been shown to converge to the minima of the original ℓ_0 problem

[305, 308]. The code is outlined in algorithm 3 and more detail is given in the supplement. The matrix $\mathbf{U}$ in this algorithm is additionally constrained to be positive, for better interpretability as "on" transition signals.

Algorithm 2 Unsupervised Learning of Control Signals

procedure LEARNCONTROLLERS(r)
  $\mathbf{U}_0 := InitializeU(\mathrm{r})$
  $\mathbf{S} := InitializeSparsityPattern(\mathbf{U}_0)$
  for $i \leftarrow 1, MaxIter$ **do**
    $\mathbf{A}, \mathbf{B} = SolveAB(\mathbf{X}, \mathbf{U}_{i-1})$
    $\mathbf{U}_i = SolveU(\mathbf{X}, \mathbf{A}, \mathbf{B})$
    $\mathbf{S} = UpdateSparsityPattern(\mathbf{S}, \mathbf{U}_i)$
    $\mathbf{U}_i(\mathbf{S}) = 0$
  end for
end procedure

See appendix A for more discussion of the subtleties with choosing a stopping point in this iterative algorithm, and for connections to information theory metrics. In particular, correlation-based and information theory-based stopping criteria are compared.

3.2.2 Package Release

The above sparse regression methodology is released as a GitHub repository written in MATLAB [100]. Examples are shown in Figs. 3.7 and 3.8 for the autocorrelation-based stopping condition. Additional stopping conditions are implemented, including AIC-based and cross-validation based.

3.2.3 Control signals in drying materials

The main application area of this method is Calcium imaging data that will be described in the next chapter. However, the intention of this method is to be used more broadly. One collaboration that I have begun is with a materials lab studying spontaneous drying events.

In more detail, the goal of this lab is to understand how materials age and how defects accumulate in that process. Depending on the material there are different types of possible defects, including drying and cracking events [174]. These events are measured using sensitive microphones, and the resulting acoustic emissions are analyzed in a variety of ways.

Fig. 3.9 shows an example waveform along with the control signal learned using this optimization procedure. The interpretation of the learned control signal is of the spontaneous formation of a small crack internal to the material. This work is still ongoing, though there are several promising avenues of work: the timing of these events can be made more precise, as the current state of the art uses simple peak detection; more complex onset signals can be analyzed; and the frequencies present in the post-crack dynamics can be studied, particularly as the samples age.

3.3 Subsampling to learn control signals

3.3.1 Sparse Optimization

The full algorithm consists of a multi-step loop and some preprocessing stages, as explained graphically in Fig. 3.10 and more generally in algorithm 3. Initially, the model structure must be chosen, in this case either linear, as in Eq. 2.3 or nonlinear as is Eq. 2.6. This "naive" model is fit to the derivative data. In some cases, particularly if the control signals span multiple orders of magnitude, taking a random intial subsampling can dramatically improve this naive model.

The next step is a loop that refines this initial model guess via modeling a well-selected subset of the data. This subset is the set of gradient points that is well reconstructed by the naive, uncontrolled model. "Well reconstructed" refers specifically to points whose errors are

Algorithm 3 Unsupervised Learning of Controlled Model

procedure LEARNCONTROLMODEL($\mathbf{X}$, $\dot{\mathbf{X}}$)
  $\hat{f}_0 := FitModelDistribution(\dot{\mathbf{X}})$ ▷ Eq. 2.3 or 2.6
  for $i \leftarrow 1, MaxIter$ **do**
    $\mathbf{R}_i := \dot{\mathbf{X}} - \hat{f}_{i-1}(\dot{\mathbf{X}})$ ▷ E.g. Eq. 2.16
    $\dot{\mathbf{X}}_{i,sub} := Subsample(\dot{\mathbf{X}}, \mathbf{R}_i, \lambda)$
    $\hat{f}_i := FitModelDistribution(\dot{\mathbf{X}}_{i,sub})$
  end for
  $U_{final} := ProcessResidual(\mathbf{R}_{final})$ ▷ Sparsify
  $Model := Fit(\dot{\mathbf{X}}, \mathbf{X}, \hat{f}_{final}, U_{final})$ ▷ Eq. 2.15
end procedure

within a factor λ of the noise envelope. Note that these are reconstructions of the gradient itself via equation 2.15, and no integration of these equations is performed. Last, once these intermediate models have converged or after a maximum number of iterations, the remaining residual between the final model and the data is processed to form the final control signal.

There are three free hyperparameters or model choices in this algorithm. First, the class of models must be chosen. In this paper, either linear (DMD) or sparse nonlinear (SINDy) frameworks are chosen. Second, the threshold to use for subsampling the points using the residual and the noise envelope, λ. Third, the convergence criterion or maximum number of iterations. All examples in this paper required a single iteration unless otherwise stated.

3.3.2 Results

Separation of linear dynamics

Fig 1.1 gives a very simple example for a dataset with linear intrinsic dynamics that can be analyzed using this method. The governing equations are simply the action of gravity:

$$\begin{aligned} \dot{v} &= a \\ \dot{a} &= -g \end{aligned} \tag{3.13}$$

where the gravitational constant $g = 9.81$. However, the two sources of external disturbance completely change the long-term behavior of the system, which would naturally simply fall forever. In addition, these control signals only act on a single variable directly: the acceleration. This means that a naive linear model, one that does not account for control, will successfully model the first term but not the second. The best fit least-squares model to this data is:

$$\begin{aligned} \dot{v} &= a \\ \dot{a} &= -0.5v + 0.2a + 7.5 \end{aligned} \tag{3.14}$$

Extra terms appear, and "g" is incorrect. If this model is integrated as in Fig. 1.1 the reconstruction is very bad, however, this model is good enough to produce a good control signal when doing point prediction of the derivative. The residual between these point predictions and the data can be processed as shown in Fig. 3.10 to produce the control signals shown in Fig. 1.1. These control signals then produce a much more accurate set of intrinsic dynamics, with $g = -9.812 \pm 0.05$

Learning nonlinear, chaotic dynamics

Many dynamical systems of interest are nonlinear, and many of these are chaotic. One classic example, which was originally designed as a simplified model of atmospheric convection [263]:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned} \tag{3.15}$$

where $\rho = 28$, $\sigma = 10$, and $\beta = 8/3$. This model can be sparsely represented with a few analytic terms, and can be recovered purely from data using the SINDy algorithm [48]. An unforced version of the attractor is shown in Fig. 3.10.

However, if there are external perturbations of unknown magnitude, frequency, and input dimension then the SINDy algorithm will not successfully recover the dynamics. For the perturbed version of the attractor is shown in Fig. 3.10, the SINDy algorithm with 2nd order library terms produces:

$$\begin{aligned} \dot{x} &= -4.6x + 6.0y - 0.4z + 20.4 \\ \dot{y} &= 23.6x + 1.1y + 18.6 - 0.9xz \\ \dot{z} &= -2.7z + 1.0xy \end{aligned} \tag{3.16}$$

In this dataset, control was only applied to the first two variables (x and y), so the z equation is correct. Most of the terms are similar but one has changed signs, and several new erroneous terms have appeared. Integrating this naive model, as shown for the x variable in Fig. 3.10, produces very poor predictions. However, as Fig. 3.10 shows, as algorithm 3 is applied, the correct attractor and equations are recovered along with the control signals.

Learning nonlinear, spiking dynamics

A model discrepancy of the form in Eq. 2.15 may not be a true external input. In particular, it could be a nonlinearity that happens very quickly relative to data collection. One example is that of spiking neurons, in the which the "reset" after a spike is defined as instantaneous. A two dimensional model that can reproduce spiking patterns from different classes of neurons [135] is:

$$\begin{aligned} \dot{v} &= 0.04v^2 + 5v + 140 - u + I \\ \dot{u} &= a(bv - u) \\ \text{if v} \geq 30 \text{ mV, then } &\begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \end{aligned} \tag{3.17}$$

Where I is the input current, the constants are chosen as in the original paper to give v and t units of mV and ms, respectively. u is a membrane recovery variable, and the parameter values used here are those suggested: $a = 0.02$, $b = 0.2$, $c = -65$, $d = 2$. The input current,

I, is 40 for the Constant Input neuron and is either 40 or 240 for the Variable Input case in Fig. 3.11.

Discrepancy modeling as a field can have a problem of identifiability [6], where the model for the discrepancy cannot be distinguished from the core model. In this framework the discrepancy can be any time series, with the assumption that it is sparsely active, thus one identifiability problem can be that of multiple control signals. As shown in 1.1, the control from the ground and the external forcing are learned as part of the single time series that is input onto the second variable (acceleration). A similar phenomenon happens with these spiking neurons, where the "control signal" associated with resetting voltage cannot be disentangled from modulation in external current input, I, as demonstrated in Fig. 3.11.

3.4 Using DMD as a network reconstruction method

Attempting to discover structure purely from data has become a popular subfield in computational neuroscience [287, 131]. There are several different perspectives that these techniques take on the problem, and they can be largely divided into correlation-based, e.g. Granger Causality [86], and model-based methods, e.g. Bayesian spike models [217, 238]. In addition, DMD appears to be exactly this type of method, with the $\mathbf{A}$ matrix acting as the adjacency matrix and describing both whether or not a connection exists, as well as the sign. The following sections detail major pitfalls that restrict the validity of this interpretation to a very narrow realm of systems.

3.4.1 Example: Diffusion

The issues with this interpretation can be directly seen using an example. We will use a system which has a matrix propagator of the exact form seen in Eq. 1.2, that of simple diffusion:

$$\dot{\mathbf{x}} = -D\nabla^2\mathbf{x} = \mathbf{A}\mathbf{x} \tag{3.18}$$

where ∇^2 is the Laplacian and D is the diffusion constant, but both can be written as a single matrix $\mathbf{A}$. An example for a spatially-discretized system is shown in Fig. 3.14. Note that this describes a spatial system that is 5 by 5, with periodic boundary conditions. Mathematically, the vector $\mathbf{x}$ in Eq. 3.18 is unrolled to be 25 by 1, making the matrix $\mathbf{A}$ 25 by 25.

3.4.2 Failure to reconstruct the network

First attempt: L2 fitting

The adjacency matrix $\mathbf{A}$ in Eq. 3.18 can be solved for using Eq. 2.4, which will produce the best L2 fit. As shown in Fig. 3.15, this regressed matrix is nothing like the true matrix! Indeed, there is no discernable structure at the single-entry level that is comparable to the true dynamics.

Second attempt: Sparsity

Recently, sparsity has become a buzzword that has found applications in a large number of data-driven problem domains [55], including DMD [146]. Thus, our second attempt was to take into account the clear sparsity of the true dynamics and thereby perhaps reconstruct the adjacency matrix.

There are two popular algorithms in use for imposing sparsity: ℓ_1 regularization, aka LASSO [281, 123], and sequential least squares thresholding [33]. Fig. 3.16 shows the failure of sequential least squares thresholding method, and Fig. 3.17 shows the same for the LASSO algorithm.

3.4.3 Explanation: repeated eigenvalues

In fact, this phenonmenon has a very simple explanation which does open up the possibility of two more successful algorithms which will be described later. This explanation centers on the presence of repeated eigenvalues in this system: 1. Repeated eigenvalues form a subspace

that is spanned by orthogonal eigenvectors. 2. Those eigenvectors are only unique up to a rotation. 3. In addition, all trajectories within the subspace will follow a straight line path, as the dynamics are entirely described by the eigenvalue in a linear system. 4. Thus, the within-subspace portion of the trajectory of any single dataset can be described by a single eigenvector and eigenvalue, rendering the rest of the subspace invisible.

Fig. 3.18 shows the multiplicity of eigenvalues for this 5 by 5 system. Although the best L2 fit produces spurious eigenvalues, even a simple rank truncation using equation 2.3 will remove these. Thus, the information that is recovered is a correct subset of the true dynamics. However, because it is a subset of the true eigenvalues and eigenvectors, the individual matrix entries do not converge to the correct system.

Note that this is a special case, for linear systems, of the fundamental limitations on network reconstruction described in [5].

3.4.4 Algorithm 1: Enough data

The first algorithm that can overcome this issue and recover the true network structure and weights simply combines multiple datasets. However, the multiple datasets must be combined at the eigenvalue and eigenvector level, as opposed to the matrix-entry level. This algorithm is described in Alg. 4, and relies on the fact that unless initial conditions are chosen very carefully, the single eigenvector that describes a trajectory for one dataset will not be the same as any other. Thus, randomly chosen initial conditions will, with high probability [55], explore the full subspace of repeated eigenvalues.

Algorithm 4 Network Reconstruction using many datasets

procedure NETWORKRECONSTRUCTION(r)
    $initialize(allVals, allVecs)$
    $i = 1$
    $hasNotConverged = true$
    while $hasNotConverged$ **do**
        $\mathbf{newVals}, \mathbf{newVecs} = DMD(\mathbf{X_i})$ ▷ Solves eq. 2.4
        $\mathbf{valClusters}, \mathbf{vecClusters} = cluster(newVals, allVals)$
        $clusterRanks, hasNotConverged = check_rank(vecClusters, clusterRanks)$
        $\mathbf{allVals}, \mathbf{allVecs} = orthogonalize(vecClusters, allVecs)$
        $i = i + 1$
    end while
end procedure

3.4.5 Algorithm 2: Spatial constraints

One way to make a network identiable from data is to destroy the rotational invariance of the subspaces spanned by the repeated eigenvalues. This would mean that even a single trajectory in that subspace could no longer be describe by a single eigenvector, but would require more basis functions to describe. In practice, this invariance can be destroyed using information that is at least partially available in many real systems of interest: disallowed connections. In the extreme, this means already knowing the adjacency matrix, also referred to as the connectome in neuronal networks. The remaining problem becomes determining the signs and weights of the connections, which would be a significant advance over the current state of the art in many systems, for example *C. elegans* where such weight information is unknown [294, 289].

Fig. 3.19 shows the successful regression in the case of an entirely known "connectome" for the diffusion problem. However, it remains for future work to show how much connectivity knowledge is enough, and if there are cases when even full connetome knowledge does not

fully constrain the single matrix entries.

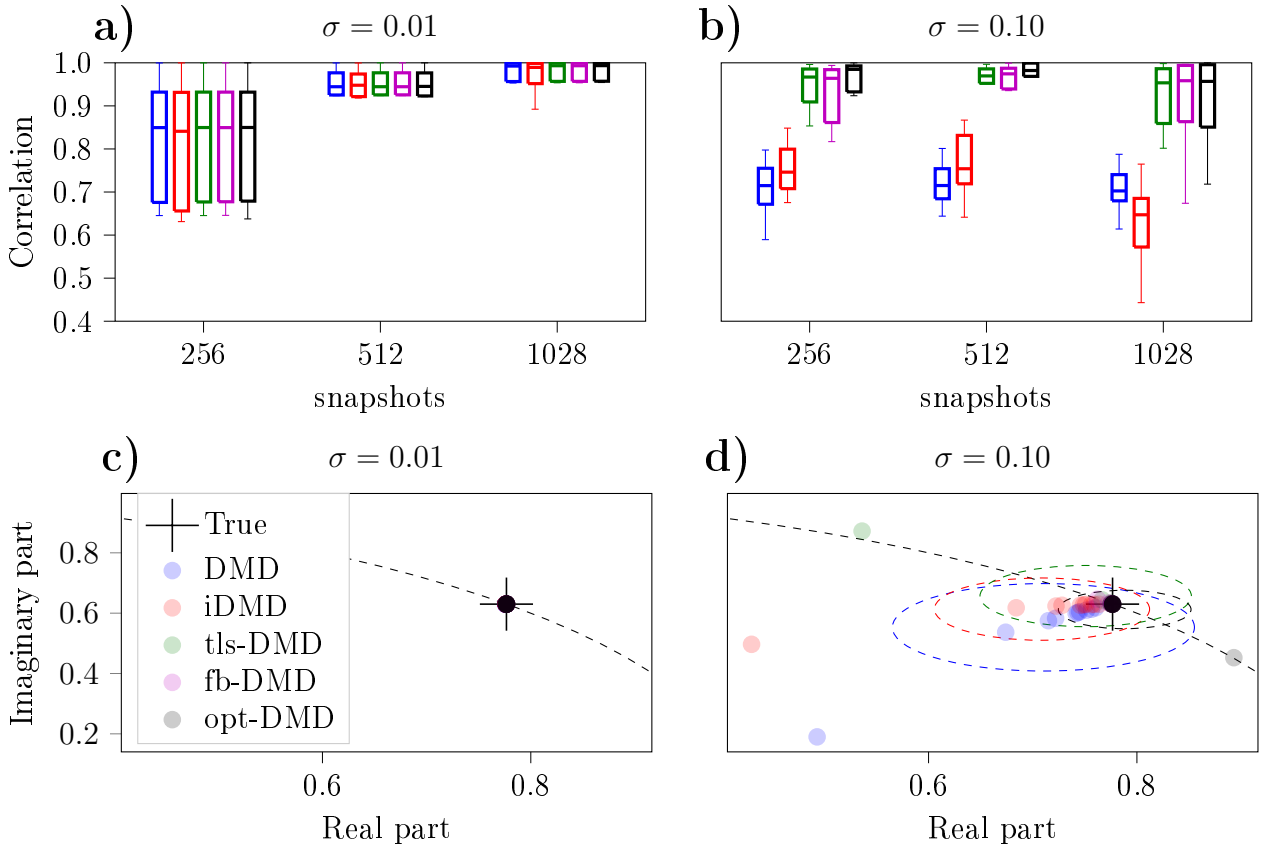

Figure 3.1: Comparisons of iDMD with four leading DMD methods in a small system ($n = r = r_0 = 2$). Overall, iDMD performs similarly to exact DMD (blue). a) The correlation of data reconstructions for different methods at different noise levels. b) The same methods and color schemes, but for one of the eigenvalues of the dynamical system as well as the unit circle (dashed black line). fb-DMD is particularly accurate, similar totls-DMD. However, the latter has a single outlier eigenvalue that drastically increase the variance of the eigenvalues. opt-DMD is generally the most accurate method available.

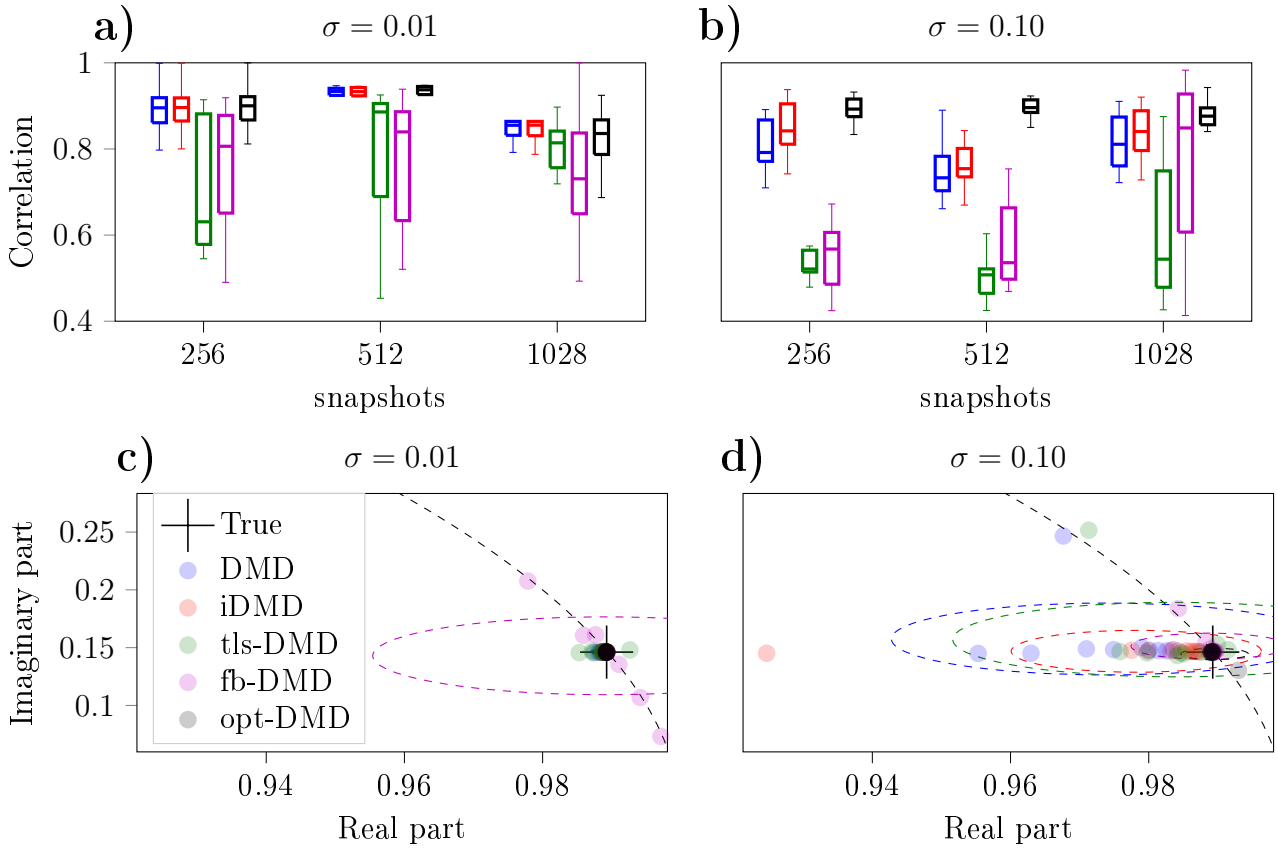

Figure 3.2: Comparisons with four leading DMD methods in a large system ($r = r_0 = 20, n = 70$). Overall, iDMD provides a performance boost over exact DMD. a) The correlation of data reconstructions for different methods at different noise levels. Even though the eigenvalues are relatively precise, if they are outside the unit circle then the reconstruction will quickly diverge. b) One of the eigenvalues and the unit circle (dashed black line). Forward-backward DMD (fb-DMD) is no longer very accurate and has unstable eigenvalues, i.e. outside the unit circle. iDMD is the most precise for both noise scenarios.

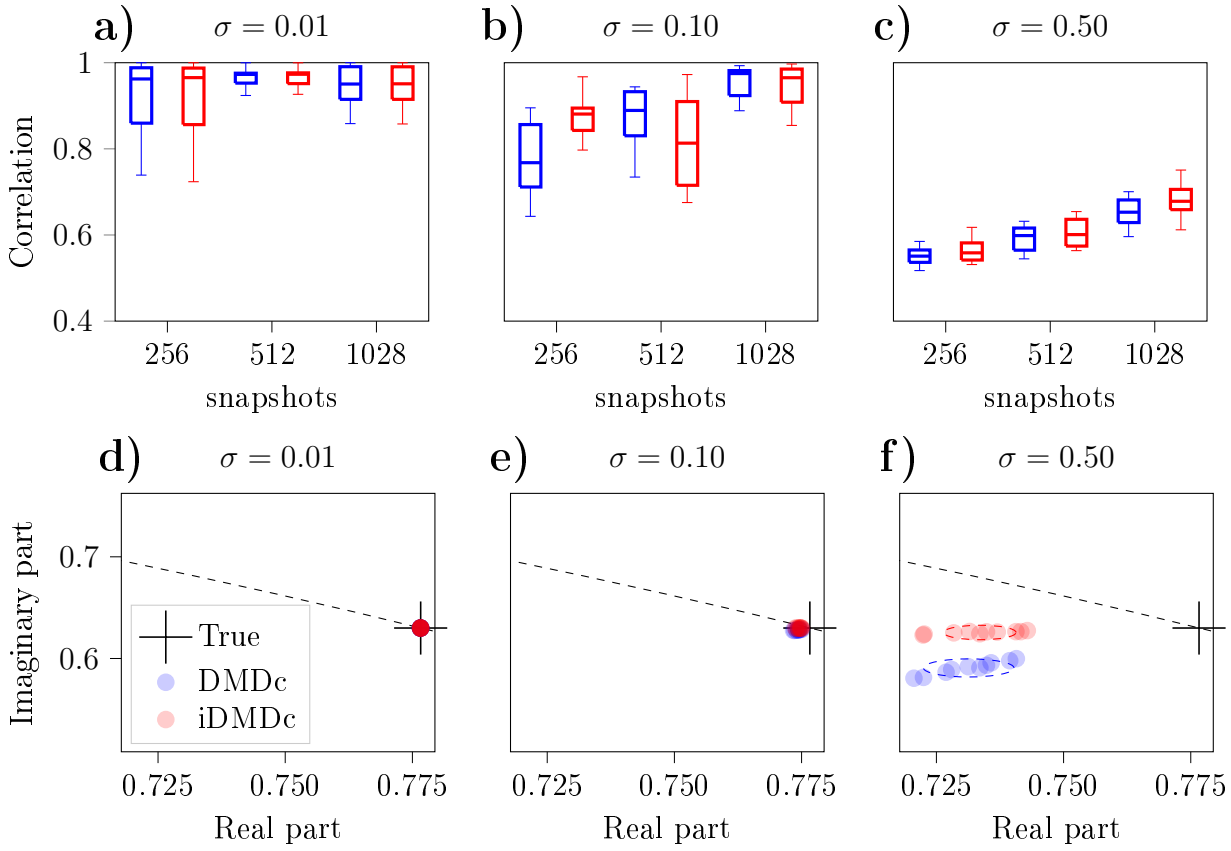

Figure 3.3: Comparison of correlations between data and reconstructions, as well as eigenvalues in the actuated setting for a small system ($n = r = r_0 = 3; r_A = 2, r_B = 1$). Both DMD and iDMD methods have similar performance for a system of this size, though the eigenvalues for DMDc (b) are more biased.

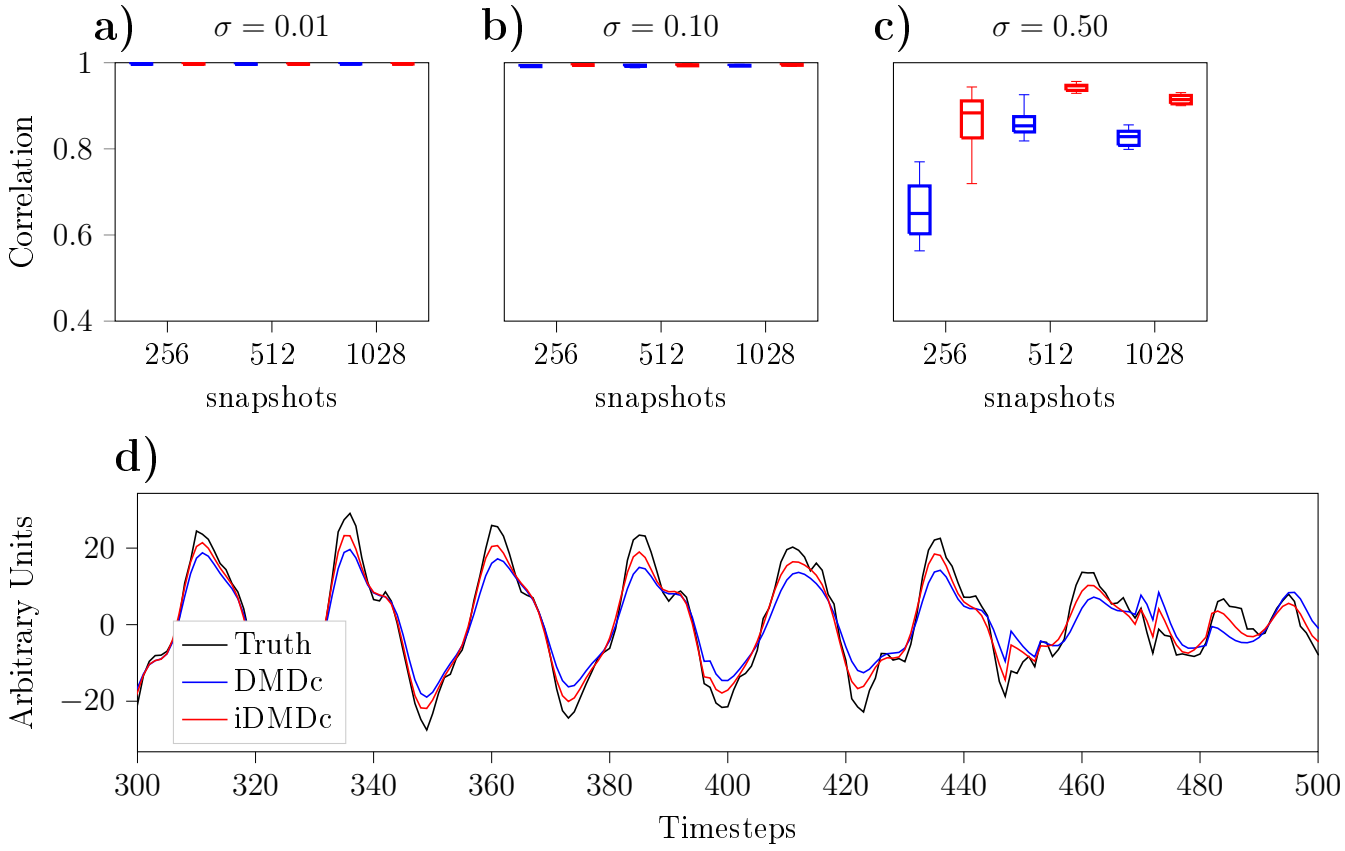

Figure 3.4: Comparison of DMDc and iDMDc methods when the truncation rank is known ($r_0 = r = 25; r_A = 20, r_B = 5; n = 70$). a) In this relatively unrealistic scenario the methods perform similarly, although iDMDc is still more accurate particularly for higher noise levels. b) For $\sigma = 0.10$, the reconstructions are of similar quality.

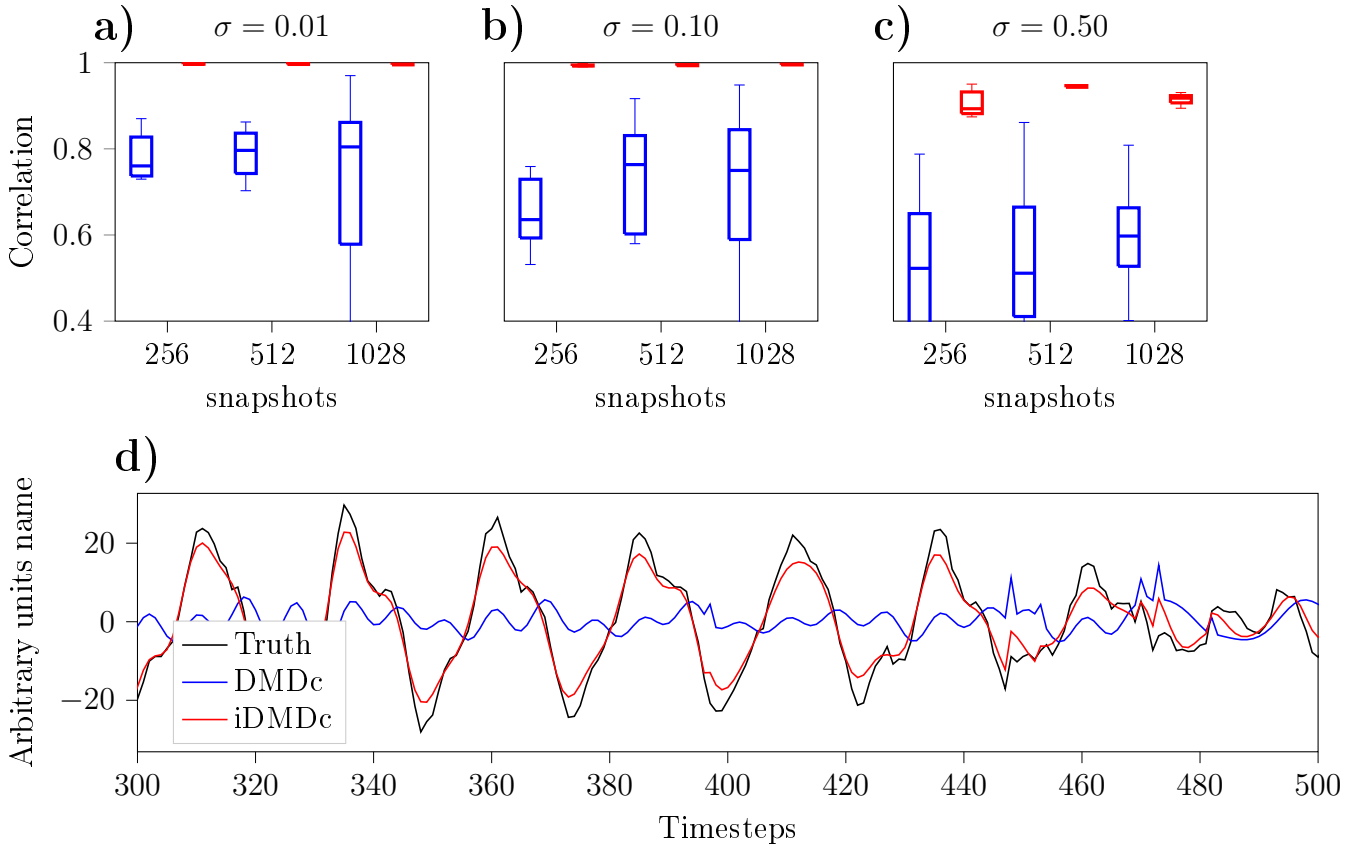

Figure 3.5: Comparison of methods when the truncation rank is unknown ($r_0 = 25; r_A = 20, r_B = 5; r =$ Adaptively determined; $n = 70$). a) In this more realistic scenario, iDMDc is significantly more accurate for all noise levels. b) For $\sigma = 0.10$, the reconstruction of iDMDc (red) is clearly better.

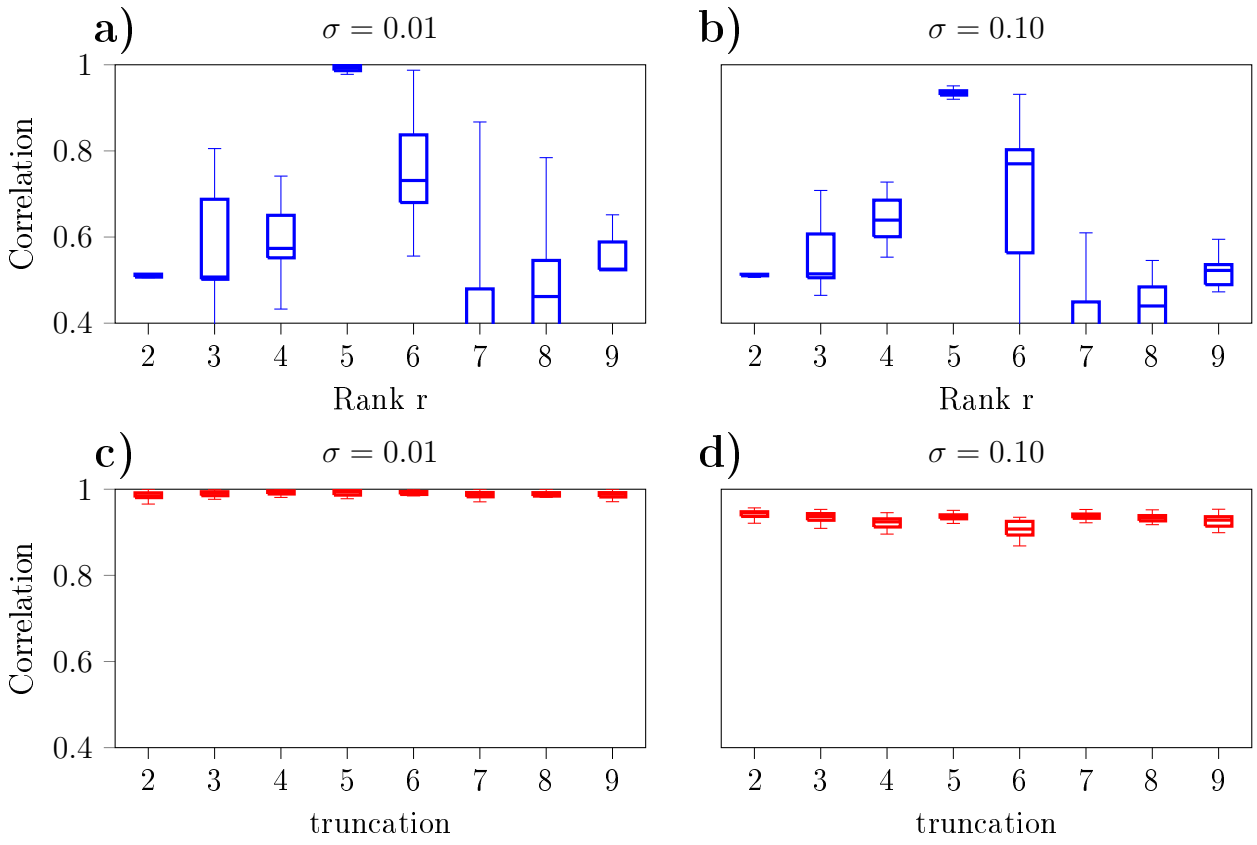

Figure 3.6: Sensitivity of algorithms to choice of parameters for a small system ($r_0 = 5; r_A = 4, r_B = 1; n = 10$). a) DMDc has good performance only for the exact choice of the underlying dimension. b) For iDMDc, as the truncation parameter is increased, γ can be chosen so that the truncation error is lower than a tolerance value in this case 10^{-8}. The performance of the algorithm is slightly better than the best DMDc value, and is not very sensitive to the truncation value.

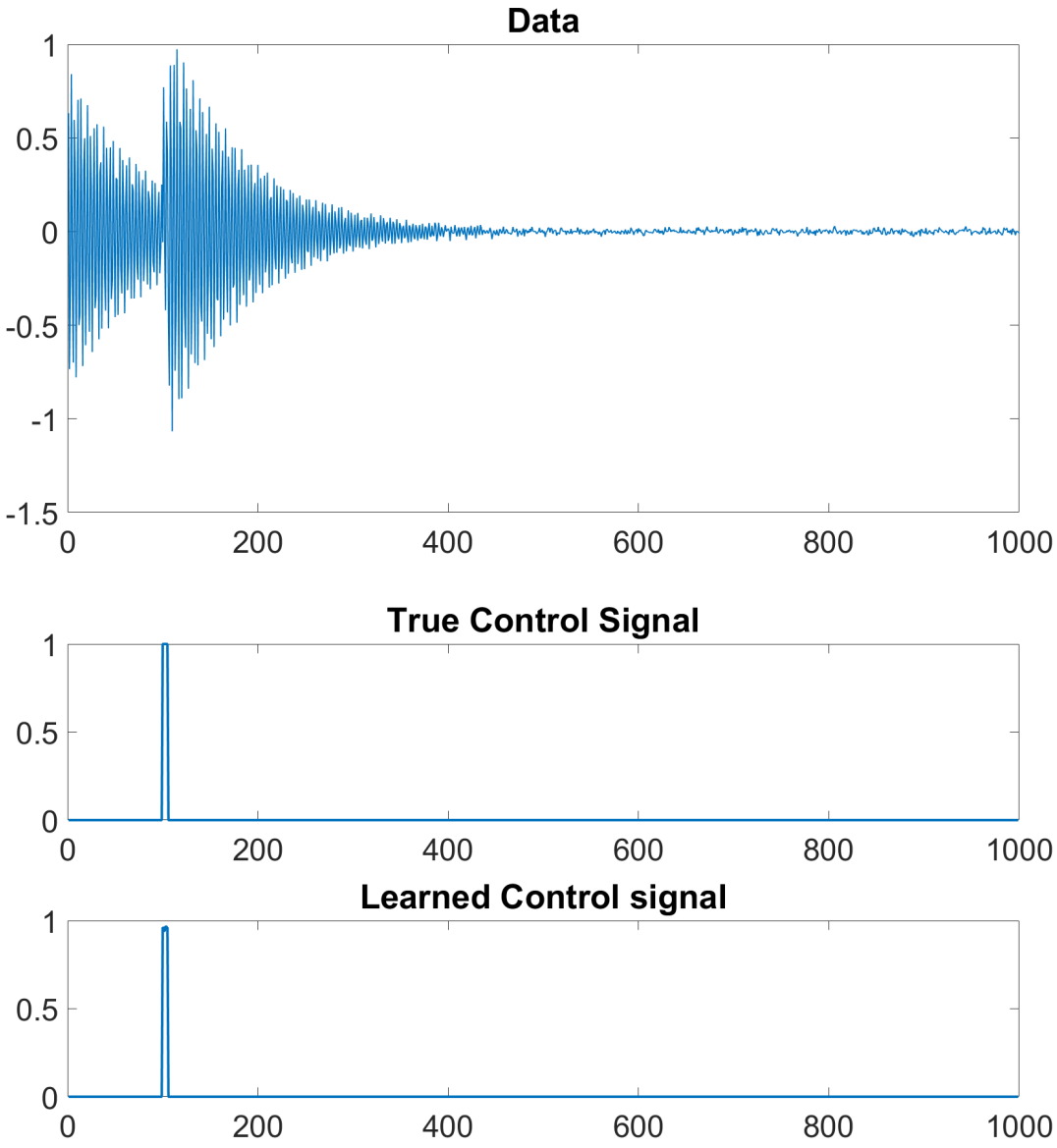

Figure 3.7: a) Example dataset with very low noise level ($\sigma = 0.01$). b) True and recovered control signal.

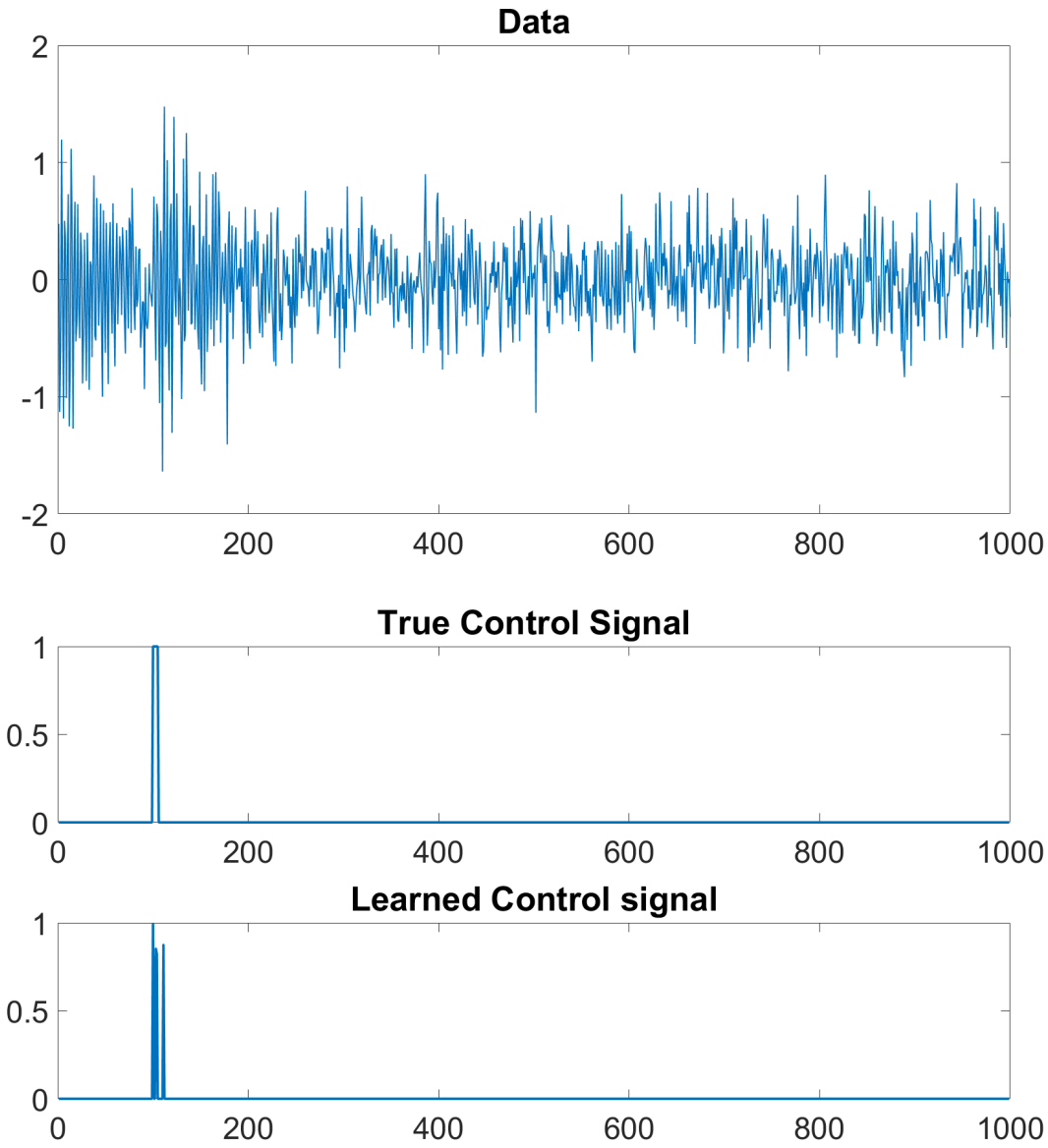

Figure 3.8: a) Example dataset with large noise level ($\sigma = 0.3$). The true dynamics is the same as Fig. 3.7 b) True and recovered control signal.

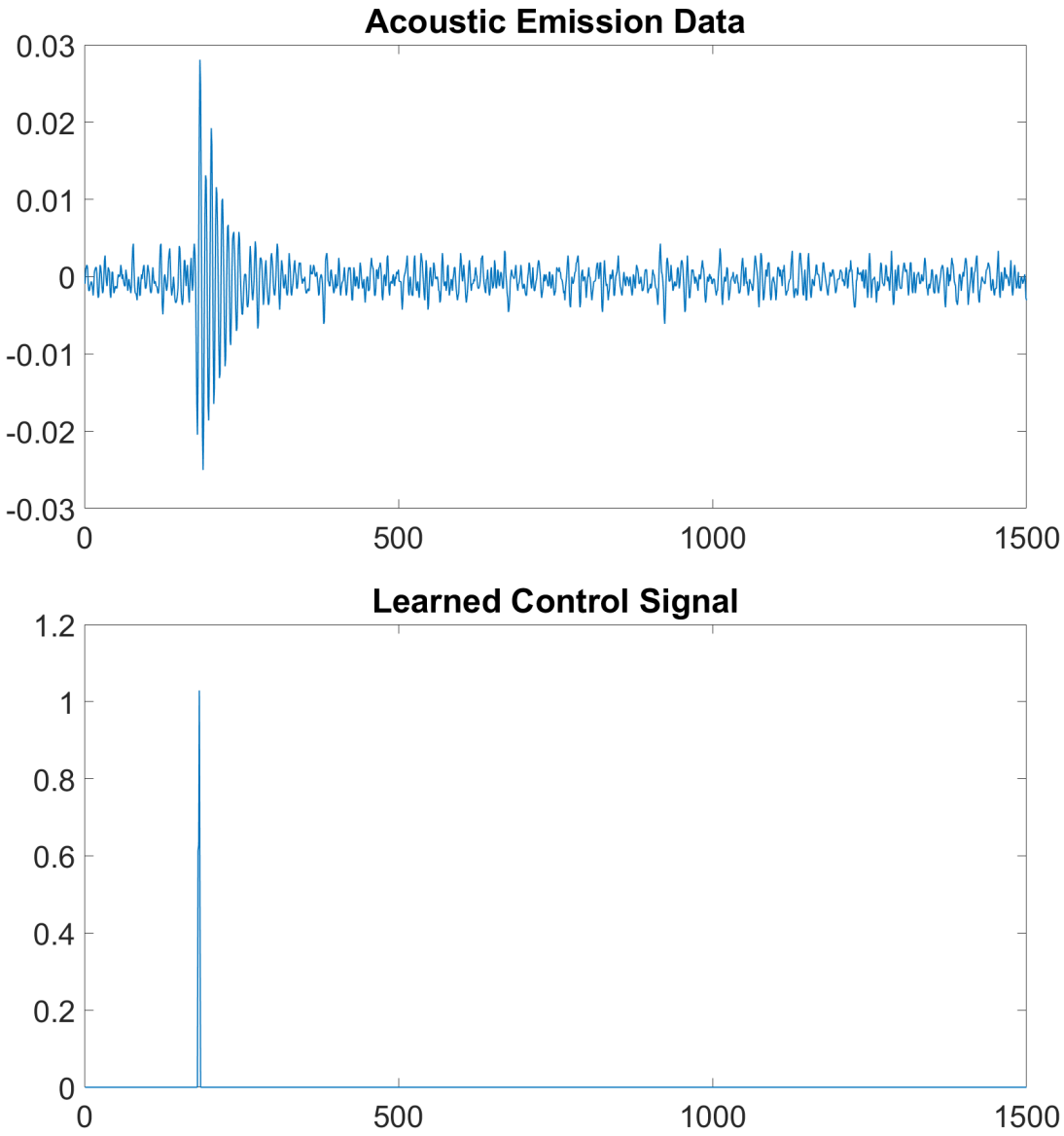

Figure 3.9: a) Example waveform for an acoustic emission from a mortar sample as it undergoes drying. b) Learned control signal

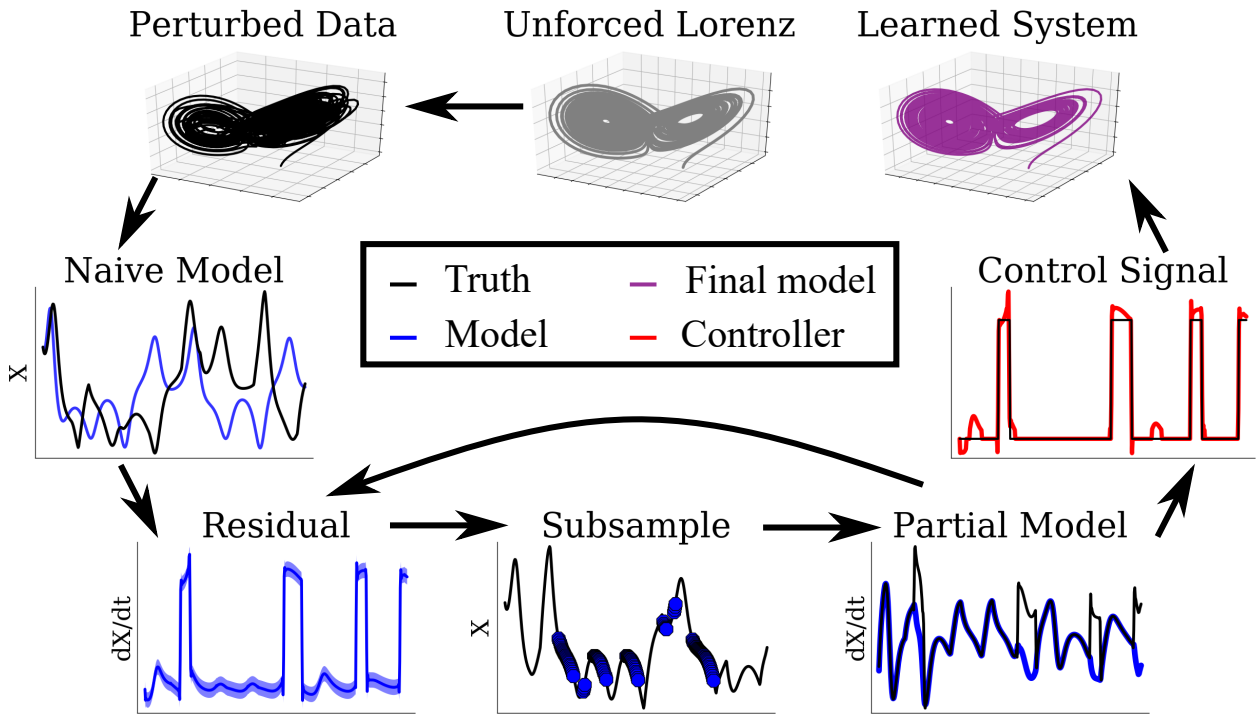

Figure 3.10: Beginning with data (in this case the Lorenz attractor with time-dependent external forcing), there are six steps to the model: 1. Fit a naive ODE. When integrated, this reconstruction will be very poor. 2. Find the posterior distribution of residuals of this naive ODE to numerically calculated derivatives. Note that this uses a collocation method, not integration. 3. Subsample the data, choosing the data points with small residual in the naive model. 4. Fit a "partial" model on the smaller sample of data. The control signals will not be captured, but the intrinsic dynamics may be. If they are not fit well, then looping back to step 2 will increase the quality of the subsample. 5. Using the residual of the final "partial" model, determine the control signals. 6. Fit the full control model, using control signals and data. This example is a chaotic system, so the individual trajectory will never be well reconstructed. Rather, the goal is to reconstruct the attractor.

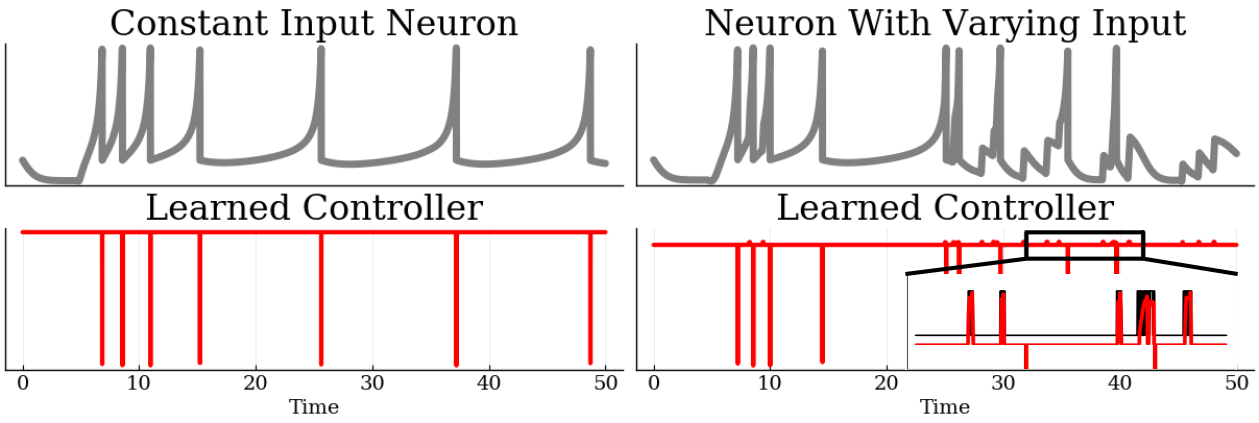

Figure 3.11: a) Voltage data from a spiking neuron model. The membrane recovery variable (u) is not shown, but is provided to the algorithm. b) The "control signal" in this case is then the fast nonlinearity, instead of a truly external input. The location is learned very accurately but because it is modeled as a true discontinuity in the equations, the exact amplitude of the derivative will depend sensitively on the sampling rate and the exact method used to numerically differentiate. c) However, if a varying input current is also applied then this will show up as an additional control signal. d) The reset nonlinearity and and external voltage are learned as a single control signal. Importantly, the learned control signals are of very different orders of magnitude.

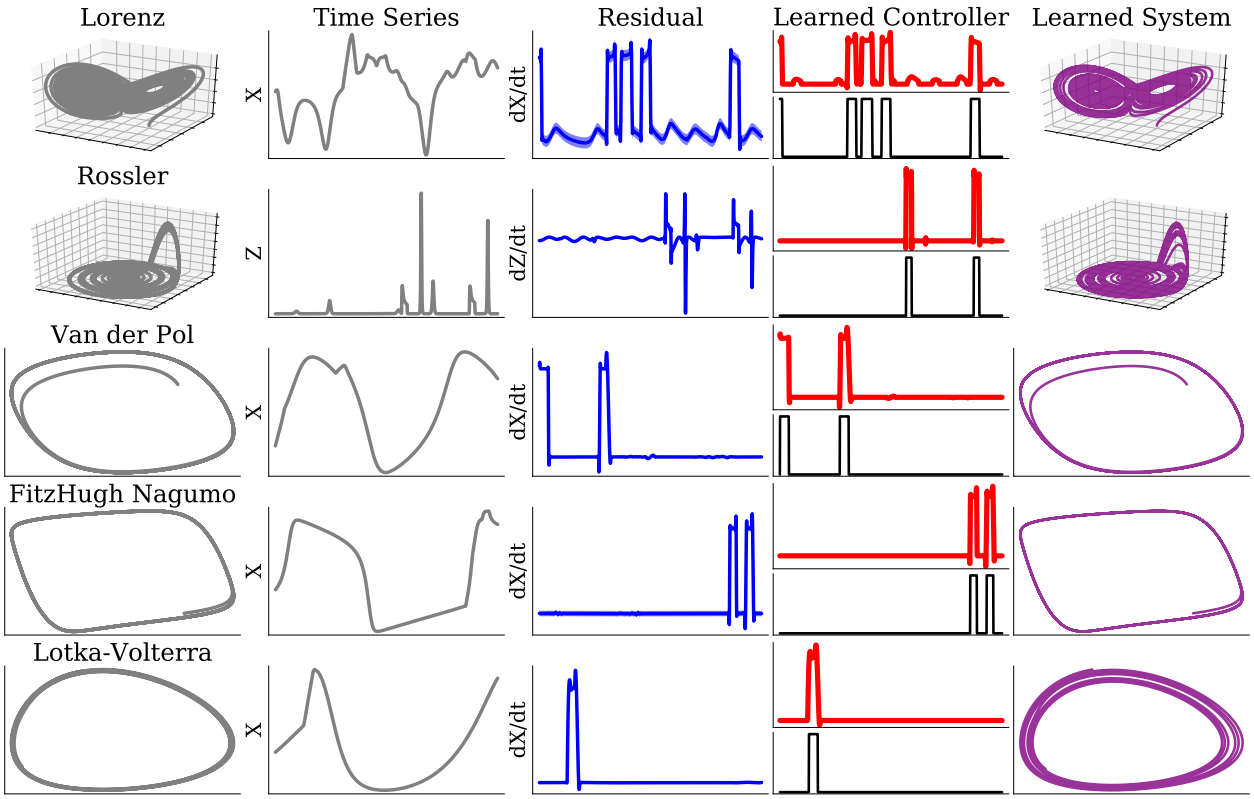

Figure 3.12: The algorithm works for a wide variety of classic models.

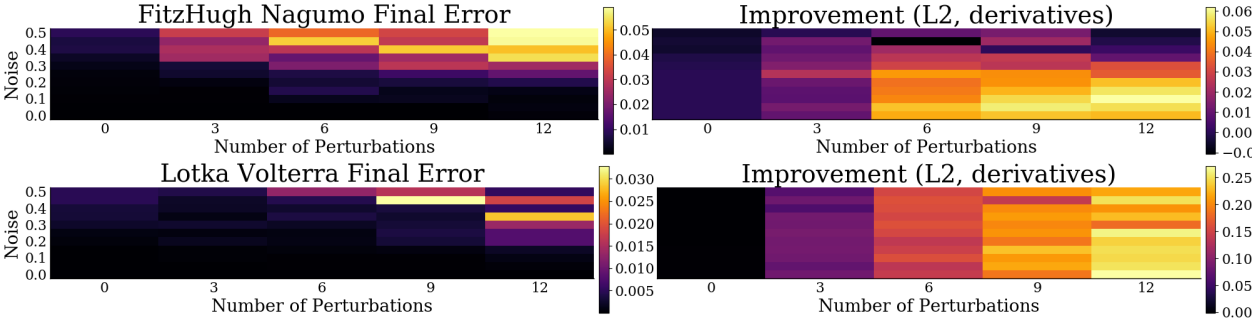

Figure 3.13: As noise increases, the performance of both this algorithm and the original SINDy algorithm can deteriorate. As the number of perturbations increase, this algorithm becomes progressively more important.

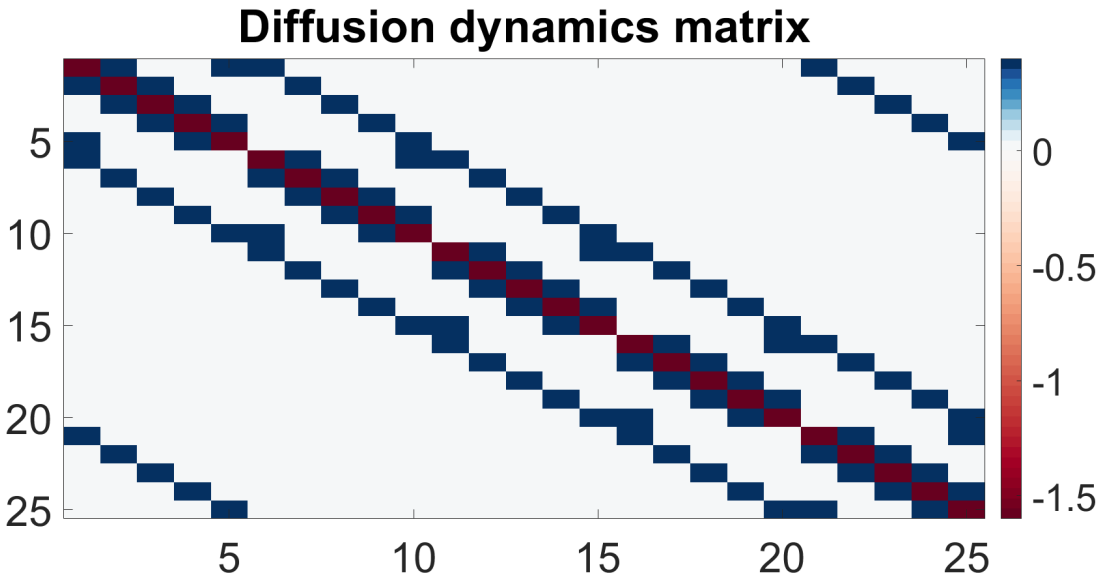

Figure 3.14: The true spatially-discretized matrix propagator that describes diffusive dynamics.

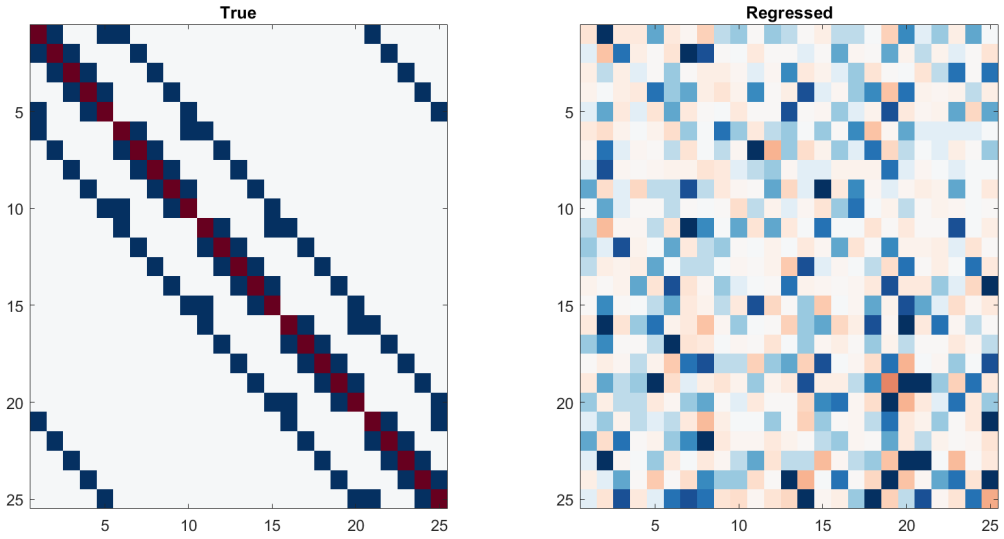

Figure 3.15: The true and best L2 fit spatially-discretized matrix propagator that describes diffusive dynamics.

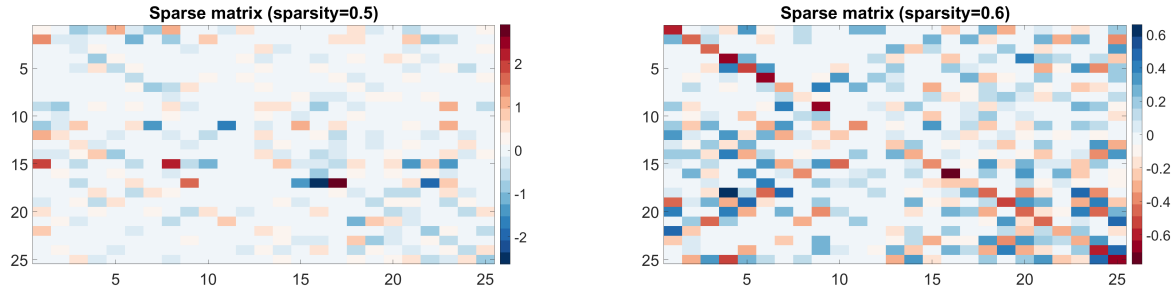

Figure 3.16: Two matrices produced for different sparsity levels by sequential least squares thresholding, both failing to recover the true structure.

Figure 3.17: A matrix produced by the LASSO algorithm for various values of λ.

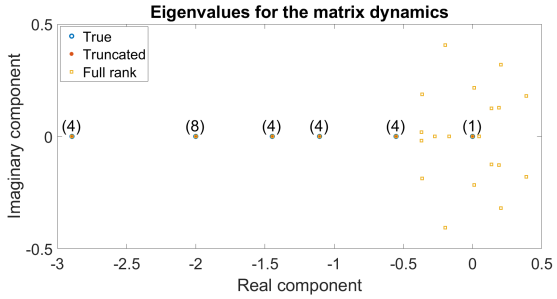

Figure 3.18: The eigenvalues of the true diffusion dynamics, along with their multiplicity. Also shown are the found eigenvalues of the best L2 fit and a rank-truncated fit.

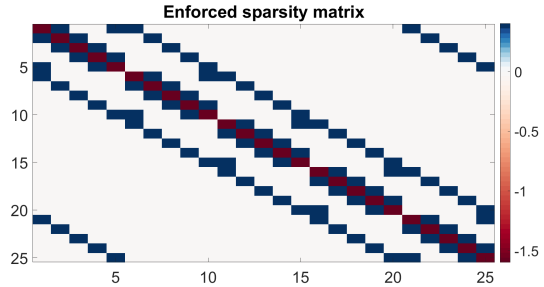

Figure 3.19: When the adjacency matrix is known and enforced, then simple L2 fitting will produce the correct signs and weights

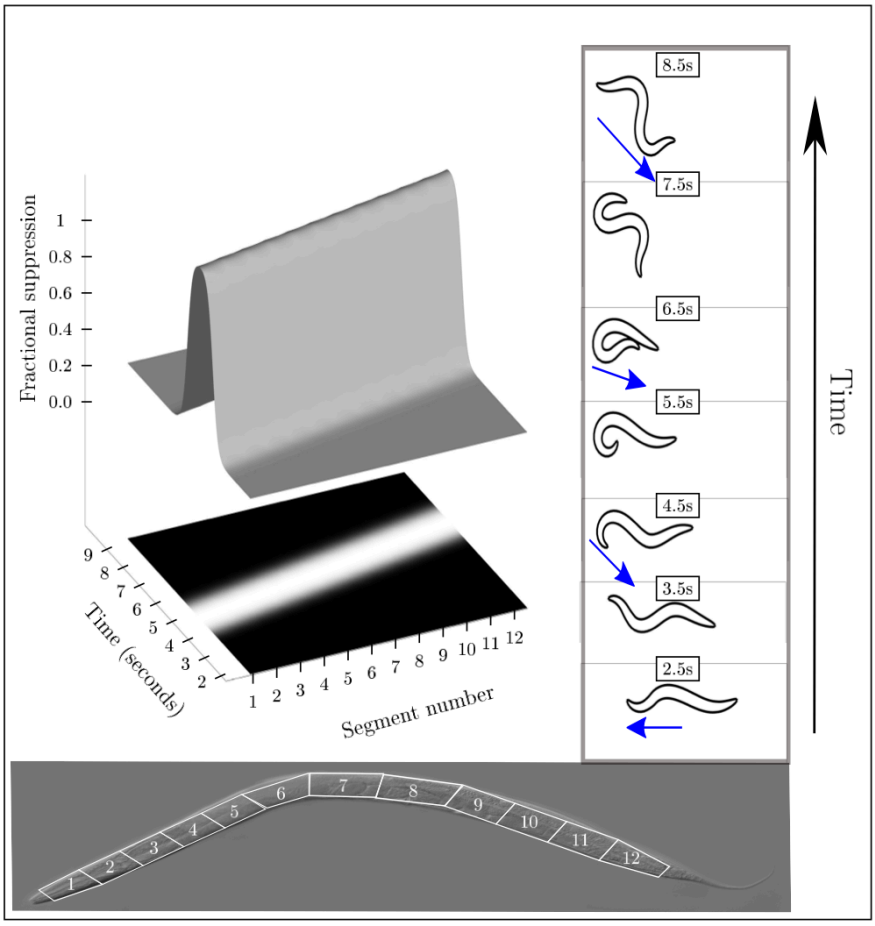

Figure 4.1: A wave of suppression on the stretch receptors produces an omega turn. The percent suppressed is shown, which travels along lasting approximately 5 seconds. The snapshots on the right are at the same times as the bold cross sections of the figure on the left.

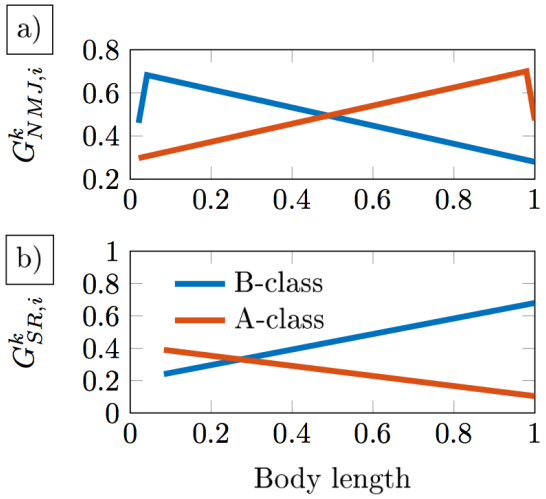

Figure 4.2: Asymmetries needed to produce backwards and forward motion. a) The nuero-muscular junctions (NMJs) decrease in strength as you travel posteriorly (anteriorly) along the body for forward (backward) motion and B- (A-) class neurons. The head (tail) is weakened in the original model in order to produce straight forward motion, and there is recent experimental evidence that the head circuit is fine tuned in a similar way [258]. b) Partially to compensate for the decrease in NMJ strength, the stretch receptor sensitivity is increased as you travel posteriorly (anteriorly) along the body for forward (backward) motion.

BIOLOGICAL RESULTS

4.1 Omega turns in a biomechanical model

We show our model can produce omega turns and other known behaviors within the model using the dynamics of proprioception.

4.1.1 Backwards motion

There are three main front-to-back asymmetries that bias the original model towards forwards motion. Two of them are shown in Fig. 4.2. For forward motion they are: a decrease in muscle strength along the length of the body, and an increase in stretch sensitivity to partially compensate for this.

Importantly, the A- and B-class subnetworks of neurons have "arms" extending in opposite

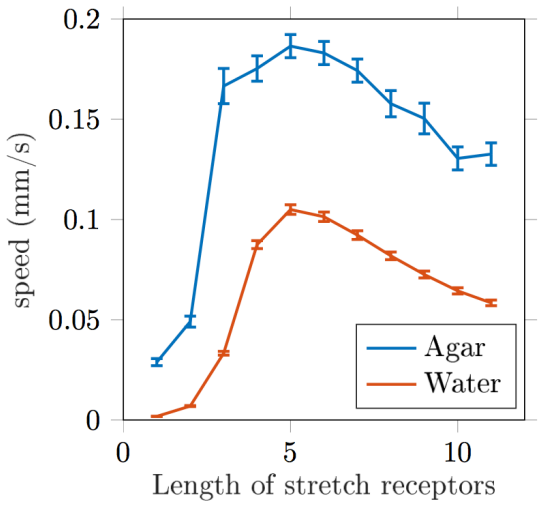

Figure 4.3: Average Center Of Mass velocity for regular forward motion as a function of the length of the stretch receptors, measured in body segments.

directions down the length of the body [294], and this is modeled as a stretch producing a signal in the anterior body segments for the A neurons. In this way, backwards motion can be plausibly and simply modeled using a mirrored subnetwork of motor neurons with oppositely aligned stretch receptors.

4.1.2 Optimized stretch receptors

Proprioceptive receptors have long been postulated, but though there are very suggestive experiments [293, 179, 4], it is not known through what mechanism the worm senses stretching. Physiological data reveals the presence of long undifferentiated "arms" that stretch away from the B- and A-class motor neurons for approximately a quarter of the body length, but their function is unknown. In order to contribute to constraints on this hypothesis, which is necessary for both simple and complex behaviors in this model, studies on the effect of changing the length of the body receptors on speed of forward motion and other metrics were performed. Figure 4.3 shows that there is a maximum center of mass velocity for proprioceptive sensors of length equal to approximately 5-6 segments, which would allow each segment

to receive information about one half wavelength in agar. The more pronounced feature of figure 4.3 is the drastic decrease in speed for very short stretch receptors of approximately 1-2 segment lengths.

4.1.3 Traveling waves of suppression produce omega turns

The addition of a set of parameters that controls a wave of suppression on the stretch receptors along the body wall is enough to realistically produce an omega turn in the model. Figure 4.1 shows a ventral turn, though this mechanism can produce turns to either side.

In order to understand this behavior, it is instructive to understand what happens to the different body segments as the wave passes through them. When the wave is initialized in the head segment (segment 1), the head becomes less able to sense the curvature of the segments immediately posterior. As the wave travels across the next few segments, the first third of the body continues to tighten its turn because the proprioceptive input is no longer present to stimulate the dorsal muscles, those opposite to those currently active. This tightening continues until the wave passes and the first segments slowly become able to sense the extreme curvature of the first half of the body. The head then starts to unwind, producing a smooth and, depending on the exact parameters, complete change in direction of motion. By tuning the timescale of this wave, the mechanism is able to produce turns of any amplitude.

The continued suppression of the stretch receptors on the posterior half of the body once the head has started to resume normal forward motion is vital to the success of this maneuver. In a realistic omega turn the rest of the body follows the head through the highly curved "omega" shape. This deep bending is resisted by any part of the body in which the proprioceptive signals remain unchanged, something observed in our model (data not shown). Non-traveling suppression of stretch receptors in one part of the body produces various types of thrashing behavior, paralysis, or changes in gait. Furthermore, the forward momentum through the curving head bend must be produced by the continued undulations of the posterior half of the body. Thus, if this proprioceptive hypothesis is correct, a traveling

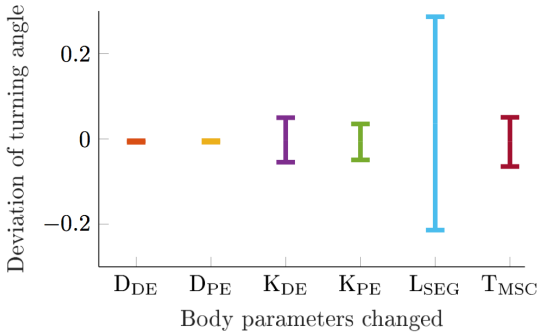

Figure 4.4: Angle change as a function of various body parameters. Displayed is the mean and standard deviation when the simulation is run for individuals with up to 10% variation in these parameters: D_*= damping coefficient; k_*= spring constant; $*_{PE}$= horizontal (passive) element; $*_{DE}$= diagonal element; L_{SEG}= segment length; T_{MSC}= muscle time constant.

suppression of signals from the stretch receptors is qualitatively necessary.

4.1.4 Robustness of omega turns

As discussed in [71], a model of a complex system that uses average values of parameters, as this model does, often gives non-average results. Therefore, it is paramount to demonstrate this behavior in an ensemble of models with different internal parameters, and a sensitivity analysis is shown in Fig. 4.4. Though the model is relatively sensitive to overall body length, even in this case the turning angle only changed about 0.2 radians, thus suggesting the mechanism is robust in producing omega turns.

4.2 Neuron population activity as a controlled dynamical system

4.2.1 Context

In this work, we exploit emerging whole-brain imaging recordings to posit a data-driven model of neurosensory integration in *C. elegans*, showing that a global linear framework, with the addition of internally generated control signals, explains and reproduces much of the activity of the network.

It has long been observed that *C. elegans* produces a small number of stable discrete behaviors (e.g. forward and backward motion, and turns), and that these behaviors change both spontaneously [236] and very quickly in response to external stimuli [310, 57, 69] or stimulation of even a single neuron [157, 178, 212]. A potential dynamical systems explanation for this observation is that of discrete behaviors as fixed points on an underlying manifold with some transition signals that move the system between them. A purely linear dynamical system of the form $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k$, cannot produce the observed multiple fixed points, where $\mathbf{x}_k$ is the data at time point k and the matrix $\mathbf{A}$ maps the state one step into the future . However, piecewise methods, like switching (hybrid) linear dynamical systems [183, 184, 75, 185], circumvent this by segmenting the dynamics into patches with different dynamics (and thus different fixed points) in each patch. An alternate method uses different phase loops and the phase along them to predict behavior, producing conserved dynamics in a special phase space [40]. Recent efforts have also attempted to explicitly model neuronal and synaptic dynamics to approximate biophysical models of the nervous system [166, 211, 76, 140, 111, 99, 38], but this has currently been limited to subsets of neurons and has moreover had difficulty capturing multiple behaviors. This work instead focuses on how a single, global, dynamical system model with simple and interpretable additions can capture the nonlinear dynamics via appropriate framing as a control problem.

The recent availability of real-time calcium imaging data allows for a neuron-level, data-driven approach. The goals of a full model of *C. elegans* neural activity are to describe how multiple states are produced in a single network, and how dynamics operating at multiple scales are integrated to produce the states and transitions between them. Our work mathematically frames this biological problem in the context of control theory by learning both the dynamics and its encoding. The dynamics portion is implemented using the recently developed data-driven method of *Dynamic Mode Decomposition with control* (DMDc) [171, 227], demonstrating that additional nonlinearities are not needed to describe many of the interactions in the system. We additionally extend this method to handle unsupervised learning of control signals as described in the Methods Section, allowing generalizability to

new complex systems and testable hypotheses in this system as discussed in the Results Section. The encoding portion is implemented using sparse variable selection methods and the novel elimination pathway of the encoding (see Methods), revealing the timescales and locations where these transition signals are encoded (see Results and Supplementary Material). We provide code written in MATLAB [102] for a full analysis pipeline that uses raw data and, if available, external behavioral labels to discover both the intrinsic dynamics, the effects of control on the state of the system, and the encoding of the control signals.

A global, linear system with control reconstructs entire time series

The manifold observed in *C. elegans* neural dynamics cannot be described by a purely linear model due to the presence of multiple stable global behaviors, as shown in Fig. 4.5.b. Specifically, linear models can only admit a single fixed state. However, the majority of neurons can be reconstructed using our *controlled, global, linear* dynamical system due to the sparse transition signals as shown in Fig. 4.5.c for expert hand-labeled signals and Fig. 4.5.d for signals learned from data. Each time snapshot of this data is reconstructed analogously to equation 2.13, and then projected onto the two dominant PCA modes of the original data so that each panel in Fig. 4.5.a-d is in the same coordinate space. Because this is a global linear model that uses a single framework for the entire state space, the need for additional nonlinear modeling can be constrained to particular groups of neurons and well-defined time windows.

In particular, across individuals the reversal class of neurons is captured very well by the supervised control signal as shown in Fig. 4.5.j and thus, up to encoding the transition signal itself, the relevant subnetwork does not appear to require nonlinearities. This means that future efforts related to nonlinear modeling may be most productive if they concentrate on the small window of time during the onset of the behavior, instead of the entire neural trace where linear models are sufficient. In addition, the type of nonlinearity required to more fully model this class of neurons is characterized: fast and short-lived spike-like activations.

Turns are also largely captured, as shown by the high correlation for the light and dark

blue boxplots in Fig. 4.5.j. The neurons involved in turning have a large number of smaller events, as shown in the SMDDL reconstruction Fig. 4.5.f; these do not lead to one of the four state transitions identified by experimentalists in this dataset, but may correspond to recently described fast time-scale states [150]. However, the unsupervised method does pick up on these smaller events and reconstructs them well as in Fig. 4.5.h, but over all datasets there is much more variability as shown in Fig. 4.5.j.

The last group of neurons, those related to forward motion, has a very large variability of correlation between the data and reconstructions. This implies that these neurons require nontrivial nonlinearities throughout the time series, not just at the onset, for full reconstruction. Although it is well known that different, dedicated subnetworks of neurons are active in forward and backwards motion [307], the functional implications have not been fully modeled. Some recent experimental work [153] characterizes an asymmetry between Forward and Reversal states as due to intrinsic bias towards the Forward state. In addition and unlike the Reversal-active neurons that require only a simple "on" transition signal, the Forward-active neurons may be continuously parametrized by speed, or contain additional behaviors like steering [132], tracking [194], or head casting [150]. Moreover, Kaplan et al. show that many neurons exhibit diverse faster time-scale fluctuations particularly during forward states [150]. Our work is consistent with these experimental results, and adds that this complexity is functionally different than that of the Reversal or Turn states.

To further characterize the effects of the control signals on the ability of this framework to capture the neural dynamics, partial models were created with a subset of control signals. Partial models using cumulative subsets of the expert-labeled control signals are shown in Fig. 4.5.k. Adding Reversal-onset signals alone does not produce a model that captures the data better than a straight-line fit to the data, but the combination of Reversal and Turning signals is significantly better. However, subsequent addition of Forward control signals is, remarkably, useless. In summary, there are several related functional observations that further work may connect to physical differences in the Reversal and Forward neuronal subnetworks: the lack of discovery of sparse Forward onset signals, which is corroborated by

the ineffectiveness of the experimentally known onset times, and the poor reconstruction of Forward-related neurons using linear dynamics.

4.3 Learning control signals: Optimization

Known transitions are discovered and characterized

Experimentalists have long separated behavior into discrete categories through careful study of individual neurons. However, open questions remain about the number of behaviors that exist and how discrete they are. Some works have posited up to six forward motion states and three reversal states, multiple turning subtypes, or even a continuum of behaviors [274, 187]. As Fig. 4.6 shows, using unsupervised optimization three behavioral onsets can be discovered: Reversal, and Dorsal and Ventral turns. In particular, the single Reversal onset signal for each individual suggests that this transition is fundamentally the same within individuals, with variability produced by activation amplitude but not a different direction in neuron-space.

However, in no individuals could a signal correlated to the onset of Forward motion be discovered. If this were a trivial state that displays no activity, a simple decay to a fixed point following a Turn state would be sufficient to achieve a good reconstruction the trajectory, even without an onset signal. However, fast-scale behaviors are known experimentally to occur within this state [150], and this complexity is reflected in the poorer reconstruction quality as shown in Fig. 4.5.j. Thus, although there is activity, its onset falls outside the sparsity assumptions of Eq. A.1. Taken together, our results imply that for the onset of these behaviors, forward motion is more complex than reversals, meaning that it cannot be described as a simple "on" signal. It is known that reversals and forward motion largely activate disjoint subnetworks of neurons [307], but the effects of this separation remain unclear.

4.4 Neural encoding of control signals

Transitions are encoded in previously unidentified neurons

Having shown the control signals to contribute significantly to the reconstruction of the data, we reconstruct the control signals themselves using time-delay data matrices and sparse linear models as shown in step 3 of Fig. 2.4 according to equation 2.11. As described in [214, 152], each of the four interpretable transition signals shown in figure 4.6 are hand-labeled using the activity of certain well-known neurons. Thus, it is not surprising that these signals can be reconstructed from data when those well-known neurons are included. In particular, as they were used to define the Dorsal Turn behavioral states, like SMDVL/R which define Ventral Turns [150], an excellent validation is that the SMDDL/R and SMDVL/R pairs of left/right neurons consistently encodes this control signal, as Fig. 4.7.a shows.

However, as the elimination path is explored further, it is revealed that these well-known neurons can be eliminated from the sparse models and the transition signals can still be reconstructed as shown in 4.7.b. Indeed, Fig. 4.7.a and and 4.7.b look nearly identical, and Fig. 4.7.c quantifies this using the percentage of false positives and negatives. Fig. 4.7.c also shows more of the elimination path and when the reconstructions eventually break down. Fig. 4.7.d and 4.7.e show the how $\mathbf{K}$ matrices in equation 2.11 change as neurons are removed. Taken together, these results reveal previously unidentified neurons that can successfully predict control signals shown to be important to reconstructing the full neural manifold. However, only rows with names are neurons that have been connected to the stereotyped *C. elegans* connectome and can thus be identified across individuals; rows with numbers cannot be so compared. In summary, this work identifies sets of neurons previously implicated in transitions and also reveals new candidates for critical actuators of network transitions. In addition, the time of encoding is revealed, which can inform further study.

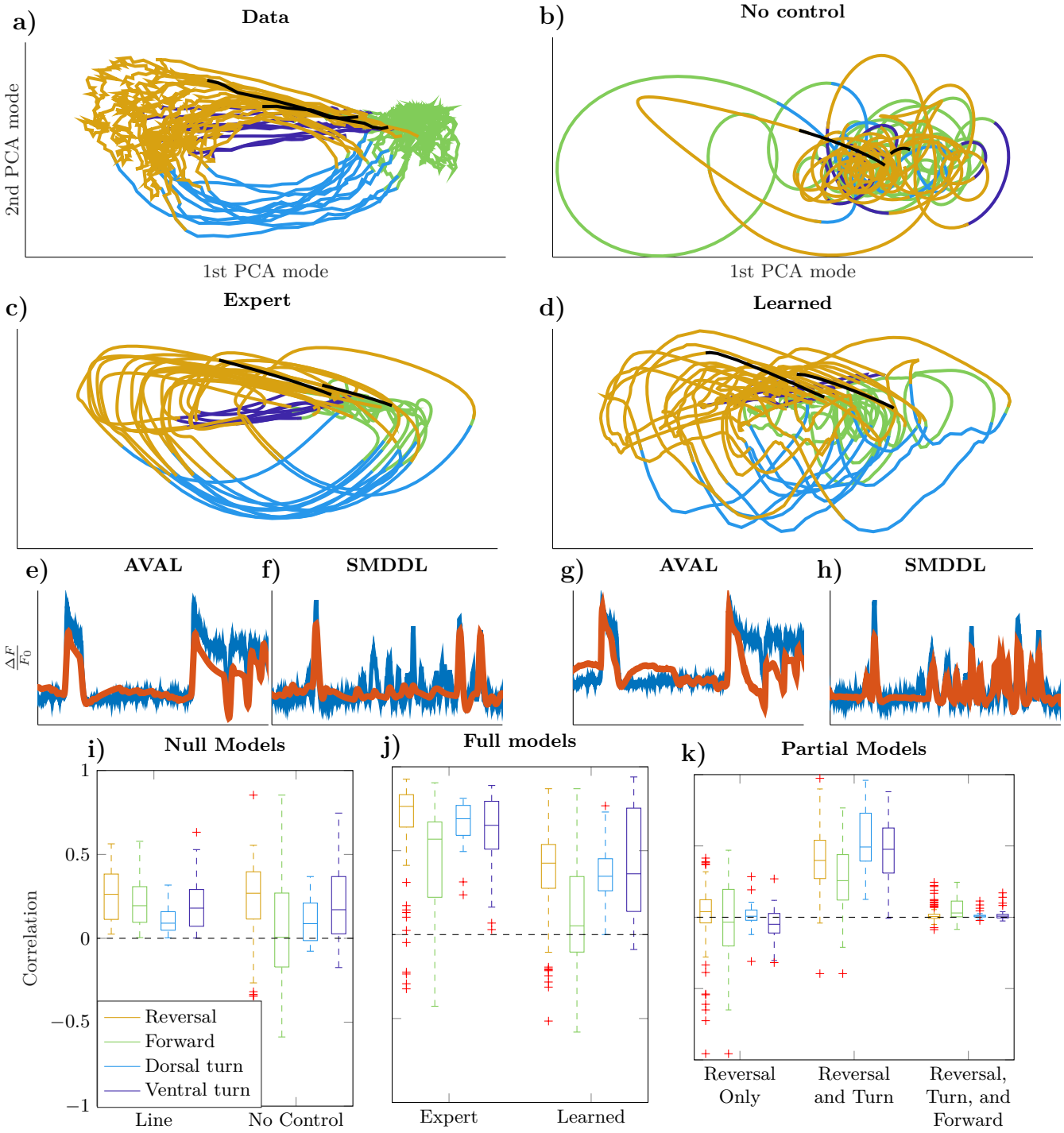

Figure 4.5: 2d PCA projections of a) data, b) an uncontrolled "null" model, c) a "supervised" model using expert-determined control signals, and d) an "unsupervised" model that uses control signals learned via algorithm 3. The governing equations matrices in are all learned from data, either uncontrolled (b, Equation 2.3) or controlled (c-d, Eq. 2.12). These data are color-coded by state: Black for Unknown, Yellow for Reversal neurons, Green for Forward, and Light (Dark) Blue for Dorsal (Ventral) turns. e-f) Example neuron datasets with reconstructions from the supervised model. A reversal-active (AVAL) and a Dorsal-Turn-active neuron (SMDDL) are shown. g-h) The same neurons shown with reconstructions from the unsupervised model. i-k) Correlations across datasets between data and reconstructions,

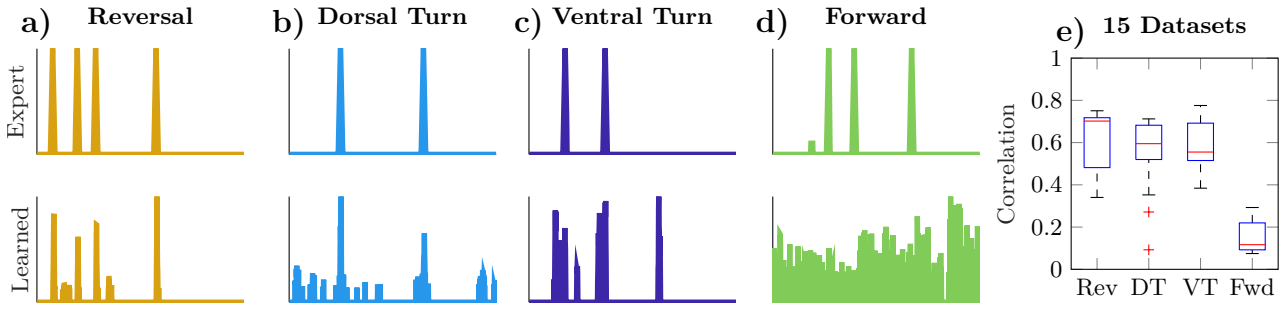

Figure 4.6: Control signals can be learned from data via algorithm 3. a-d) The onset of well-known states as determined by experts (above) and as learned (below). All signals are normalized to have a maximum of 1.0. e) Correlation between expert and learned signals across 15 individual datasets. Reversals (Rev), Dorsal (DT) and Ventral Turns (VT) are consistently learned, but Forward state (Fwd) onsets are never significant.

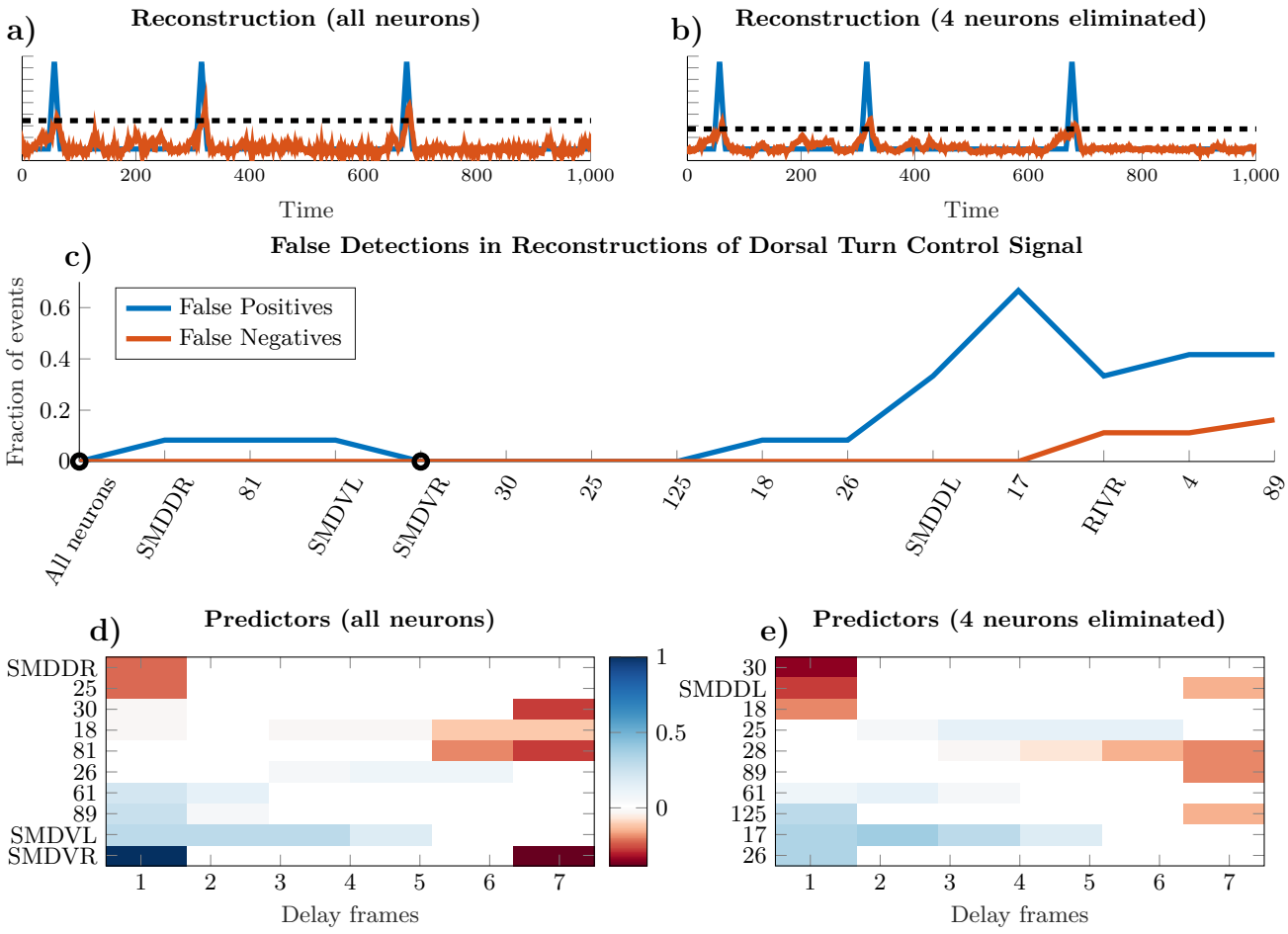

Figure 4.7: a-b) Control signal reconstructions via linear encoding on the data including time delays, with all neurons (a) or 4 neurons removed (b). The removed neurons are: SMDDR, 81, SMDVL, and SMDVR as shown on the x axis of (c). Event detection is determined via a hard threshold for each signal (dotted line). See supplement for more discussion of this threshold. c) Neurons are eliminated in order of the largest magnitude given to them by the linear model. The number of false detections increases significantly only after 8 neurons have been removed. d-e) The weights given to the top 10 most important neurons for different iterations.

Chapter 5

DISCUSSION

This thesis has described advances in both first-principles and data-driven modeling with the goal of understanding the engineering and dynamics of neuronal networks, especially in the model organism *C. elegans*. Each of these modeling paradigms has several possible areas of extension and application.

5.1 Overview: "Complexity-scale" Separation

The organization of brains is clearly not random, and yet it is unclear how structure facilitates, stabilizes, and limits the functions that an animal performs. The overarching theme of this thesis is that progress can be made by performing a sort of scale separation, akin to the slow-fast scale separation techniques common in the analysis of physical systems [120]. This perspective proposes that a seemingly complex repertoire of behaviors can be desribed as follows: simple default state or behavior of a system that can be perturbed by a *control signal* to cause a transient, and possibly more complex, behavior.

The following sections place the results of this thesis within this context. First, the biomechanical modeling perspective is discussed, in which a well known model in the literature designed to produce a default state of forward motion was augmented with a control signal for omega turns. Second, extensions of two data driven methods are described, one that is a extension of DMD, and two for use in natural systems where the control signals are unknown. In other words, the proper "complexity separation" is not known, and must be discovered from data. Third, the application of these techniques to a real system is described, as well as future directions for generalization

5.2 Biomechanical Modeling

We have presented a biophysically realistic model that can reproduce the essential repertoire of *C. elegans* locomotive behaviors: forward motion, backward motion, head sweeps and omega turns. Recent work has shown that extra-synaptic modulation is important in more complex behavioral patterns in *C. elegans* [87], and this is the first biomechanical model to our knowledge that uses this information and proposes an interpretable mechanism for behaviors beyond forward motion.

This model relies fundamentally on the hypothesis that there exist stretch receptors in *C. elegans*, and that they are posteriorly (anteriorly) directed for B(A)-class neurons. The model further allows for testable predictions about the characteristics of those receptors. Specifically, there must be some type of suppression of the signal sent by those receptors for deep turning behaviors like the omega turn to exist. Mutant studies should be able to identify potential chemical or neural control candidates, which in turn might help illuminate the network involved in proprioception. Another class of experimental tests is through optogenetic manipulation of worms trapped in microfluidic devices along the lines of Quen et al. [293]. If neurons associated with omega turns, e.g. RIV, SMDV, or RIM, are stimulated and a transient down-regulation of the proprioceptive signal is measured, that would be strong evidence for this mechanism.

A traveling wave of modulation, specifically suppression, on the proprioceptive stretch receptors is required to robustly produce omega turns, but we also found that an increase in muscle strength can drastically improve the turning angles of the worm. Though it is possible that the muscle activation is modulated during this behavior, quantitative statements are complicated by many approximations. These issues are discussed in more depth in the original model paper [38]. Thus, unlike the qualitative proposal of a modulatory mechanism on the stretch receptors, no strong quantitative statement can be made about the muscle dynamics themselves.

A general goal of *C. elegans* research is to understand how sensory information is trans-

muted into behaviors that exist on many different time scales. One step towards understanding this process is through finding the control mechanisms of the neural network that physically cause those behaviors. The ability of a single model to produce all of these behaviors helps elucidate the control structure of *C. elegans*, and can help inform what types of outputs must be produced by the internal dynamics of the network [163]. Omega turns and the type of modulation required to produce them were a critical contribution of this paper. The posited mechanism can also produce much shallower or deeper turns, both of which have been proposed as distinct categories of behavior [156, 41].

Much recent modeling work has contributed to discussions surrounding simple behaviors like forward motion, and we hope that this paper can contribute to similar discussions of more complex behavioral dynamics involving omega turns. Importantly, we have illustrated that dynamic processes can play a critical role in controlling the *C. elegans* repertoire of behavior. In future work, we hope to integrate the connectomic dynamics with the proposed biomechanial model in order to understand the "inside" and "outside" of the worm and how the connectome serves as the controller for behavioral outputs.

5.3 Data-Driven Methods

5.3.1 Infinite Series DMD

We have proposed a variation of the Dynamic Mode Decomposition algorithm that penalizes higher-order terms for improved accuracy and stability. Control can be incorporated, producing the most accurate and robust algorithm for solving the DMDc problem.

One advantage of Schmid's original DMD algorithm [248] is that the full matrix $\mathbf{A}$ was not explicitly calculated, and instead the eigenvalues and eigenvectors ("DMD modes") were calculated directly. In other words, Schmid solved the underlying optimization problem from the "mode perspective," whereas we have solved it from the "propagator" perspective. The uncontrolled DMD problem admits a closed-form formula for forecasting into the future using these eigenvalues and DMD modes, as opposed to the "propagator" perspective which

forecasts into the future via direct matrix multiplication as in equation 2.3. However, once control is introduced into the system, there is no closed formula using the DMD modes, and matrix multiplication must be used as in equation 2.12. For some problems where the $\mathbf{A}$ matrix cannot be represented in memory, for example in extremely high dimensional fluid mechanics problems, iDMD cannot be used.

The two series presented here are not unique in their usefulness, and other infinite or finite series variations are possible. In particular, there is a close form solution for the finite sum of a geometric series. However, the exact solution for the finite sum does not allow terms to be collected when control is added, and thus the infinite series approximation is key for that application. As the controlled case is the most significant contribution of this work, the infinite case was most emphasized.

In real-world data sets, it is often known that linear dynamics are only appropriate for a very short time-scale approximation of underlying nonlinearities. An additional consideration for the choice of truncation parameters comes into play, as at different time scales the approximate linearized propagator may be different or cease to exist. Similar difficulties may occur when the underlying system is stochastic (i.e. has process noise), because there is no propagator that can deterministically map the current time forward. Although this time horizon is rarely known in such complex situations, a short truncation is likely to be helpful for many of these cases.

DMD and DMDc are powerful data-driven methods that have been used in many real-world applications. We hope that our more robust and accurate variation allows DMDc in particular to be more widely useful.

5.3.2 Learning control signals: subsampling

Many data-driven modeling techniques posit that the dynamics present are largely predictable from the past history of the data, and do not undergo disturbances outside the modeling framework. However, this assumption may be violated in many ways, two of which have been treated here: connections to the outside environment, and very fast nonlinearities

or discontinuities. Our unsupervised modeling framework explicitly accounts for these external inputs, and successfully models the underlying intrinsic dynamics by pulling out the control signals or fast nonlinearities.

There are several assumptions that are necessary for good numerical performance. For the examples studied in this paper the intrinsic dynamics could be modeled by the imposed uncontrolled model, and in particular for the nonlinear examples, the true dynamics were sparse in the measured basis. This issue of finding the correct basis is an active field of research and beyond the scope of this paper [60]. A less well studied issue is that of the types of control signals that can be successfully separated out from "intrinsic" dynamics. This paper dealt with smooth intrinsic dynamics and effectively discontinuous control signals, which were thus separable because they could not be modeled using the set of library functions proposed in the SINDy algorithm. However, the other extreme is common, that of controllers that are of similar functional forms to the intrinsic dynamics themselves. For example, a popular controller in engineering and potentially in many biological systems is a PID controller, which uses terms proportional to simple functions of the state in order to acheive control. This is by design continuous, and in many cases the controlled system can be written as the uncontrolled system with a change of parameters. This algorithm would not separate out the intrinsic dynamics, because it assumes that the structure and parameter values are unknown and must be discovered along with the control signal. However, if there is domain knowledge of the intrinsic dynamics our algorithm may be extendable to cases with smooth controllers; this is outside the scope of this work.

A separate limitation of this work is that a fully generative model is not produced. That is, extrapolation beyond the training data can only be achieved for the uncontrolled, intrinsic dynamics. Of course, if the control signal is truly external then extrapolation that requires such knowledge is not possible. However, if the "control signal" is actually a function of phase space, for example the ground in the bouncing ball example and the reset discontinuity in the neuron example, a generative model is possible but is not produced. A promising area future work is to separate out which portions of the learned control signal can be modeled

as some function of the data, and which cannot.

A universal difficulty is that of white noise, which is magnified in particular by two elements of the algorithm. First, the need to take a numerical derivative, which is known to be very sensitive to measurement noise. This is a well studied problem , but there are no universal answers. Second, because the residual of a naive model is an object of study, it is vulnerable to large fluctuations. Recent work on simultaneous denoising and derivative calculations could be a fruitful area of future work [243].

Our algorithm adds to the landscape of data-driven algorithms that derive symbolic governing equations and extends the range of applicability into new problem domains.

5.3.3 Learning Control Signals: Sparse Optimization

This is the second novel approach for learning control signals from data. Specifically, this approach is designed as an extension of DMDc

We have presented two distinct strategies for learning control signals from data.

5.4 Scale separation in C. elegans

We have presented the first data-driven model that uses a single set of intrinsic dynamics that can reconstruct the multiple behavioral regimes present in a real animal and transitions between them. In this study, we have analyzed neuronal recordings from C. elegans that lacked any acutely delivered and time-varying sensory inputs, therefore behavioral transitions are likely internally driven [152] and governed by stochastic processes [236]. Here, control signals cannot be inferred simply from sensory neuron activity. To overcome this challenge, we provide an unsupervised approach for identifying such internally driven control signals and their underlying neuronal identities. The fact that this controlled linear model accurately reproduces both short and long time-scale dynamics places clear restrictions on the need, specifically the lack thereof, for nonlinearities in this system, and provides hypotheses about the neurons that may contain those nonlinearities as well as their role in the global dynamics of the system. In addition, we have embedded this model in a math-

ematical framework of control, which can be generalized to other organisms or to include hypothesized nonlinearities.

Much excitement has been generated by the availability of the *C. elegans* anatomical connectome, and one aim of data-driven modeling efforts is to produce a functional connectome that can complement the anatomical data. The DMDc algorithm in this paper is similar to several algorithms in the engineering literature that attempt similar network reconstruction tasks, namely System Identification [191]. One strategy to fully disambiguate the effects of the intrinsic dynamics and the external control signals uses known external perturbations should be applied and the system response measured. Such perturbations are not generally available in biological systems and thus the data collected are "uninformative" [5] in the sense that the underlying structure cannot be determined. Thus, although this is a promising avenue for future work, it is outside the scope of this paper.

A limitation of this model is that it is not generative; it cannot be used to predict a system response that includes transitions to novel stimuli. To accomplish this, the transition signals must be written as a function of the data. Step three of our method does this with a linear encoding and demonstrates that the signals can be successfully reconstructed with all neurons to a certain level of accuracy. If this level of accuracy were sufficient, then the system would be fully linear and an uncontrolled model would produce a good reconstruction, as is clearly not the case. Recent methods for incorporating nonlinearities into controlled systems (e.g. [49, 296]) have the potential to create a fully closed-loop feedback system and this is an active area of further research [210].

This methodology uses two key hypotheses about this system. First, although the true biological system includes many thousands of nonlinear interactions, we assume that the leading order or dominant balance dynamics of the dynamical system within certain regimes is simple. Recent work in the fluids community [27] has shown that even when the full global model is perfectly known, almost every region of phase space is well described by a simpler model with fewer or no nonlinear terms. In the same way, this paper posits that a single set of linear dynamics capture the dominant activity of large regions of phase space.

The second observation is of a time-scale separation between activity within a state, and transitions between states. This leads to a hypothesis about when complex nonlinearities may be active: sparsely, during transitions between behaviors. Thus, this methodology is directly applicable when trajectories follow simplified, linear dynamics produced by intrinsic neural dynamics in large regions of phase space, with short periods of complex dynamics. An area of future work is to explore how these control signals could be produced biologically, and a strength of this method is that the control signals may be well modeled by an intermediate spike-type thresholding of a more complex signal.

A potential criticism of this method is that we have used discrete labeled states in our model, despite ongoing debate regarding how uniform "states" in *C. elegans* are across instances, and if they should be subdivided or are simply continuous [274, 187, 130]. We have contributed to this debate by providing evidence that the reversal and turn states in fact appear to be simple and have well-defined initiation signals. However, the forward "state" is much more complex, and breaks the assumptions of our model. Specifically, the intrinsic dynamics may be different in the forward state as compared to the rest of the phase space, and may be a different linear system as posited in [185] or nonlinear as in [210]. Related, the "transition" into this state may not follow the sparsity assumptions of Algorithm 3, perhaps due to a continuum of states as opposed to a discrete transition. Moreover, Kaplan et al. [150] show that during the forward state many neurons fluctuate at faster timescales, contributing to this complexity. We argue that this is a strength of this methodology: because this is not a that can universally approximate arbitrary dynamics, the fact that a state and its transition cannot be reconstructed gives additional information about that state, and about its complexity in relation to other states. However, it is conceivable that failure to identify control signals during the Forward state, and subsequent low reconstruction quality, is simply due to lack of sufficient data. As more neurons are imaged or longer recording times become experimentally feasible, so far undiscovered control signals and neuronal candidates during the Forward state may be revealed.

An alternate approach to modeling complex systems in order to understand structure is

to use locally linear models [75, 184, 183, 185]. In this methodology, the initial network as described by the matrix $\mathbf{A_k}$ is replaced by a new matrix, $\mathbf{A_{k+1}}$, at certain change points. These have achieved great success in reconstructing nonlinear datasets, and in fact can reconstruct arbitrary dynamics given enough change points, and is an active field in machine learning research. However, it is difficult to interpret what such a replacement of the underlying dynamics would mean biologically, particularly if many separate matrices $\mathbf{A_k}$ are required. On the other hand, the language of control theory from engineering meshes directly with the biological intuition that certain states are initiated by relatively unique signals produced by a small number of neurons. We propose that our framework for constructing a single, global model of the dynamics of this neural system is promising not only in its ready generalizability to include nonlinearities, but also in its biological interpretability.

We have produced the first, to our knowledge, global data-driven model of both the intrinsic and control dynamics of *C. elegans*.

Here, we provide a framework for the identification of neurons critical in actuating network transitions, which can be tested in future experiments.

BIBLIOGRAPHY

[1] Pca wikipedia.

[2] Wormatlas website.

[3] Wormwiring.

[4] A. Albeg, C. J. Smith, M. Chatzigeorgiou, D. G. Feitelson, D. H. Hall, W. R. Schafer, D. M. Miller, and M. Treinin. C. elegans multi-dendritic sensory neurons: morphology and function. *Molecular and Cellular Neuroscience*, 46(1):308–317, 2011.

[5] Marco Tulio Angulo, Jaime A Moreno, Gabor Lippner, Albert-László Barabási, and Yang-Yu Liu. Fundamental limitations of network reconstruction from temporal data. *Journal of the Royal Society Interface*, 14(127):20160966, 2017.

[6] Paul D Arendt, Daniel W Apley, and Wei Chen. Quantification of model uncertainty: Calibration, model discrepancy, and identifiability. *Journal of Mechanical Design*, 134(10):100908, 2012.

[7] Paul D Arendt, Daniel W Apley, and Wei Chen. Quantification of model uncertainty: Calibration, model discrepancy, and identifiability. *Journal of Mechanical Design*, 134(10):100908, 2012.

[8] Paul D Arendt, Daniel W Apley, and Wei Chen. Quantification of model uncertainty: Calibration, model discrepancy, and identifiability. *Journal of Mechanical Design*, 134(10):100908, 2012.

[9] Paul D Arendt, Daniel W Apley, Wei Chen, David Lamb, and David Gorsich. Improving identifiability in model calibration using multiple responses. *Journal of Mechanical Design*, 134(10):100909, 2012.

[10] Paul D Arendt, Daniel W Apley, Wei Chen, David Lamb, and David Gorsich. Improving identifiability in model calibration using multiple responses. *Journal of Mechanical Design*, 134(10):100909, 2012.

[11] George B Arhonditsis, Gurbir Perhar, Weitao Zhang, Evangelia Massos, Molu Shi, and Argho Das. Addressing equifinality and uncertainty in eutrophication models. *Water Resources Research*, 44(1), 2008.

[12] George B Arhonditsis, Gurbir Perhar, Weitao Zhang, Evangelia Massos, Molu Shi, and Argho Das. Addressing equifinality and uncertainty in eutrophication models. *Water Resources Research*, 44(1), 2008.

[13] Travis Askham and J Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018.

[14] Travis Askham and J Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018.

[15] Travis Askham, Peng Zheng, Aleksandr Aravkin, and J Nathan Kutz. Robust and scalable methods for the dynamic mode decomposition. *arXiv preprint arXiv:1712.01883*, 2017.

[16] Travis Askham, Peng Zheng, Aleksandr Aravkin, and J Nathan Kutz. Robust and scalable methods for the dynamic mode decomposition. *arXiv preprint arXiv:1712.01883*, 2017.

[17] Karl Johan Åström and Tore Hägglund. *PID controllers: theory, design, and tuning*, volume 2. Instrument society of America Research Triangle Park, NC, 1995.

[18] Karl Johan Åström and Tore Hägglund. *PID controllers: theory, design, and tuning*, volume 2. Instrument society of America Research Triangle Park, NC, 1995.

[19] C Atkinson, GEH Reuter, and CJ Ridler-Rowe. Traveling wave solution for some nonlinear diffusion equations. *SIAM Journal on Mathematical Analysis*, 12(6):880–892, 1981.

[20] Omri Azencot, Wotao Yin, and Andrea Bertozzi. Consistent dynamic mode decomposition. *arXiv preprint arXiv:1905.09736*, 2019.

[21] M. Backholm, A. K. S. Kasper, R. D. Schulman, W. S. Ryu, and K. Dalnoki-Veress. The effects of viscosity on the undulatory swimming dynamics of c. elegans. *Phys Fluids*, 27:091901, 2015.

[22] Shervin Bagheri. Koopman mode decomposition of the cylinder wake. *Journal of Fluid Mechanics*, 726:596–623, 2013.

[23] Shervin Bagheri. Koopman-mode decomposition of the cylinder wake. *Journal of Fluid Mechanics*, 726:596–623, 2013.

[24] Shervin Bagheri. Koopman mode decomposition of the cylinder wake. *Journal of Fluid Mechanics*, 726:596–623, 2013.

[25] Cornelia I Bargmann. Beyond the connectome: how neuromodulators shape neural circuits. *Bioessays*, 34(6):458–465, 2012.

[26] Cornelia I Bargmann and H Robert Horvitz. Chemosensory neurons with overlapping functions direct chemotaxis to multiple chemicals in c. elegans. *Neuron*, 7(5):729–742, 1991.

[27] Cx K Batchelor and GK Batchelor. *An introduction to fluid dynamics*. Cambridge university press, 2000.

[28] N Benjamin Erichson, Steven L Brunton, and J Nathan Kutz. Compressed singular value decomposition for image and video processing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1880–1888, 2017.

[29] Barry Bentley, Robyn Branicky, Christopher L Barnes, Yee Lian Chew, Eviatar Yemini, Edward T Bullmore, Petra E Vértes, and William R Schafer. The multilayer connectome of caenorhabditis elegans. *PLoS computational biology*, 12(12):e1005283, 2016.

[30] S. Berri, J. H. Boyle, M. Tassieri, I. A. Hope, and N. Cohen. Forward locomotion of the nematode c. elegans is achieved through modulation of a single gait. *HFSP journal*, 3(3):186–193, 2009.

[31] A. Blau, F. Callaly, S. Cawley, A. Coffey, A. De Mauro, G. Epelde, L. Ferrara, F. Krewer, C. Liberale, P. Machado, and G. Maclair. July. In In Conference on Biomimetic and . Biohybrid Systems, editors, *The Si elegans Project–The Challenges and Prospects of Emulating Caenorhabditis elegans*, pages 436–438, International Publishing, 2014. Springer.

[32] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[33] Thomas Blumensath and Mike E Davies. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5-6):629–654, 2008.

[34] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[35] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[36] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[37] J. H. Boyle, J. Bryden, and N. Cohen. An integrated neuro-mechanical model of c. elegans forward locomotion. *Lect Notes Comput Sci*, 4984:37–47, 2008.

[38] Jordan Hylke Boyle, Stefano Berri, and Netta Cohen. Gait modulation in c. elegans: an integrated neuromechanical model. *Frontiers in computational neuroscience*, 6:10, 2012.

[39] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[40] Connor Brennan and Alexander Proekt. A quantitative model of conserved macroscopic dynamics predicts future motor commands. *eLife*, 8:e46814, 2019.

[41] O. D. et al. Broekmans. Resolving coiled shapes reveals new reorientation behaviors in c. elegans. *eLife*, 5, 2016.

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

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

[44] 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.

[45] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2):e0150171, 2016.

[46] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2):e0150171, 2016.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] J. A. Bryden and N. Cohen. Neural control of caenorhabditis elegans forward locomotion: the role of sensory feedback. *Biol Cybern*, 98:339–351, 2008.

[52] Jenny Brynjarsdottir and Anthony O'Hagan. Learning about physical parameters: The importance of model discrepancy. *Inverse problems*, 30(11):114007, 2014.

[53] Jenny Brynjarsdottir and Anthony O'Hagan. Learning about physical parameters: The importance of model discrepancy. *Inverse problems*, 30(11):114007, 2014.

[54] John A Burns, Eugene M Cliff, and Kasie Farlow. Parameter estimation and model discrepancy in control systems with delays. *IFAC Proceedings Volumes*, 47(3):11679–11684, 2014.

[55] Emmanuel J Candès et al. Compressive sampling. In *Proceedings of the international congress of mathematicians*, volume 3, pages 1433–1452. Madrid, Spain, 2006.

[56] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[57] Sreekanth H Chalasani, Nikos Chronis, Makoto Tsunozaki, Jesse M Gray, Daniel Ramot, Miriam B Goodman, and Cornelia I Bargmann. Dissecting a circuit for olfactory behaviour in caenorhabditis elegans. *Nature*, 450(7166):63, 2007.

[58] Kathleen Champion, Steven L Brunton, and J Nathan Kutz. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *arXiv preprint arXiv:1805.07411*, 2018.

[59] Kathleen Champion, Steven L Brunton, and J Nathan Kutz. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *arXiv preprint arXiv:1805.07411*, 2018.

[60] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[61] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[62] Kathleen P Champion, Steven L Brunton, and J Nathan Kutz. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1):312–333, 2019.

[63] Kathleen P Champion, Steven L Brunton, and J Nathan Kutz. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1):312–333, 2019.

[64] Kevin K Chen, Jonathan H Tu, and Clarence W Rowley. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *Journal of nonlinear science*, 22(6):887–915, 2012.

[65] Kevin K Chen, Jonathan H Tu, and Clarence W Rowley. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *Journal of nonlinear science*, 22(6):887–915, 2012.

[66] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[67] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[68] Xinfu Chen, Yuanwei Qi, and Yajing Zhang. Existence of traveling waves of autocatalytic systems with decay. *Journal of Differential Equations*, 260(11):7982–7999, 2016.

[69] Catherine M Chiba and Catharine H Rankin. A developmental analysis of spontaneous and reflexive reversals in the nematode caenorhabditis elegans. *Journal of neurobiology*, 21(4):543–554, 1990.

[70] N. Cohen and T. A Ranner. *New Computational Method for a Model of C. elegans Biomechanics: Insights into Elasticity and Locomotion Performance.* [physics.bio-ph], 2017.

[71] D. D. Cook and D. J. Robertson. The generic modeling fallacy: Average biomechanical models often produce non-average results! *Journal of Biomechanics*, 49(15):3609–3615, 2016.

[72] Steven J Cook, Travis A Jarrell, Christopher A Brittin, Yi Wang, Adam E Bloniarz, Maksim A Yakovlev, Ken CQ Nguyen, Leo T-H Tang, Emily A Bayer, Janet S Duerr, et al. Whole-animal connectomes of both caenorhabditis elegans sexes. *Nature*, 571(7763):63–71, 2019.

[73] Theodore Cornforth and Hod Lipson. Symbolic regression of multiple-time-scale dynamical systems. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 735–742. ACM, 2012.

[74] Theodore Cornforth and Hod Lipson. Symbolic regression of multiple-time-scale dynamical systems. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 735–742. ACM, 2012.

[75] Antonio C Costa, Tosif Ahamed, and Greg J Stephens. Adaptive, locally linear models of complex dynamics. *Proceedings of the National Academy of Sciences*, 116(5):1501–1510, 2019.

[76] Alicia Costalago-Meruelo, Pedro Machado, Kofi Appiah, Andoni Mujika, Peter Leskovsky, Roberto Alvarez, Gorka Epelde, and T Martin McGinnity. Emulation of chemical stimulus triggered head movement in the c. elegans nematode. *Neurocomputing*, 290:60–73, 2018.

[77] John P Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16(1):2859–2900, 2015.

[78] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010.

[79] 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.

[80] 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.

[81] Brian de Silva, David M Higdon, Steven L Brunton, and J Nathan Kutz. Discovery of physics from data: Universal laws and discrepancy models. *arXiv preprint arXiv:1906.07906*, 2019.

[82] Brian de Silva, David M Higdon, Steven L Brunton, and J Nathan Kutz. Discovery of physics from data: Universal laws and discrepancy models. *arXiv preprint arXiv:1906.07906*, 2019.

[83] Nicola Demo, Marco Tezzele, and Gianluigi Rozza. Pydmd: Python dynamic mode decomposition. *Journal of Open Source Software*, 3(22):530, 2018.

[84] Nicola Demo, Marco Tezzele, and Gianluigi Rozza. Pydmd: Python dynamic mode decomposition. *Journal of Open Source Software*, 3(22):530, 2018.

[85] X. Deng, J. X. Xu, J. Wang, G. Y. Wang, and Q. S. Chen. Biological modeling the undulatory locomotion of c. elegans using dynamic neural network approach. *Neurocomputing*, 186:207–217, 2016.

[86] Mingzhou Ding, Yonghong Chen, and Steven L Bressler. 17 granger causality: basic theory and application to neuroscience. *Handbook of time series analysis: recent theoretical developments and applications*, 437, 2006.

[87] J. L. Donnelly, C. M. Clark, A. M. Leifer, J. K. Pirri, M. Haburcak, M. M. Francis, A. D. Samuel, and M. J. Alkema. Monoaminergic orchestration of motor programs in a complex c. elegans behavior. *PLoS Biol*, 11:4, 2013.

[88] Jamie L Donnelly, Christopher M Clark, Andrew M Leifer, Jennifer K Pirri, Marian Haburcak, Michael M Francis, Aravinthan DT Samuel, and Mark J Alkema. Monoaminergic orchestration of motor programs in a complex c. elegans behavior. *PLoS biology*, 11(4):e1001529, 2013.

[89] David L Donoho. For most large underdetermined systems of linear equations the minimal ℓ1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(6):797–829, 2006.

[90] Daniel Duke, Julio Soria, and Damon Honnery. An error analysis of the dynamic mode decomposition. *Experiments in Fluids*, 52(2):529–542, 2012.

[91] Daniel Duke, Julio Soria, and Damon Honnery. An error analysis of the dynamic mode decomposition. *Experiments in Fluids*, 52(2):529–542, 2012.

[92] P. Erdös and E. Niebur. The neural basis of the locomotion of nematodes. *Lect Notes Phys*, 1990(368):253–267, 1990.

[93] N. B. Erichson, S. L. Brunton, and J. N. Kutz. Compressed dynamic mode decomposition for real-time object detection. arXiv preprint arXiv:1512.04205, 2015.

[94] N. B. Erichson, S. L. Brunton, and J. N. Kutz. Compressed dynamic mode decomposition for real-time object detection. arXiv preprint arXiv:1512.04205, 2015.

[95] C. Fieseler.

[96] C. Fieseler. Dmd in python.

[97] C. Fieseler. idmd code in matlab.

[98] C. Fieseler. Infinite series dmd.

[99] C. Fieseler, J. Kunert-Graf, and J. Nathan Kutz. The control structure of the nematode caenorhabditis elegans: Neuro-sensory integration and proprioceptive feedback. *Journal of Biomechanics*, 74:1–8, June 2018.

[100] Charles Fieseler. Learning control signals through sparse optimization.

[101] Charles Fieseler. Github. Code in MATLAB and C++, 2017.

[102] Charles Fieseler. Celegansmodel, 2019.

[103] Johannes Friedrich, Pengcheng Zhou, and Liam Paninski. Fast online deconvolution of calcium imaging data. *PLoS computational biology*, 13(3):e1005423, 2017.

[104] Thomas Gallagher, Theresa Bjorness, Robert Greene, Young-Jai You, and Leon Avery. The geometry of locomotive behavioral states in c. elegans. *PloS one*, 8(3):e59865, 2013.

[105] S. Gao and M. Zhen. Action potentials drive body wall muscle contractions in caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 108(6):2557–2562, 2011.

[106] Shangbang Gao, Sihui Asuka Guan, Anthony D Fouad, Jun Meng, Taizo Kawano, Yung-Chi Huang, Yi Li, Salvador Alcaire, Wesley Hung, Yangning Lu, et al. Excitatory motor neurons are local oscillators for backward locomotion. *Elife*, 7:e29915, 2018.

[107] Matan Gavish and David L Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.

[108] Matan Gavish and David L Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.

[109] Hong Ge, Kai Xu, Adam Scibior, Zoubin Ghahramani, et al. The turing language for probabilistic programming. In *Artificial Intelligence and Statistics*, 2018.

[110] Hong Ge, Kai Xu, Adam Scibior, Zoubin Ghahramani, et al. The turing language for probabilistic programming. In *Artificial Intelligence and Statistics*, 2018.

[111] Padraig Gleeson, David Lung, Radu Grosu, Ramin Hasani, and Stephen D Larson. c302: a multiscale framework for modelling the nervous system of caenorhabditis elegans. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1758):20170379, 2018.

[112] Andrew Gordus, Navin Pokala, Sagi Levy, Steven W Flavell, and Cornelia I Bargmann. Feedback from network states generates variability in a probabilistic olfactory circuit. *Cell*, 161(2):215–227, 2015.

[113] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing runge-kutta schemes. *Mathematics of computation*, 67(221):73–85, 1998.

[114] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing runge-kutta schemes. *Mathematics of computation*, 67(221):73–85, 1998.

[115] Jesse M Gray, Joseph J Hill, and Cornelia I Bargmann. A circuit for navigation in caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 102(9):3184–3191, 2005.

[116] Peter J Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):149–170, 1984.

[117] J. Grosek and J. N. Kutz. *Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video. arXiv preprint, arXiv:1404.7592*, 2014.

[118] Jacob Grosek and J Nathan Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.

[119] Jacob Grosek and J Nathan Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.

[120] Jeremy Gunawardena. A linear framework for time-scale separation in nonlinear biochemical systems. *PloS one*, 7(5), 2012.

[121] Andrew C Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[122] G. Haspel and M. J. O'Donovan. A perimotor framework reveals functional segmentation in the motoneuronal network controlling locomotion in caenorhabditis elegans. *Journal of Neuroscience*, 31(41):14611–14623, 2011.

[123] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

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

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

[126] Massimo A Hilliard, Cornelia I Bargmann, and Paolo Bazzicalupo. C. elegans responds to chemical repellents by integrating sensory inputs from the head and the tail. *Current Biology*, 12(9):730–734, 2002.

[127] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.

[128] Yuzo Hosono. Traveling waves for some biological systems with density dependent diffusion. *Japan Journal of Applied Mathematics*, 4(2):297, 1987.

[129] JP Huke. Embedding nonlinear dynamical systems: A guide to takens' theorem. 2006.

[130] Ingrid Hums, Julia Riedl, Fanny Mende, Saul Kato, Harris S Kaplan, Richard Latham, Michael Sonntag, Lisa Traunmüller, and Manuel Zimmer. Regulation of two motor patterns enables the gradual adjustment of locomotion strategy in caenorhabditis elegans. *Elife*, 5:e14116, 2016.

[131] R Matthew Hutchison, Thilo Womelsdorf, Elena A Allen, Peter A Bandettini, Vince D Calhoun, Maurizio Corbetta, Stefania Della Penna, Jeff H Duyn, Gary H Glover, Javier Gonzalez-Castillo, et al. Dynamic functional connectivity: promise, issues, and interpretations. *Neuroimage*, 80:360–378, 2013.

[132] Yuichi Iino and Kazushi Yoshida. Parallel use of two behavioral mechanisms for chemotaxis in caenorhabditis elegans. *Journal of Neuroscience*, 29(17):5370–5380, 2009.

[133] Mike Innes. Flux: Elegant machine learning with julia. *J. Open Source Software*, 3(25):602, 2018.

[134] Mike Innes. Flux: Elegant machine learning with julia. *J. Open Source Software*, 3(25):602, 2018.

[135] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

[136] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

[137] E. J. Izquierdo and R. D. Beer. Connecting a connectome to behavior: an ensemble of neuroanatomical models of c. elegans klinotaxis. *PLoS Comput Biol*, 9:2, 2013.

[138] E. J. Izquierdo and R. D. Beer. An integrated neuromechanical model of steering in c. elegans. In *Proceeding of the European Conference on Artificial Life*, pages 199–206, 2015.

[139] E. J. Izquierdo and R. D. Beer. The whole worm: brain-body-environment models of c. elegans. *Current Opinion in Neurobiology*, 40:23–30, 2016.

[140] Eduardo J Izquierdo. Role of simulation models in understanding the generation of behavior in c. elegans. *Current Opinion in Systems Biology*, 2018.

[141] Travis A Jarrell, Yi Wang, Adam E Bloniarz, Christopher A Brittin, Meng Xu, J Nichol Thomson, Donna G Albertson, David H Hall, and Scott W Emmons. The connectome of a decision-making neural network. *Science*, 337(6093):437–444, 2012.

[142] Sean Jewell and Daniela Witten. Exact spike train inference via ℓ_0 optimization. *The annals of applied statistics*, 12(4):2457, 2018.

[143] Meghan A Jobson, Chris M Valdez, Jann Gardner, L Rene Garcia, Erik M Jorgensen, and Asim A Beg. Spillover transmission is mediated by the excitatory gaba receptor lgc-35 in c. elegans. *Journal of Neuroscience*, 35(6):2803–2816, 2015.

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

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

[146] MR Jovanovic, PJ Schmid, and JW Nichols. Low-rank and sparse dynamic mode decomposition. *Center for Turbulence Research Annual Research Briefs*, 2012:139–152, 2012.

[147] Kadierdan Kaheman, Eurika Kaiser, Benjamin Strom, J Nathan Kutz, and Steven L Brunton. Learning discrepancy models from experimental data. *arXiv preprint arXiv:1909.08574*, 2019.

[148] Kadierdan Kaheman, Eurika Kaiser, Benjamin Strom, J Nathan Kutz, and Steven L Brunton. Learning discrepancy models from experimental data. *arXiv preprint arXiv:1909.08574*, 2019.

[149] Kadierdan Kaheman, Eurika Kaiser, Benjamin Strom, J Nathan Kutz, and Steven L Brunton. Learning discrepancy models from experimental data. *arXiv preprint arXiv:1909.08574*, 2019.

[150] Harris S Kaplan, Oriana Salazar Thula, Niklas Khoss, and Manuel Zimmer. Nested neuronal dynamics orchestrate a behavioral hierarchy across timescales. *Neuron*, 2019.

[151] J. Karbowski, G. Schindelman, C. J. Cronin, A. Seah, and P. W. Sternberg. Systems level circuit model of c. elegans undulatory locomotion: mathematical modeling and molecular genetics. *J Comput Neurosci*, 24:253–276, 2008.

[152] Saul Kato, Harris S Kaplan, Tina Schrödel, Susanne Skora, Theodore H Lindsay, Eviatar Yemini, Shawn Lockery, and Manuel Zimmer. Global brain dynamics embed the motor command sequence of caenorhabditis elegans. *Cell*, 163(3):656–669, 2015.

[153] Taizo Kawano, Michelle D Po, Shangbang Gao, George Leung, William S Ryu, and Mei Zhen. An imbalancing act: gap junctions reduce the backward motor circuit activity to bias c. elegans for forward locomotion. *Neuron*, 72(4):572–586, 2011.

[154] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[155] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[156] D. Kim, S. Park, L. Mahadevan, and J. H. Shin. The shallow turn of a worm. *Journal of Experimental Biology*, 214(9):1554–1559, 2011.

[157] Askin Kocabas, Ching-Han Shen, Zengcai V Guo, and Sharad Ramanathan. Controlling interneuron activity in caenorhabditis elegans to evoke chemotactic behaviour. *Nature*, 490(7419):273, 2012.

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

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

[160] Ivan Koryakovskiy, Manuel Kudruss, Heike Vallery, Robert Babuška, and Wouter Caarls. Model-plant mismatch compensation using reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3):2471–2477, 2018.

[161] Ivan Koryakovskiy, Manuel Kudruss, Heike Vallery, Robert Babuška, and Wouter Caarls. Model-plant mismatch compensation using reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3):2471–2477, 2018.

[162] WJ Krzanowski. Cross-validation in principal component analysis. *Biometrics*, pages 575–584, 1987.

[163] J. Kunert, E. Shlizerman, and J. N. Kutz. Low-dimensional functionality of complex network dynamics: Neurosensory integration in the caenorhabditis elegans connectome. *Physical Review E*, 89:5, 2014.

[164] J. M. Kunert, P. D. Maia, and J. N. Kutz. Functionality and robustness of injured connectomic dynamics in c. elegans: Linking behavioral deficits to neural circuit damage. *PLoS Computational Biology*, 13:1, 2017.

[165] J. M. Kunert, J. L Proctor, S. L Brunton, and J. N. Kutz. Spatiotemporal feedback and network structure drive and encode caenorhabditis elegans locomotion. *PLoS Computational Biology*, 13:1, 2017.

[166] James Kunert, Eli Shlizerman, and J Nathan Kutz. Low-dimensional functionality of complex network dynamics: Neurosensory integration in the caenorhabditis elegans connectome. *Physical Review E*, 89(5):052805, 2014.

[167] J. M. Kunert-Graf, E. Shlizerman, and J. N. Kutz. Multistability and long-timescale transients encoded by network structure in a model of c. elegans connectome dynamics. *Frontiers in Computational Neuroscience*, 11, 2017.

[168] James M Kunert-Graf, Eli Shlizerman, Andrew Walker, and J Nathan Kutz. Multistability and long-timescale transients encoded by network structure in a model of c. elegans connectome dynamics. *Frontiers in computational neuroscience*, 11:53, 2017.

[169] J Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

[170] J Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

[171] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[172] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*, volume 149. SIAM, 2016.

[173] Dmitry Laptev. Robustpca.

[174] Chven Mitchell Laura Pyrak-Nolte. Personal conversations.

[175] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

[176] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

[177] S. H. Lee and Kang Sh. Characterization of the crawling activity of caenorhabditis elegans using a hidden Markov model. *Theory Biosci*, 134:117–125, 2015.

[178] Andrew M Leifer, Christopher Fang-Yen, Marc Gershow, Mark J Alkema, and Aravinthan DT Samuel. Optogenetic manipulation of neural activity in freely moving caenorhabditis elegans. *Nature methods*, 8(2):147, 2011.

[179] W. Li, Z. Feng, P. W. Sternberg, and X. S. Xu. A c. elegans stretch receptor neuron revealed by a mechanosensitive trp channel homologue. *Nature*, 440(7084):684–687, 2006.

[180] Zhaoyu Li, Jie Liu, Maohua Zheng, and XZ Shawn Xu. Encoding of both analog-and digital-like behavioral outputs by one c. elegans interneuron. *Cell*, 159(4):751–765, 2014.

[181] J. Lighthill. Flagellar hydrodynamics. *SIAM review*, 18(2):161–230, 1976.

[182] M. A. Lim, J. Chitturi, V. Laskova, J. Meng, D. Findeis, A. Wiekenberg, B. Mulcahy, L. Luo, Y. Li, Y. Lu, and W. Hung. Neuroendocrine modulation sustains the c. elegans forward motor state. elife. 5, 2016.

[183] Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *International Conference on Machine Learning*, pages 1413–1421, 2014.

[184] Scott W Linderman, Andrew C Miller, Ryan P Adams, David M Blei, Liam Paninski, and Matthew J Johnson. Recurrent switching linear dynamical systems. *arXiv preprint arXiv:1610.08466*, 2016.

[185] Scott W Linderman, Annika LA Nichols, David M Blei, Manuel Zimmer, and Liam Paninski. Hierarchical recurrent state space models reveal discrete and continuous dynamics of neural activity in c. elegans. *bioRxiv*, page 621540, 2019.

[186] Hexuan Liu, Jimin Kim, and Eli Shlizerman. Functional connectomics from data: Probabilistic graphical models for neuronal network of c. elegans. *arXiv preprint arXiv:1711.00193*, 2017.

[187] Mochi Liu, Anuj K Sharma, Joshua W Shaevitz, and Andrew M Leifer. Temporal processing and context dependency in caenorhabditis elegans response to mechanosensation. *Elife*, 7:e36419, 2018.

[188] Ping Liu, Bojun Chen, Roger Mailler, and Zhao-Wen Wang. Antidromic-rectifying gap junctions amplify chemical transmission at functionally mixed electrical-chemical synapses. *Nature communications*, 8(1):1–16, 2017.

[189] Ping Liu, Bojun Chen, and Zhao-Wen Wang. Postsynaptic current bursts instruct action potential firing at a graded synapse. *Nature communications*, 4(1):1–12, 2013.

[190] Q. Liu, G. Hollopeter, and E. M. Jorgensen. Graded synaptic transmission at the caenorhabditis elegans neuromuscular junction. *Proceedings of the National Academy of Sciences*, 106(26):10823–10828, 2009.

[191] Lennart Ljung. System identification. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2001.

[192] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, page 108925, 2019.

[193] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, page 108925, 2019.

[194] Linjiao Luo, Damon A Clark, David Biron, L Mahadevan, and Aravinthan DT Samuel. Sensorimotor control during isothermal tracking in caenorhabditis elegans. *Journal of experimental biology*, 209(23):4652–4662, 2006.

[195] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

[196] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

[197] R. Mailler, J. Avery, J. Graves, and N. Wily. Biologically accurate 3d model of the locomotion of caernorhabditis elegans. In *International Conference on Biosciences*, pages 84–90, 2010.

[198] T. Majmudar, E. E. Keaveny, J. Zhang, and M. J. Shelley. Experiments and theory of undulatory locomotion in a simple structured medium. *J R Soc Interface*, 9:1809–1823, 2012.

[199] 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.

[200] 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.

[201] Niall M Mangan, J Nathan Kutz, Steven L Brunton, and Joshua L Proctor. Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2204):20170009, 2017.

[202] Niall M Mangan, J Nathan Kutz, Steven L Brunton, and Joshua L Proctor. Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2204):20170009, 2017.

[203] J. Mann and J. N. Kutz. Dynamic mode decomposition for financial trading strategies. *J. Quant. Finance*, 2016.

[204] J. Mann and J. N. Kutz. Dynamic mode decomposition for financial trading strategies. *J. Quant. Finance*, 2016.

[205] Jordan Mann and J Nathan Kutz. Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, 16(11):1643–1655, 2016.

[206] Jordan Mann and J Nathan Kutz. Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, 16(11):1643–1655, 2016.

[207] Steven L Mclntire, Erik Jorgensen, Joshua Kaplan, and H Robert Horvitz. The gabaergic nervous system of caenorhabditis elegans. *Nature*, 364(6435):337, 1993.

[208] J. E. Mellem, P. J. Brockie, D. M. Madsen, and A. V. Maricq. Action potentials contribute to neuronal signaling in c. elegans. *Nature neuroscience*, 11(8):865–867, 2008.

[209] Jerry E Mellem, Penelope J Brockie, David M Madsen, and Andres V Maricq. Action potentials contribute to neuronal signaling in c. elegans. *Nature neuroscience*, 11(8):865, 2008.

[210] Megan Morrison, Charles Fieseler, and J Nathan Kutz. Nonlinear control in the nematode c. elegans. *arXiv preprint arXiv:2001.08332*, 2020.

[211] Andoni Mujika, Peter Leškovskỳ, Roberto Álvarez, Miguel A Otaduy, and Gorka Epelde. Modeling behavioral experiment interaction and environmental stimuli for a synthetic c. elegans. *Frontiers in neuroinformatics*, 11:71, 2017.

[212] Georg Nagel, Martin Brauner, Jana F Liewald, Nona Adeishvili, Ernst Bamberg, and Alexander Gottschalk. Light activation of channelrhodopsin-2 in excitable cells of caenorhabditis elegans triggers rapid behavioral responses. *Current Biology*, 15(24):2279–2284, 2005.

[213] J Nathan Kutz, Joshua L Proctor, and Steven L Brunton. Applied koopman theory for partial differential equations and data-driven modeling of spatio-temporal systems. *Complexity*, 2018, 2018.

[214] Annika LA Nichols, Tomáš Eichler, Richard Latham, and Manuel Zimmer. A global brain state underlies c. elegans sleep behavior. *Science*, 356(6344):eaam6851, 2017.

[215] E. Niebur and P. Erdös. *Computer simulation of networks of electrotonic neurons*. Cambridge University Press, In Computer Simulation in Brain Science. . pp.148-163, 1988.

[216] E. Niebur and P. Erdös. Theory of the locomotion of nematodes: dynamics of undulatory progression on a surface. *Biophysical journal*, 60(5):1132–1146, 1991.

[217] Murat Okatan, Matthew A Wilson, and Emery N Brown. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural computation*, 17(9):1927–1961, 2005.

[218] Beresford N Parlett. The rayleigh quotient iteration and some generalizations for nonnormal matrices. *Mathematics of Computation*, 28(127):679–693, 1974.

[219] Barak A Pearlmutter. Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural networks*, 6(5):1212–1228, 1995.

[220] Barak A Pearlmutter. Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural networks*, 6(5):1212–1228, 1995.

[221] Laura Pereira, Paschalis Kratsios, Esther Serrano-Saiz, Hila Sheftel, Avi E Mayo, David H Hall, John G White, Brigitte LeBoeuf, L Rene Garcia, Uri Alon, et al. A cellular and regulatory map of the cholinergic nervous system of c. elegans. *Elife*, 4:e12432, 2015.

[222] Beverly J Piggott, Jie Liu, Zhaoyang Feng, Seth A Wescott, and XZ Shawn Xu. The neural circuits and synaptic mechanisms underlying motor initiation in c. elegans. *Cell*, 147(4):922–933, 2011.

[223] Jennifer K Pirri, Adam D McPherson, Jamie L Donnelly, Michael M Francis, and Mark J Alkema. A tyramine-gated chloride channel coordinates distinct motor programs of a caenorhabditis elegans escape response. *Neuron*, 62(4):526–538, 2009.

[224] T. E. Portegys. Training sensory-motor behavior in the connectome of an artificial c. elegans. *Neurocomputing*, 168:128–134, 2015.

[225] J. L. Proctor and P. A. Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International Health*, 2(7):139–145, 2015.

[226] J. L. Proctor and P. A. Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International Health*, 2(7):139–145, 2015.

[227] 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.

[228] 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.

[229] 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.

[230] Y. Rabets, M. Backholm, K. Dalnoki-Veress, and W. S. Ryu. Direct measurements of drag forces in c. elegans crawling locomotion. *Biophys J*, 107:1980–1987, 2014.

[231] Christopher Rackauckas and Qing Nie. Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017.

[232] Christopher Rackauckas and Qing Nie. Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017.

[233] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data. *arXiv preprint arXiv:1808.04327*, 2018.

[234] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data. *arXiv preprint arXiv:1808.04327*, 2018.

[235] Janet E Richmond and Erik M Jorgensen. One gaba and two acetylcholine receptors function at the c. elegans neuromuscular junction. *Nature neuroscience*, 2(9):791–797, 1999.

[236] William M Roberts, Steven B Augustine, Kristy J Lawton, Theodore H Lindsay, Tod R Thiele, Eduardo J Izquierdo, Serge Faumont, Rebecca A Lindsay, Matthew Cale Britton, Navin Pokala, et al. A stochastic neuronal model predicts random search behaviors at multiple spatial scales in c. elegans. *Elife*, 5, 2016.

[237] M. Rönkkö and G. Wong. Modeling the c. elegans nematode and its environment using a particle system. *J Theor Biol*, 253:316–322, 2008.

[238] Yasser Roudi, Joanna Tyrcha, and John Hertz. Ising model for neural data: model quality and approximate methods for extracting functional connectivity. *Physical Review E*, 79(5):051915, 2009.

[239] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-driven identification of parametric partial differential equations. *arXiv preprint arXiv:1806.00732*, 2018.

[240] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-driven identification of parametric partial differential equations. *arXiv preprint arXiv:1806.00732*, 2018.

[241] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[242] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[243] Samuel H Rudy, J Nathan Kutz, and Steven L Brunton. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *Journal of Computational Physics*, 396:483–506, 2019.

[244] Samuel H Rudy, J Nathan Kutz, and Steven L Brunton. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *Journal of Computational Physics*, 396:483–506, 2019.

[245] K. Sakata and R. Shingai. Neural network model to generate head swing in locomotion of caenorhabditis elegans. *Network*, 15:199–216, 2004.

[246] Jarred Sanders, Stanislav Nagy, Graham Fetterman, Charles Wright, Millet Treinin, and David Biron. The caenorhabditis elegans interneuron ala is (also) a high-threshold mechanosensor. *BMC neuroscience*, 14(1):156, 2013.

[247] Elizabeth R Sawin, Rajesh Ranganathan, and H Robert Horvitz. C. elegans locomotory rate is modulated by the environment through a dopaminergic pathway and by experience through a serotonergic pathway. *Neuron*, 26(3):619–631, 2000.

[248] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[249] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[250] Peter J Schmid. Application of the dynamic mode decomposition to experimental data. *Experiments in fluids*, 50(4):1123–1130, 2011.

[251] Peter J Schmid. Application of the dynamic mode decomposition to experimental data. *Experiments in fluids*, 50(4):1123–1130, 2011.

[252] Peter J Schmid, Larry Li, MP Juniper, and O Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.

[253] Peter J Schmid, Larry Li, MP Juniper, and O Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.

[254] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[255] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[256] Monika Scholz, Ashley N Linder, Francesco Randi, Anuj K Sharma, Xinwei Yu, Joshua W Shaevitz, and Andrew Leifer. Predicting natural behavior from whole-brain neural dynamics. *bioRxiv*, page 445643, 2018.

[257] Tina Schrödel, Robert Prevedel, Karin Aumayr, Manuel Zimmer, and Alipasha Vaziri. Brain-wide 3d imaging of neuronal activity in caenorhabditis elegans with sculpted light. *Nature methods*, 10(10):1013, 2013.

[258] Y. et al. Shen. An extrasynaptic gabaergic signal modulates a pattern of forward movement in caenorhabditis elegans. *eLife*, 5, 2016.

[259] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[260] Michael Small, Kevin Judd, and Alistair Mees. Modeling continuous processes from data. *Physical Review E*, 65:046704, 2002.

[261] Michael Small, Kevin Judd, and Alistair Mees. Modeling continuous processes from data. *Physical Review E*, 65:046704, 2002.

[262] Steven W Smith et al. The scientist and engineer's guide to digital signal processing. 1997.

[263] Colin Sparrow. *The Lorenz equations: bifurcations, chaos, and strange attractors*, volume 41. Springer Science & Business Media, 2012.

[264] Colin Sparrow. *The Lorenz equations: bifurcations, chaos, and strange attractors*, volume 41. Springer Science & Business Media, 2012.

[265] Greg J Stephens, Matthew Bueno de Mesquita, William S Ryu, and William Bialek. Emergence of long timescales and stereotyped behaviors in caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 108(18):7286–7289, 2011.

[266] Greg J Stephens, Bethany Johnson-Kerner, William Bialek, and William S Ryu. Dimensionality and dynamics in the behavior of c. elegans. *PLoS computational biology*, 4(4):e1000028, 2008.

[267] Christoph Stosiek, Olga Garaschuk, Knut Holthoff, and Arthur Konnerth. In vivo two-photon calcium imaging of neuronal networks. *Proceedings of the National Academy of Sciences*, 100(12):7319–7324, 2003.

[268] Tilo Strutz. Data fitting and uncertainty. *A practical introduction to weighted least squares and beyond. Vieweg+ Teubner*, 2010.

[269] Weijie Su, Małgorzata Bogdan, Emmanuel Candes, et al. False discoveries occur early on the lasso path. *The Annals of Statistics*, 45(5):2133–2150, 2017.

[270] Amit Surana. Koopman operator framework for time series modeling and analysis. *Journal of Nonlinear Science*, pages 1–34, 2018.

[271] David Sussillo, Rafal Jozefowicz, LF Abbott, and Chethan Pandarinath. Lfads-latent factor analysis via dynamical systems. *arXiv preprint arXiv:1608.06315*, 2016.

[272] M. Suzuki, T. Tsuji, and H. Ohtake. A model of motor control of the nematode c. elegans with neuronal circuits. *Artificial Intelligence in Medicine*, 35(1):75–86, 2005.

[273] B. Szigeti, P. Gleeson, M. Vella, S. Khayrulin, A. Palyanov, J. Hokanson, M. Currie, M. Cantarelli, G. Idili, and S. Larson. Openworm: an open-science approach to modeling caenorhabditis elegans. *Frontiers in computational neuroscience*, 8:137, 2014.

[274] Balázs Szigeti, Ajinkya Deogade, and Barbara Webb. Searching for motifs in the behaviour of larval drosophila melanogaster and caenorhabditis elegans reveals continuity between behavioural states. *Journal of the Royal Society Interface*, 12(113):20150899, 2015.

[275] Naoya Takeishi, Yoshinobu Kawahara, Yasuo Tabei, and Takehisa Yairi. Bayesian dynamic mode decomposition. In *IJCAI*, pages 2814–2821, 2017.

[276] Naoya Takeishi, Yoshinobu Kawahara, Yasuo Tabei, and Takehisa Yairi. Bayesian dynamic mode decomposition. In *IJCAI*, pages 2814–2821, 2017.

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

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

[279] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*, volume 89. siam, 2005.

[280] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*, volume 89. siam, 2005.

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

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

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

[284] Matthew Turk and Alex Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pages 586–587, 1991.

[285] Michael Unser, Akram Aldroubi, and Murray Eden. B-spline signal processing. i. theory. *IEEE transactions on signal processing*, 41(2):821–833, 1993.

[286] Michael Unser, Akram Aldroubi, and Murray Eden. B-spline signal processing. i. theory. *IEEE transactions on signal processing*, 41(2):821–833, 1993.

[287] Martijn P Van Den Heuvel and Hilleke E Hulshoff Pol. Exploring the brain network: a review on resting-state fmri functional connectivity. *European neuropsychopharmacology*, 20(8):519–534, 2010.

[288] Wim Van Drongelen. *Signal processing for neuroscientists*. Academic press, 2018.

[289] L. R. Varshney, B. L. Chen, E. Paniagua, D. H. Hall, and D. B. Chklovskii. Structural properties of the caenorhabditis elegans neuronal network. *PLoS Comput Biol*, 7:2, 2011.

[290] Vivek Venkatachalam, Ni Ji, Xian Wang, Christopher Clark, James Kameron Mitchell, Mason Klein, Christopher J Tabone, Jeremy Florman, Hongfei Ji, Joel Greenwood, et al. Pan-neuronal imaging in roaming caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 113(8):E1082–E1088, 2016.

[291] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[292] Daniel K Wells, William L Kath, and Adilson E Motter. Control of stochastic and induced switching in biophysical networks. *Physical Review X*, 5(3):031036, 2015.

[293] Q. Wen, M. D. Po, E. Hulme, S. Chen, X. Liu, S. W. Kwok, M. Gershow, A. M. Leifer, V. Butler, C. Fang-Yen, and T. Kawano. Proprioceptive coupling within motor neurons drives c. elegans forward locomotion. *Neuron*, 76(4):750–761, 2012.

[294] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the nematode caenorhabditis elegans. *Philos Trans R Soc Lond B Biol Sci*, 314(1165):1–340, 1986.

[295] 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.

[296] 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.

[297] R Wolke and H Schwetlick. Iteratively reweighted least squares: algorithms, convergence analysis, and numerical comparisons. *SIAM journal on scientific and statistical computing*, 9(5):907–921, 1988.

[298] Jianhong Wu and Xingfu Zou. Traveling wave fronts of reaction-diffusion systems with delay. *Journal of Dynamics and Differential Equations*, 13(3):651–687, 2001.

[299] Thomas Wutzler. Efficient treatment of model discrepancy by gaussian processes-importance for imbalanced multiple constraint inversions. *arXiv preprint arXiv:1812.07801*, 2018.

[300] Thomas Wutzler. Efficient treatment of model discrepancy by gaussian processes-importance for imbalanced multiple constraint inversions. *arXiv preprint arXiv:1812.07801*, 2018.

[301] Tianqi Xu, Jing Huo, Shuai Shao, Michelle Po, Taizo Kawano, Yangning Lu, Min Wu, Mei Zhen, and Quan Wen. Descending pathway facilitates undulatory wave propagation in caenorhabditis elegans through gap junctions. *Proceedings of the National Academy of Sciences*, 115(19):E4493–E4502, 2018.

[302] Chen Yao and Erik M Bollt. Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems. *Physica D: Nonlinear Phenomena*, 227(1):78–99, 2007.

[303] Chen Yao and Erik M Bollt. Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems. *Physica D: Nonlinear Phenomena*, 227(1):78–99, 2007.

[304] Eviatar Yemini, Tadas Jucikas, Laura J Grundy, André EX Brown, and William R Schafer. A database of caenorhabditis elegans behavioral phenotypes. *Nature methods*, 10(9):877, 2013.

[305] Linan Zhang and Hayden Schaeffer. On the convergence of the sindy algorithm. *arXiv preprint arXiv:1805.06445*, 2018.

[306] Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006.

[307] Mei Zhen and Aravinthan DT Samuel. C. elegans locomotion: small circuits, complex functions. *Current opinion in neurobiology*, 33:117–126, 2015.

[308] Peng Zheng, Travis Askham, Steven L Brunton, J Nathan Kutz, and Aleksandr Y Aravkin. A unified framework for sparse relaxed regularized regression: Sr3. *IEEE Access*, 7:1404–1423, 2018.

[309] Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996.

[310] Manuel Zimmer, Jesse M Gray, Navin Pokala, Andy J Chang, David S Karow, Michael A Marletta, Martin L Hudson, David B Morton, Nikos Chronis, and Cornelia I Bargmann. Neurons detect increases and decreases in oxygen levels using distinct guanylate cyclases. *Neuron*, 61(6):865–879, 2009.

[311] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

Appendix A

APPENDIX A: INFORMATION THEORY FOR CONTROL SIGNALS

The following supplement details the analysis, algorithms and mathematical theory used to learn control signals from data and reproduce the observed low-dimensional connectomic dynamics in *C. elegans* whole-brain recordings. In particular, two points are more fully examined: 1) Comparisons between learned transitions and expert hand-labeled transitions, which are given for entire times series; and 2) Stopping criteria for the iterative algorithm. Additionally, more information about the variable selection algorithms and how they can be biologically interpreted is given. In the main text, the variable selection for a single control signal for a single individual was shown, and here additional examples are shown. Some summary statistics across individuals are also shown and interpreted. Finally, we also show more PCA projections of data reconstructions for additional individuals. All code and data are available through github for reproducibility and verification.

A.1 Biological systems as feedback-controlled systems

Biological units, in this case neurons, are generally not linear. However, some of them may behavior linear *up to an external signal*. For example, the actual transition between "on" and "off" in a two-state network transition may involve a strong nonlinearity, but the dynamics within each state may be well described by a linear model, that is, slow exponential decay or simple oscillations. In other words, this can be treated as a multi-scale system in which the transitions are much faster than the within-state dynamics, which are themselves simple.

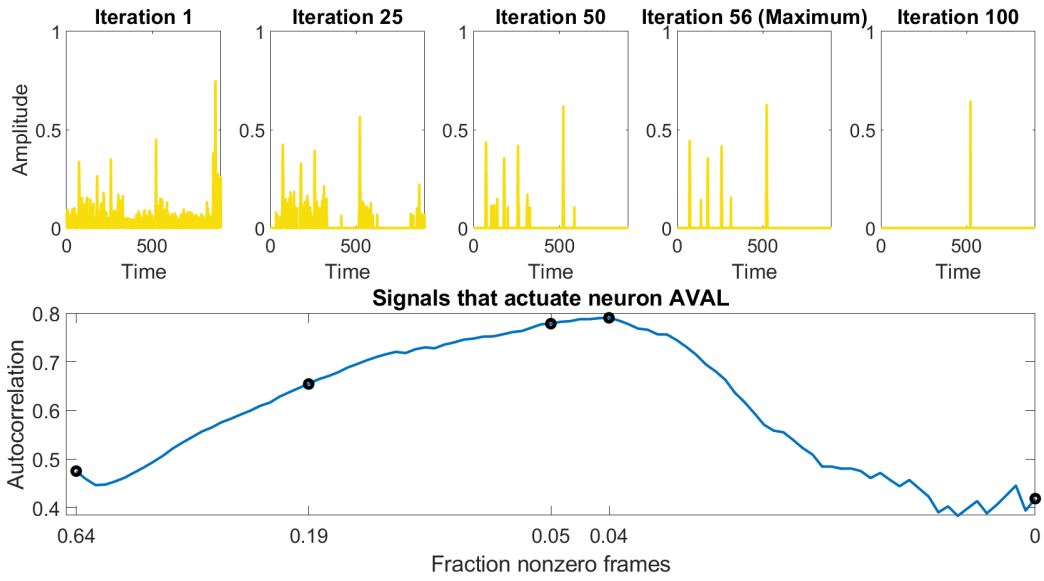

Figure A.1: Increasing sparsity of an example well-reconstructed signal, onset of Reversal.

A.2 Learning control signals from data

A.2.1 The optimization problem

The overall optimization problem we will solve is given in the main text:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{U}} \left[||\mathbf{A}\mathbf{X}_1 + \mathbf{B}\mathbf{U} - \mathbf{X}_2||_2 + \lambda ||\mathbf{U}||_0 \right] \tag{A.1}$$

However, because this has a trivial solution it must be regularized and initialized with care. Most of the following discussion involves motivating and explaining the details of the regularization.

A.2.2 The residuals in DMD vs. DMDc

The "naive" DMD problem can be cast as an opimization problem seeking to minimize the one-step residual:

$$\epsilon_{naive} = \mathbf{A}_{naive}\mathbf{X}_1 - \mathbf{X}_2$$

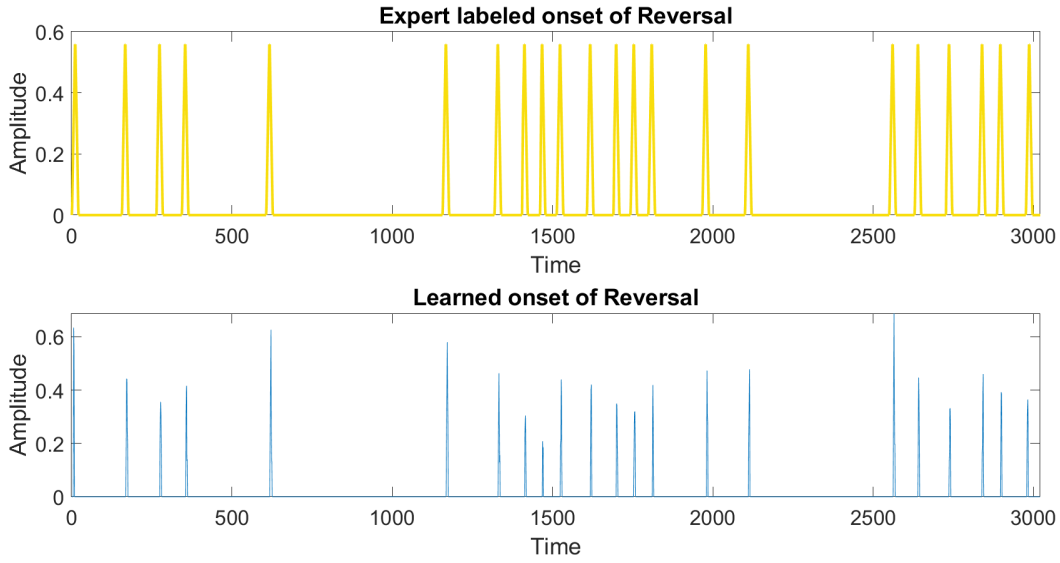

Figure A.2: The entire time series of a well-reconstructed signal, onset of Reversal.

Because the norm being minimized is the L2 norm, this is equivalent to a Gaussian prior on the residual term, ϵ_{naive}. Conceptually, this is consistent with a true underlying linear system with sensor noise (or process noise on the same timescale as the data timesteps), but is inconsistent with a forced linear system. Indeed, one diagnostic of whether or not DMD is an appropriate model for the system is whether this residual is in fact Gaussian, and this is the basis of this algorithm for learning control signals from data.

The "controlled" DMD problem, i.e. DMDc, has a different residual and a fundamentally different matrix $\mathbf{A}_{control}$:

$$\epsilon_{control} = \mathbf{A}_{control}\mathbf{X}_1 + \mathbf{B}\mathbf{U} - \mathbf{X}_2 \tag{A.2}$$

If the underlying system truly is linear, then this residual, $\epsilon_{control}$, will actually be Gaussian while ϵ_{naive} will not be. Thus, it may be possible to decompose the non-Gaussian components of the original error:

$$\epsilon_{naive} \approx \tilde{\mathbf{U}} + \epsilon_{control} \tag{A.3}$$

where this equation will need assumptions on $\tilde{\mathbf{U}}$, e.g. that it is sparse and/or low-rank.

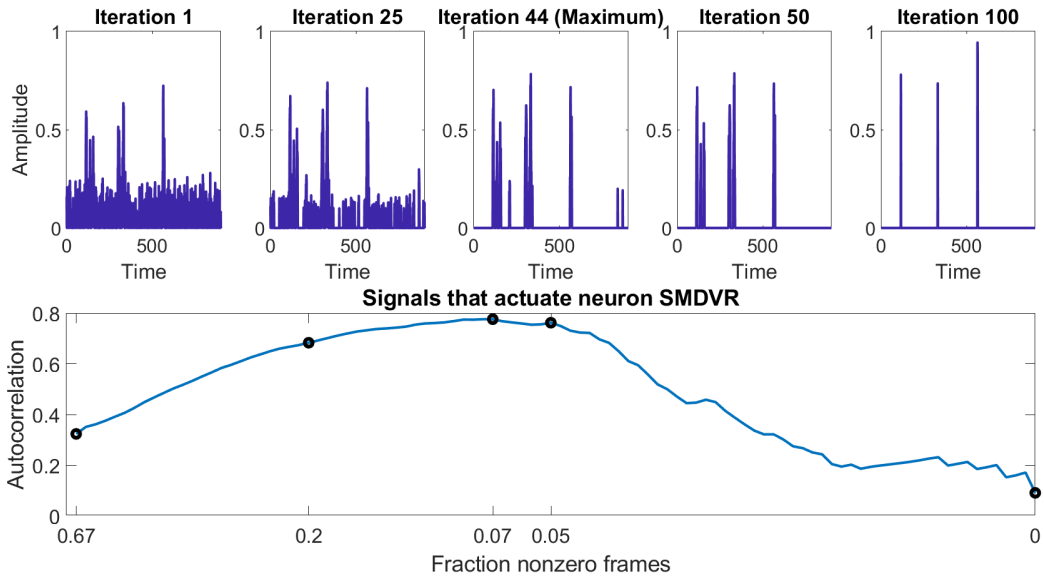

Figure A.3: Increasing sparsity of an example well-reconstructed signal, onset of Ventral Turn. There are more spurious spikes identified in this signal as compared to the Reversal signal.

This cannot be achieved in a single decomposition, because the matrix $\mathbf{A}_{naive}$ attempts to compensate for the missing control signal, and thus is some combination of $\mathbf{A}_{control}$ and B. However, this inspires an iterative algorithm to search for sparse control signals to be explained in the next section.

Note that in biological datasets, neither of these models will be accurate enough to obtain a truly Gaussian residual due to true nonlinearities, stochastic forcing terms, and unmeasured neurons. Thus it is important to keep in mind that this algorithm and indeed this entire approach seeks to inform more sophisticated modeling and experiments, rather than finding a true model.

A.2.3 Biological context

The *C. elegans* neural system gives rise to multiple discrete and mutually exclusive states, namely forward and backwards motion, and dorsal and ventral turns. It is a subject of ongo-

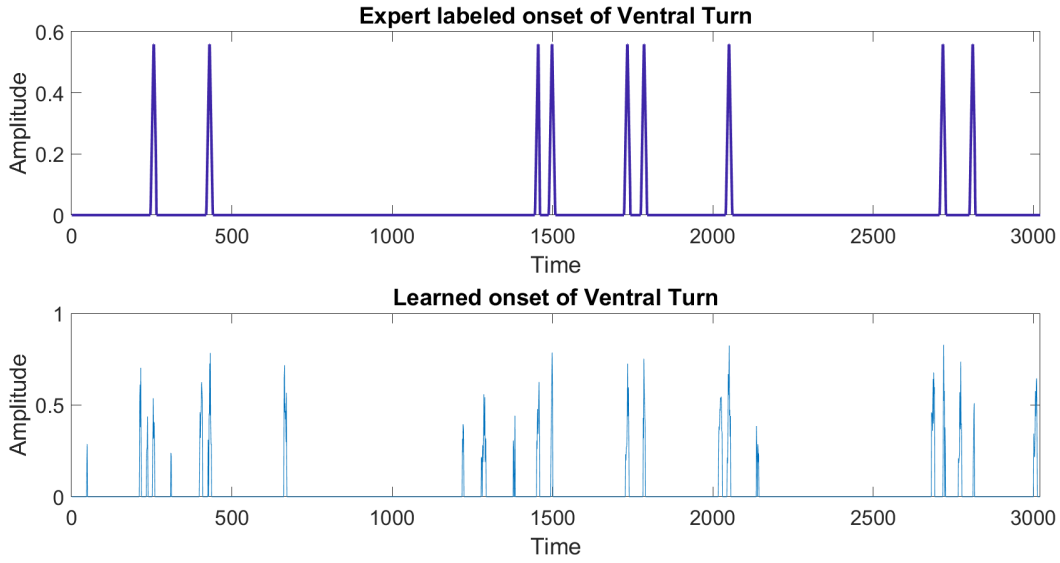

Figure A.4: The entire time series of a well-reconstructed signal, onset of Ventral Turn.

ing debate as to how discrete these are, and there is work showing that forward motion both within a single medium or between different media has multiple substates [38]. In addition, turns can be of variable depths, and many authors propose distinct very deep "omega" and distinct shallower turns though some work shows a continuum of turning angles [41, 156]. Yet another caveat is that immobilized worm whole-brain imaging shows a significantly different pattern of activation from that of freely moving individuals [256], which may be partly explained by the lack of some natural states and/or the presence of unnatural ones.

While acknowledging these points, most modeling work starts from the viewpoint of distinct states, and we use 5 experimentalist identified states here: forward, reversal, dorsal turn, ventral turn, and (in some datasets) quiescence. This gives rise to a natural assumption that can be placed on the control signal decomposition (equation A.3): $\tilde{\mathbf{U}}$ is sparse.

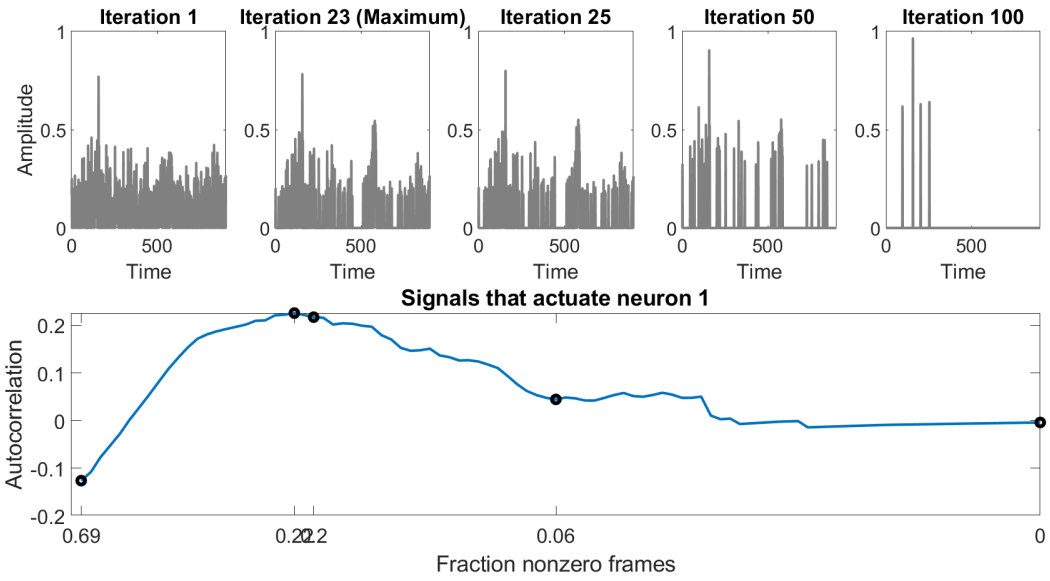

Figure A.5: A signal that does not correspond to any known behavior and is of low quality. The neuron actuated most strongly by this signal does not have a name, and thus cannot be compared to literature studies or other datasets.

A.2.4 An iterative algorithm for sparse control signals

The problem here has the same form as the minimization of DMDc (equation A.2), but now we are also minimizing over $\tilde{\mathbf{U}} = \mathbf{BU}$, where there is no unique decomposition of the matrices $\mathbf{B}$ and $\mathbf{U}$. Formally, this can be written as:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{U}} \left[||\mathbf{AX}_1 + \mathbf{BU} - \mathbf{X}_2||_2 + \lambda ||\mathbf{U}||_0 \right] \tag{A.4}$$

The 0-norm is implemented as sequential least-squares thresholding [305], and $\mathbf{A}$ and $\mathbf{B}$ are minimized in alternating steps with $\mathbf{U}$:

1. Initialize $\mathbf{U}$ and determine the rank of the control signal (i.e. the number of rows, r)

2. Solve for $\mathbf{A}$ and $\mathbf{B}$ given $\mathbf{U}$

3. Solve for $\mathbf{U}$ only, without any penalties, given $\mathbf{A}$ and $\mathbf{B}$

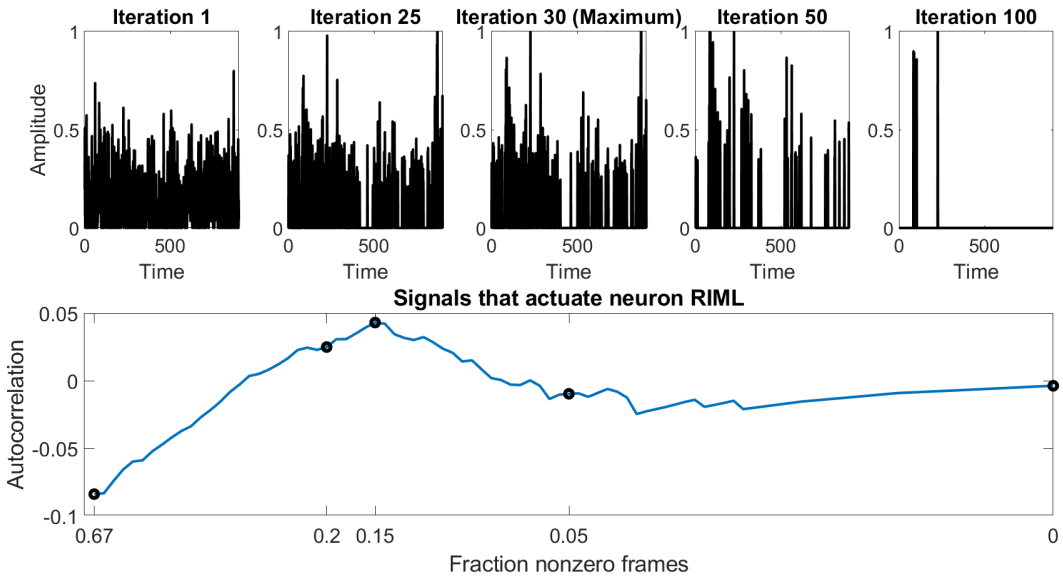

Figure A.6: A signal that does not correspond to any known behavior and is of low quality. The neuron actuated most strongly by this signal is identified as 'RIML' and is known to be important in reversals. However, this signal is almost entirely noise and cannot be interpreted as real.

4. Sparsify $\mathbf{U}$ by removing entries corresponding to a) the previous sparsity pattern and b) the bottom $y\%$ for each row

5. Save the new sparsity patttern and repeat steps 2-5

This is a greedy algorithm that simply removes the smallest entries from $\mathbf{U}$ in each step, and will be heavily dependent upon the initialization and the chosen rank.

A.2.5 Stopping condition and choosing the "right" signal

Goals of a metric

Distinguishing signals from noise is a common and unsolved problem. In this work as in many algorithms, it is only possible to separate signals out by imposing some conditions, namely that they are sparse. However, there is no guarantee that signals that input into

different neurons have the same level of sparsity, and thus optimizing a global parameter is difficult. In addition, because the initialization of the algorithm is based on the residuals of the model being studied, it is important to not include signals that are simply noise.

Currently implemented metric

This work uses a stopping condition that tries to accomplish both of the above goals:

$$\max_{sparsity} (acf(x))$$

where $acf(x)$ is the one-step autocorrelation function. Intuitively, a signal will have much higher autocorrelation than the noise; additional metrics with separate motivations are discussed in the next section. As shown in Figure A.7, the maximum of the autocorrelation shows up at a different sparsity level for different signals, and it peaks smoothly without much non-convexity. In addition, some signals always have lower autocorrelation, which is evidence for them being either noise or of signals that are not amenable to this methodology. For example, it is possible to construct signals that would be very sparse if the self-dynamics of the model were extended to include nonlinearities but would be dense and not autocorrelated when considering linear residuals; this is an area for future work. At this time, signals with an autocorrelation below a threshold are discarded.

Additional metrics

There are many possible metrics that could be used as stopping conditions, particularly those related to information theory. Each of these metrics uses an intuition about what it means to be a "noise mode."

A first additional idea: Compare the lengths of contiguous blocks of non-zero activity. As sparsity increases, a Gaussian signal will have non-zero blocks distributed according to an exponential distribution. The sparsity for which this is least true (i.e. the worst fit) as measured by negative log likelihood or BIC has the "most signal." As shown in figure S1,

this discriminates signals into the same categories as the simple autocorrelation metric with very similar minima although there is significantly more noise.

However, a major caveat is that comparing negative log likelihoods or BIC scores across different datasets (i.e. the increasingly sparse signal) is a statistically dubious procedure. In addition, the number of measurements (block lengths) drastically decreases as the sparsity is increased, changing the number of data points used to fit the model.

A second alternative: use time series entropy measures, with the minima corresponding to the least noisy signal. The sampling and permutation entropies are calculated and shown in figure S2. Again, these produce the same discrimination of signal classes and nearly the same "best" sparsity as with the simple autocorrelation metric.

These methods also come with a major caveat: difficult to interpret external parameters. Although these entropy measurements are not very sensitive to the choice of parameters, it makes the method more difficult to generalize. In addition, it is again unclear how to normalize these measurements across differing amounts of data, and a straight line fit must be subtracted out in order to get clean global minima for the permutation entropy, and the sampling entropy is divided by the number of data points without rigorous justification.

Because the $||\mathbf{U}||_0$is replaced with sequential least-squares thresholding, the (adaptive) threshold parameter X shows up directly in the algorithm instead of λ.

A.2.6 Overview of learning algorithm and results

Figure A.7 shows that as the rank of the control signal to be learned is increased, "on" signals that are highly correlated to experimentally observed ones appear. In addition, these signals are consistently present and have a consistent level of autocorrelation even as the rank is drastically increased. Importantly these signals are not simply fitting noise, as measured by the autocorrelation (greater than 0.8) and other information metrics. This holds across datasets, and a signal that is significantly correlated to the hand-labeled set is found in all 15 individuals analyzed.

As the rank is increased further, signals that are not clearly distinguishable from noise

appear; these are shown as black if their autocorrelation is below 0.5, and grey in the region 0.5-0.8. These grey signals of intermediate quality may have some interpretability; this is discussed more in the supplement.

Neurons with names can be identified across datasets, but those with numbers cannot. Thus, although there are high-quality signals that actuate potentially interesting neurons (labeled 1 and 64), no broader conclusions can be drawn.

A.3 Variable selection

The main text described the elimination path of a single control signal, the Dorsal Turn, for a single individual. This behavior has been well studied and a chief result of these studies are which neurons are involved in this behavior. In additional to these well-known neurons, additional neurons are identified that predict the onset of the Dorsal turn, and these are good targets for follow-up studies.

The encoding of two separate behaviors are described here, both of which are similarly well studied: Ventral Turns in Figures A.8 to A.11, and Reversals in Figures A.12 to A.15. The Ventral Turn encoding is similar to the Dorsal Turn encoding shown in the main text: the well-known neurons SMDVL/R are important for encoding, but additional unidentified neurons are also important. Interestingly, the neurons RIVL/R (which are also associated with this behavior) only appear much later in the elimination path and are not very important for actually encoding the behavior. This is due to the timing of activation of these neurons, which is largely after the onset of the Ventral Turn; without significant activity before the behavioral onset, they are not true encoding neurons but seem to carry out the action once it has been "turned on."

In a similar way, there are interesting subtleties in the encodings of the Reversal onset signal. There are eight neurons (4 pairs) known to be important in this behavior, AIBL/R, AVAL/R, AVEL/R, and RIML/R, but they are not prominent in the elimination path. In this way they are similar to the RIVL/R neurons above: many of these neurons only have significant activity after the onset of the behavior. However, looking closer at the all-neuron

encoding reveals something interesting: AVAR appears with a negative weight. This means that preceding a reversal, this neuron actually has a significant dip in activity. The signal is also encoded in several other neurons, notably several associated with Forward motion (RIBL/R and AVBL/R) appear with negative weights. Two more interesting weightings are RIS, which is associated with sleep, and several unidentified neurons. These could be the subject of follow-up experimental studies to determine how they might trigger this behavior.

A.4 Reconstructions for additional individuals

The main text showed PCA projections and individual neuron reconstructions for a select individual and neurons. Figures in section IV show PCA projections of model reconstructions using supervised or no control signals, as well as the data.

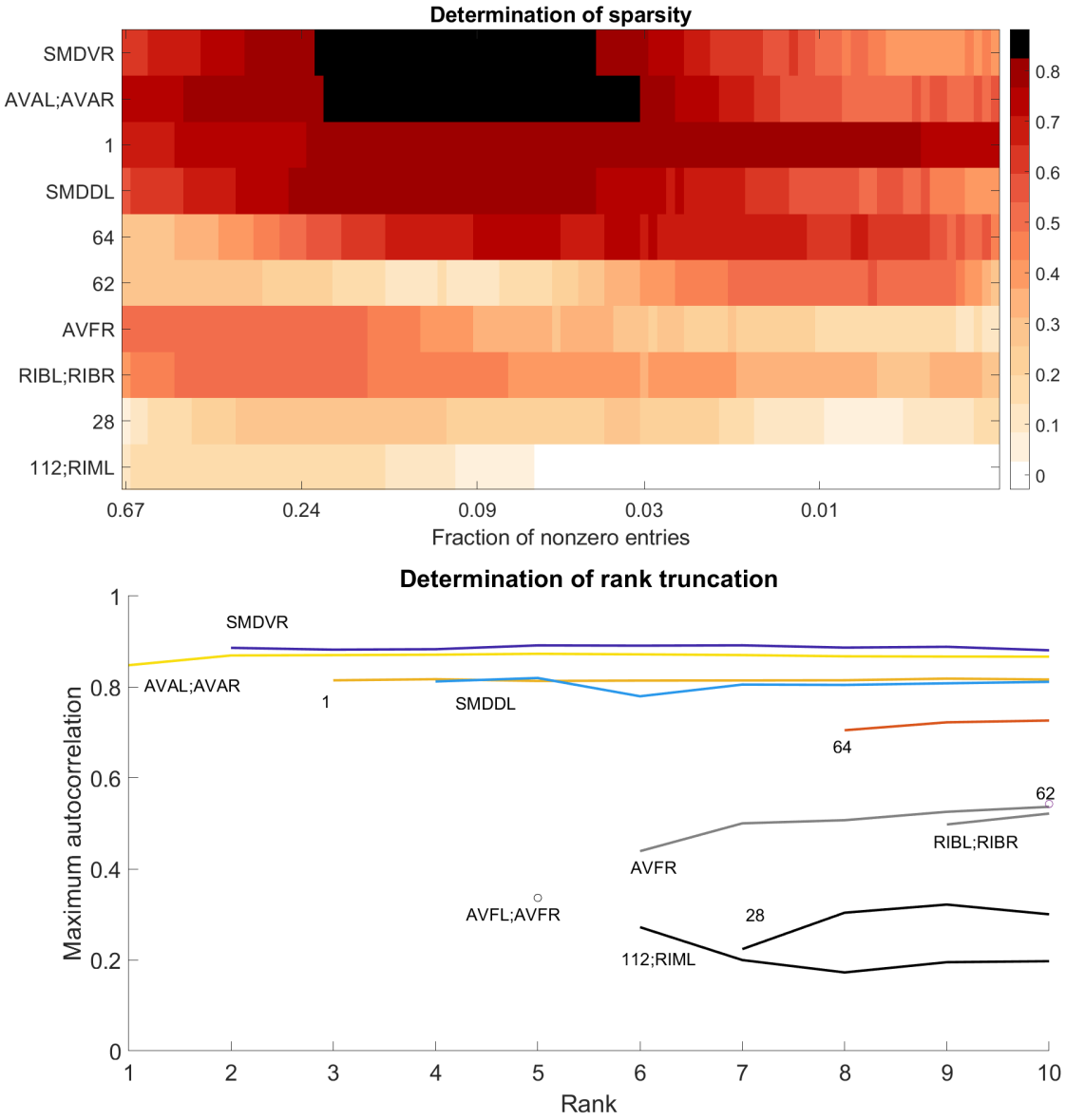

Figure A.7: a. A heatmap of the autocorrelations of all signals across iterations. Each of the ten signals has a different maximum correlation, and the top two high quality signals (which actuate SMDVR and AVAL/R) are shown in more detail in other figures. b. As the free parameter 'r' (number of control signals) is increased, identifiable signals appear. Most have a similar quality (maximum autocorrelation) across different ranks.

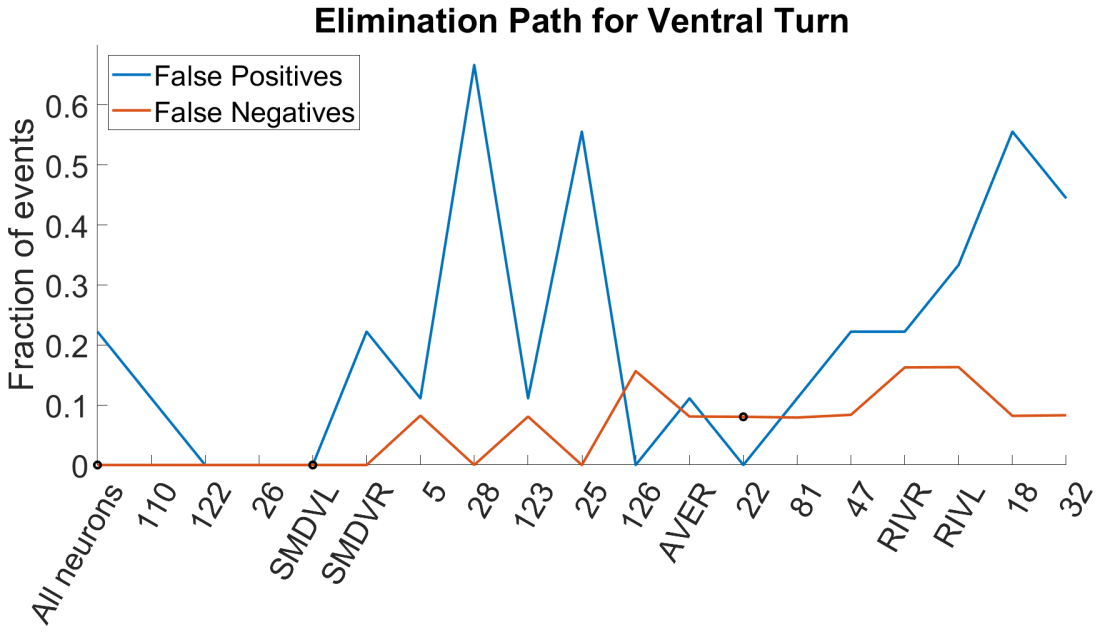

Figure A.8: The elimination path of the Ventral turn signal for one individual dataset.

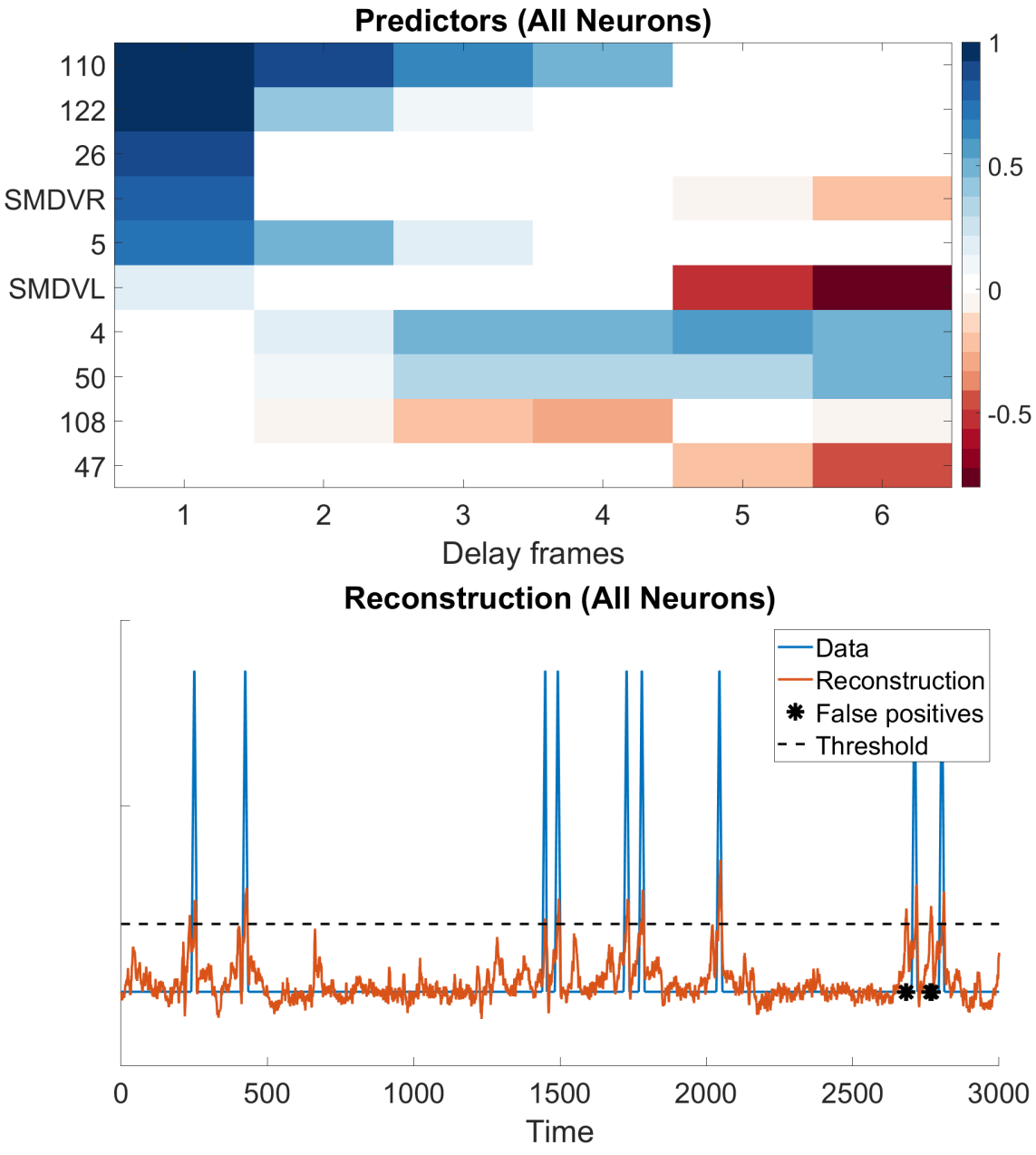

Figure A.9: a) The encoding variables when all neurons are allowed in the reconstruction. The well-known neuron Left/Right neuron pair, SMDVL/R, are highly predictive. Additionally, other unidentified neurons appear and may be promising for future study. b) The reconstruction across the entire time series using the variables selected above.

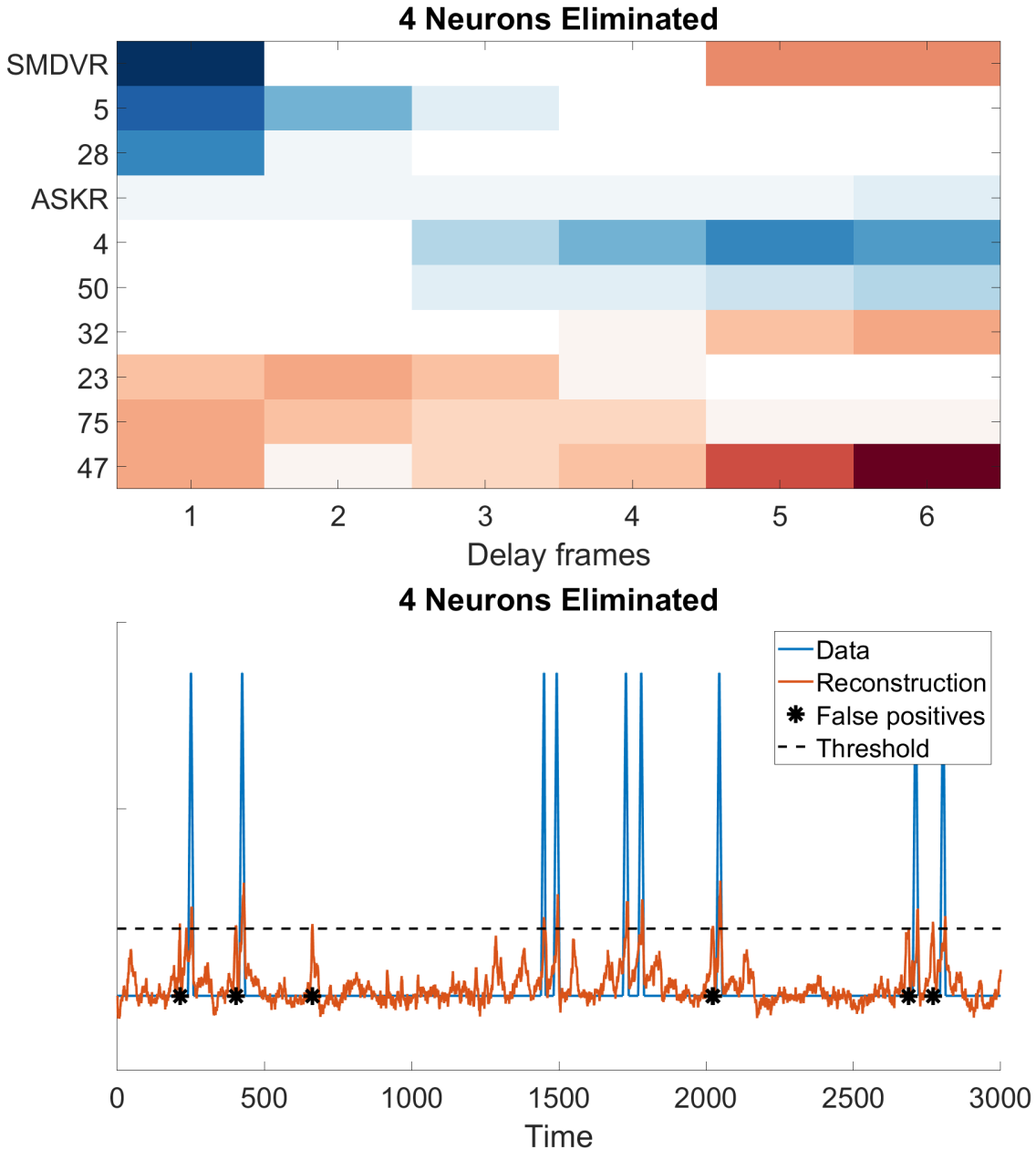

Figure A.10: a) The encoding variables when some neurons are removed from the reconstruction. Additional neurons appear. b) The reconstruction across the entire time series using the variables selected above.

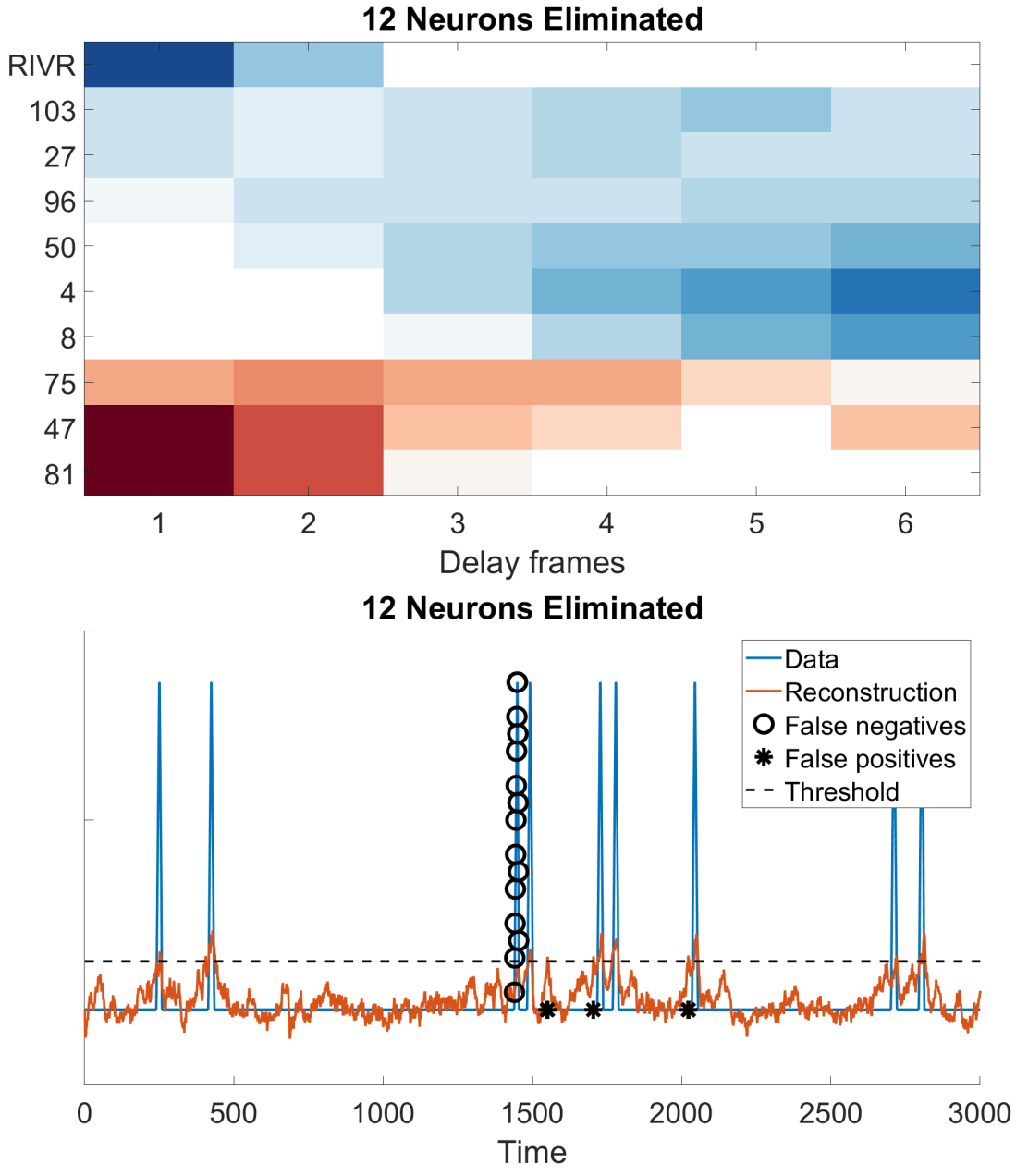

Figure A.11: a) The encoding variables when some neurons are removed from the reconstruction. Additional neurons appear. b) The reconstruction across the entire time series using the variables selected above.

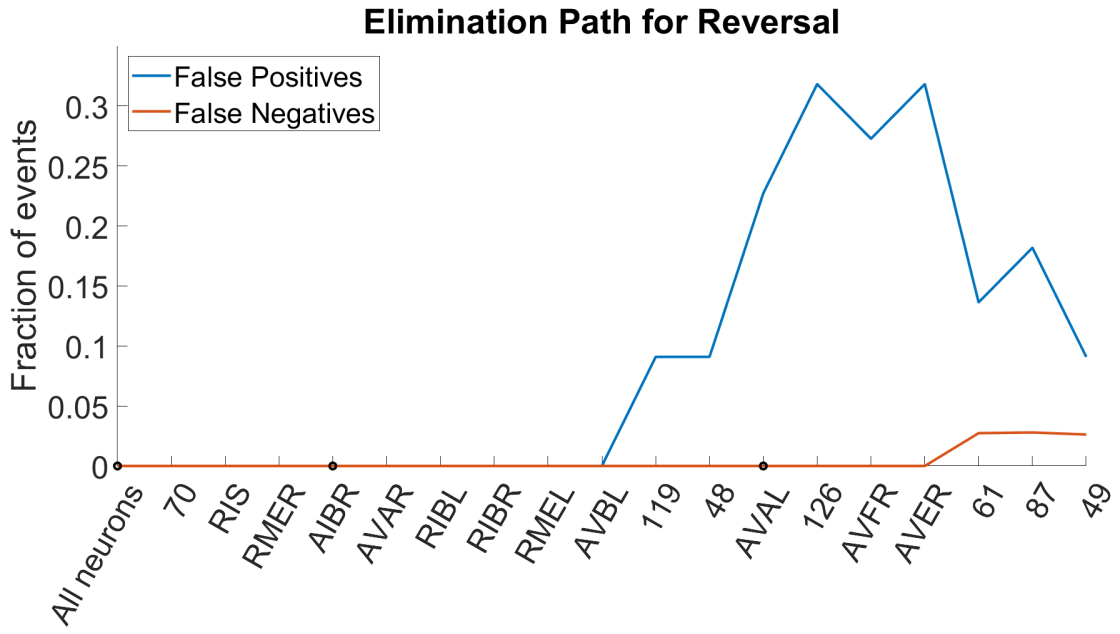

Figure A.12: The elimination path of the Reversal signal for one individual dataset. Several neuron pairs are well-known to be associated with this behavior, particularly AIBL/R, AVAL/R, AVEL/R, and RIML/R. These neurons do not show up as much in the elimination path because their activity only increases significantly after the onset of the reversal behavior, which is much slower than the turns.

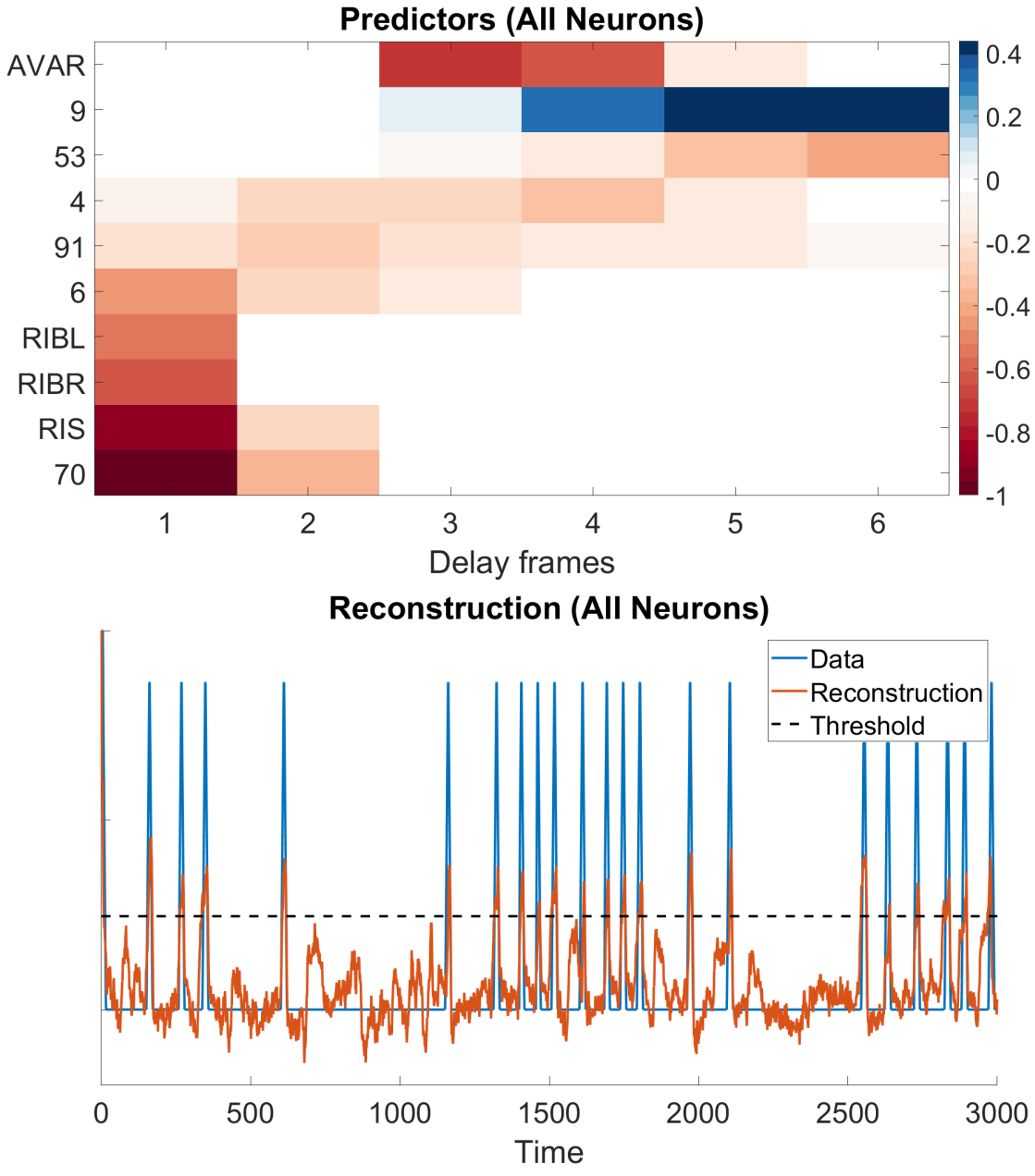

Figure A.13: a) The encoding variables when all neurons are allowed in the reconstruction. Many well-known neurons appear, as well as some unidentified ones. b) The reconstruction across the entire time series using the variables selected above.

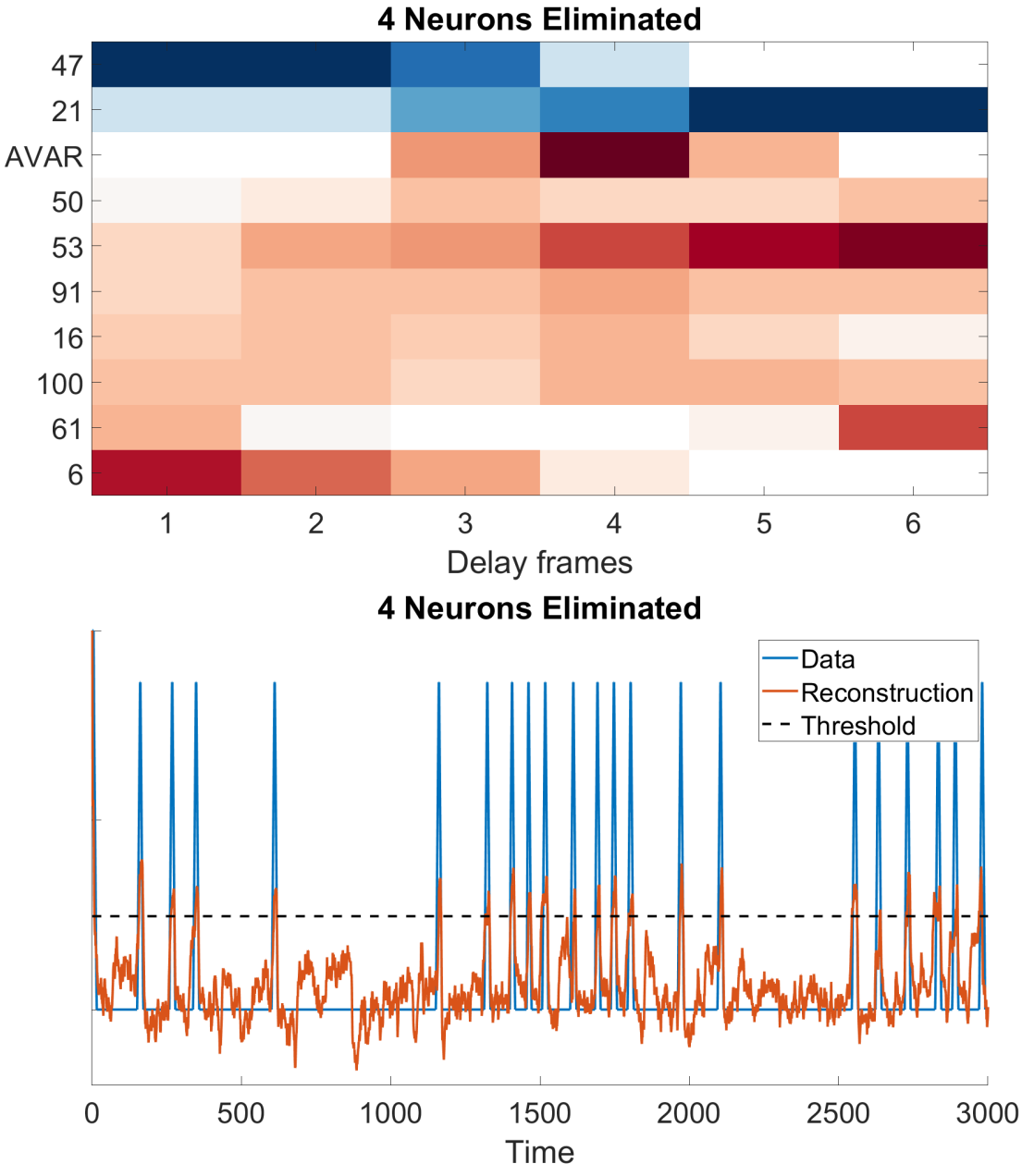

Figure A.14: a) The encoding variables when some neurons are removed from the reconstruction. Additional neurons appear. b) The reconstruction across the entire time series using the variables selected above.

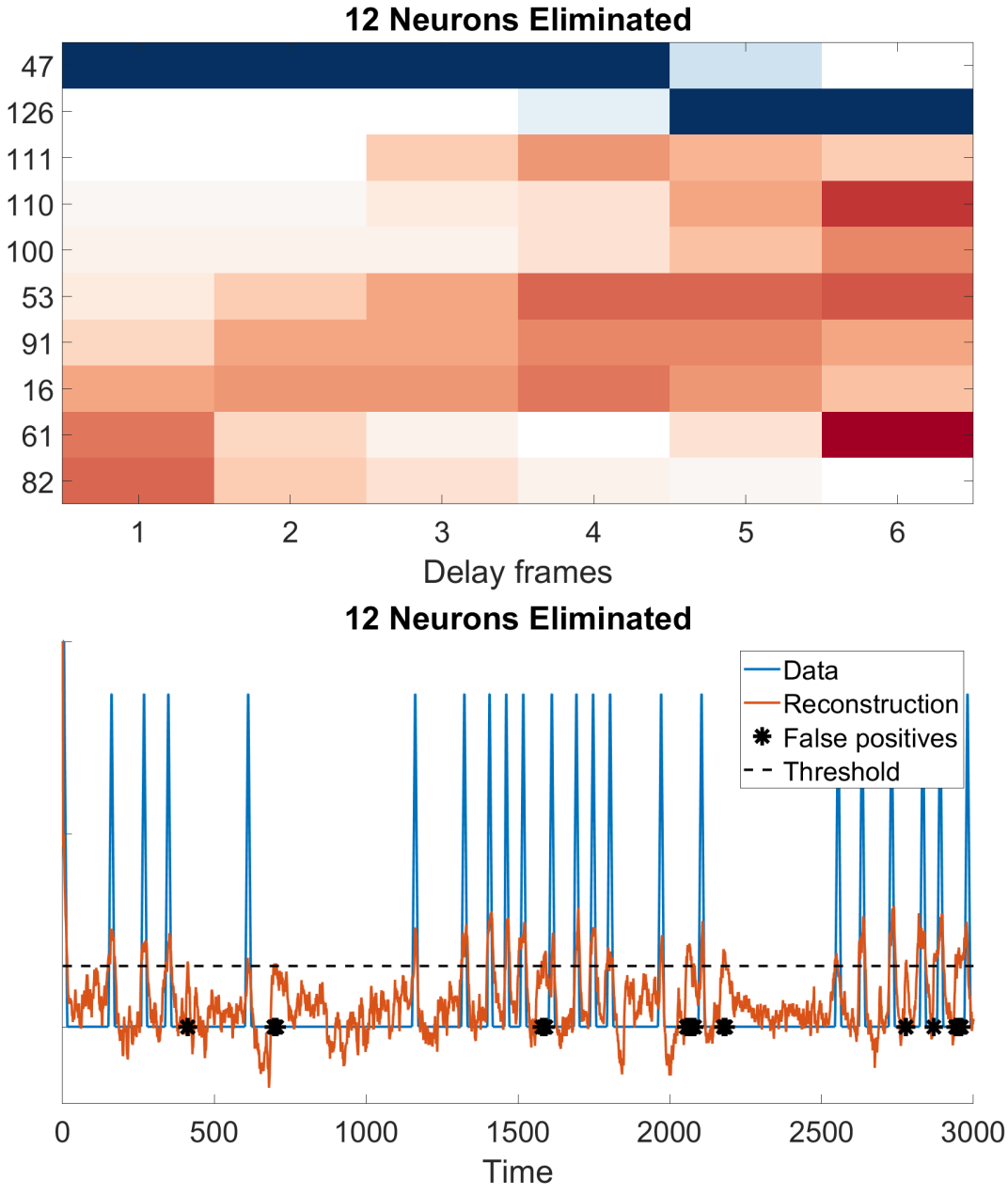

Figure A.15: a) The encoding variables when some neurons are removed from the reconstruction. Additional neurons appear. b) The reconstruction across the entire time series using the variables selected above.

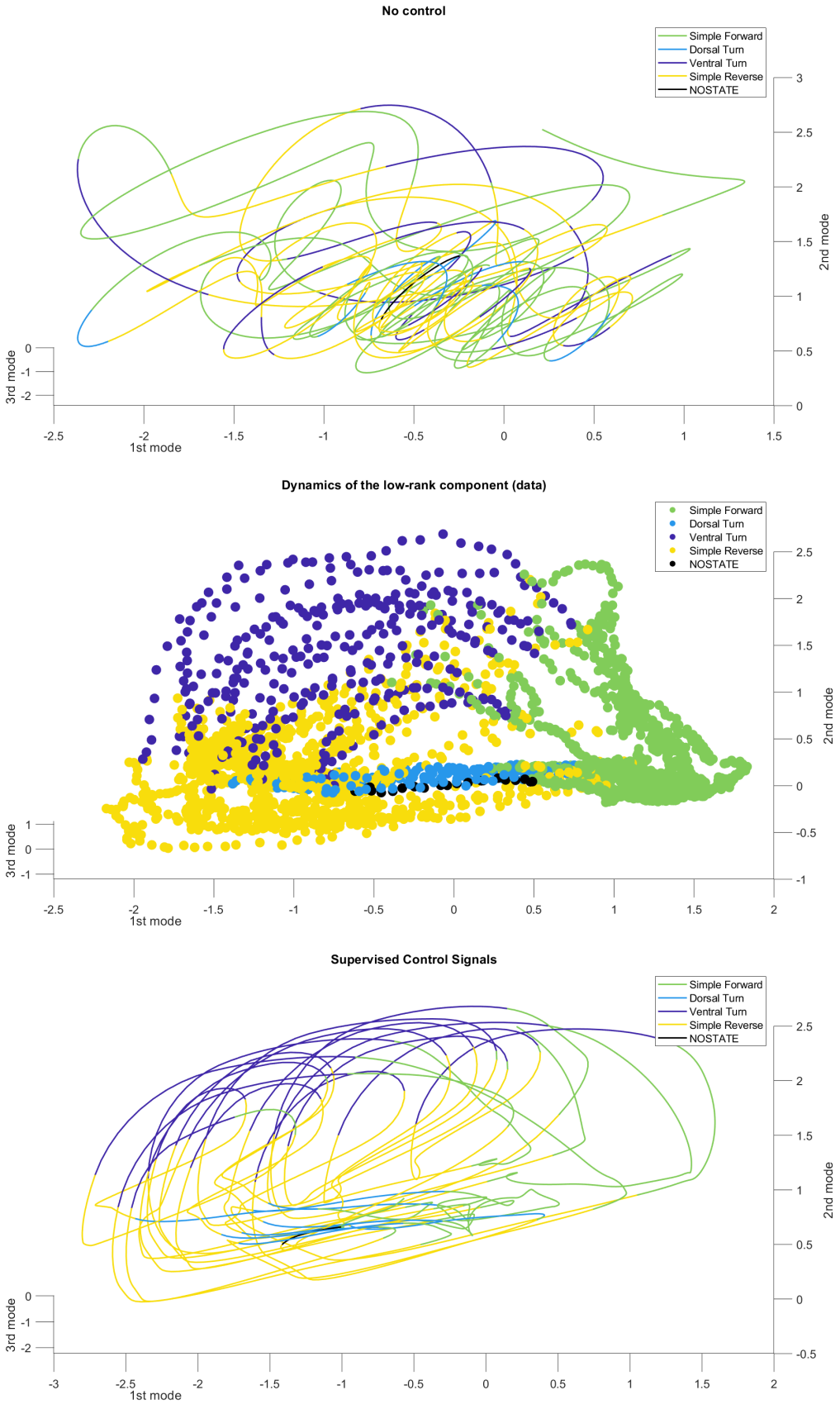

Figure A.16: PCA projections for individual 1.

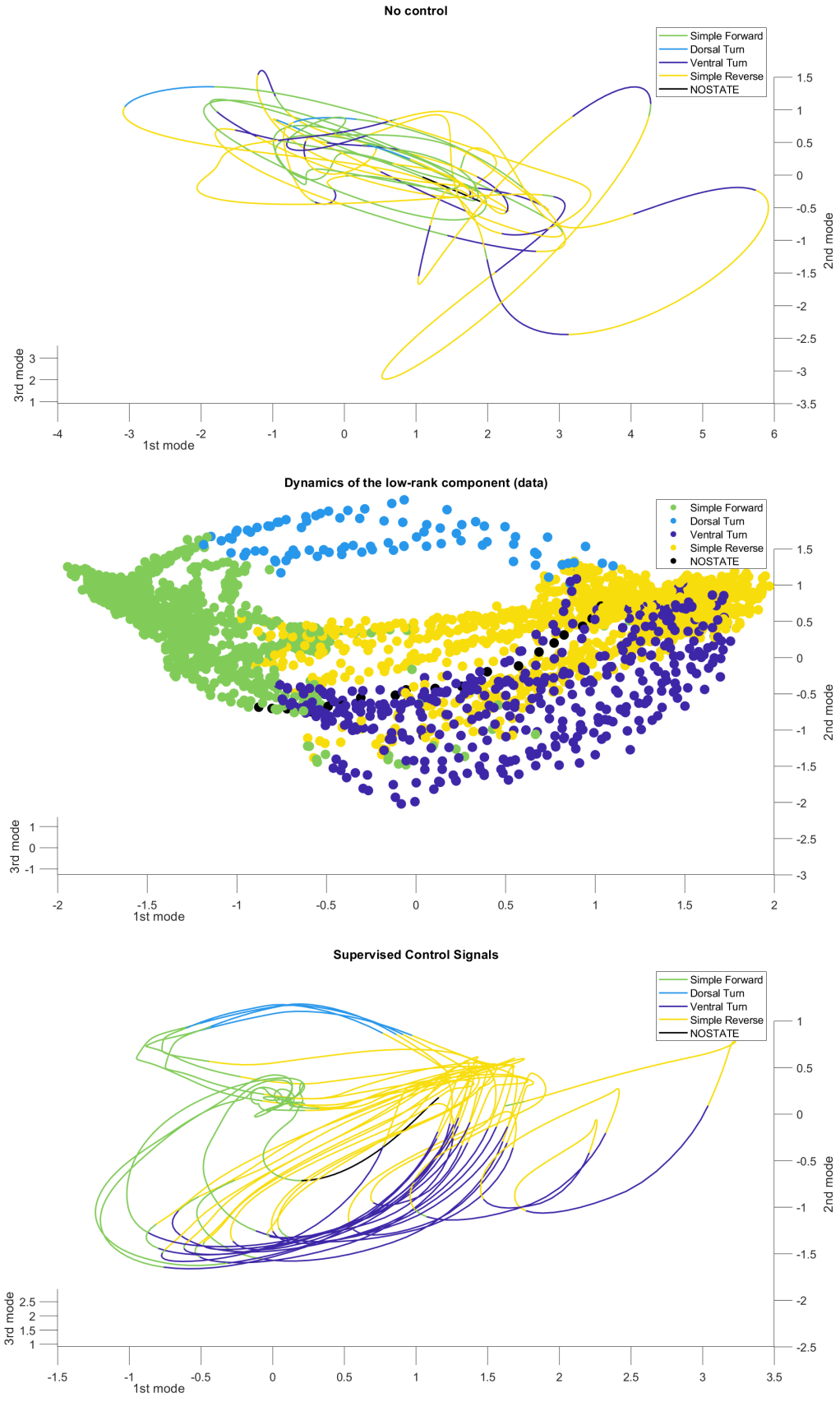

Figure A.17: PCA projections for individual 2.

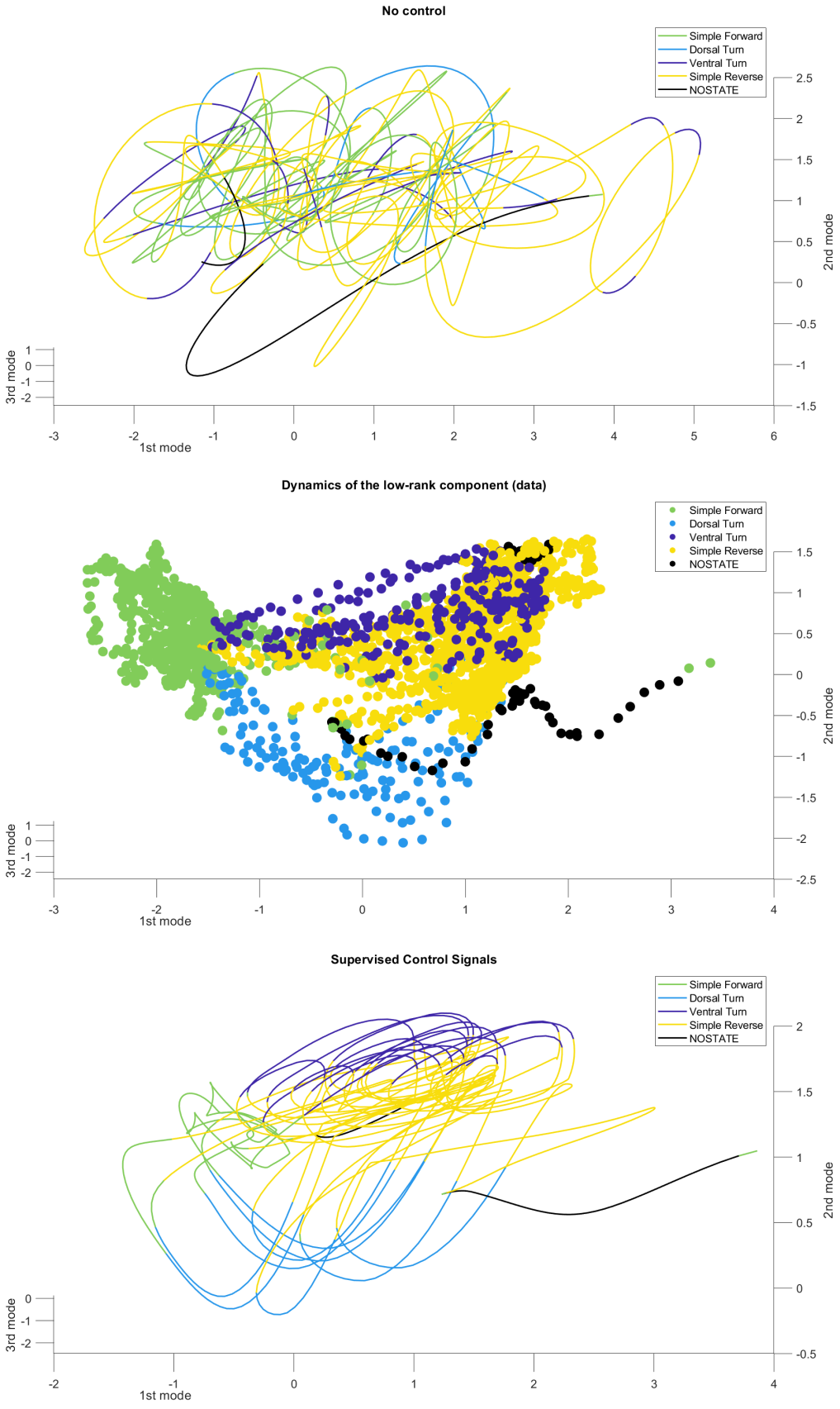

Figure A.18: PCA projections for individual 3.

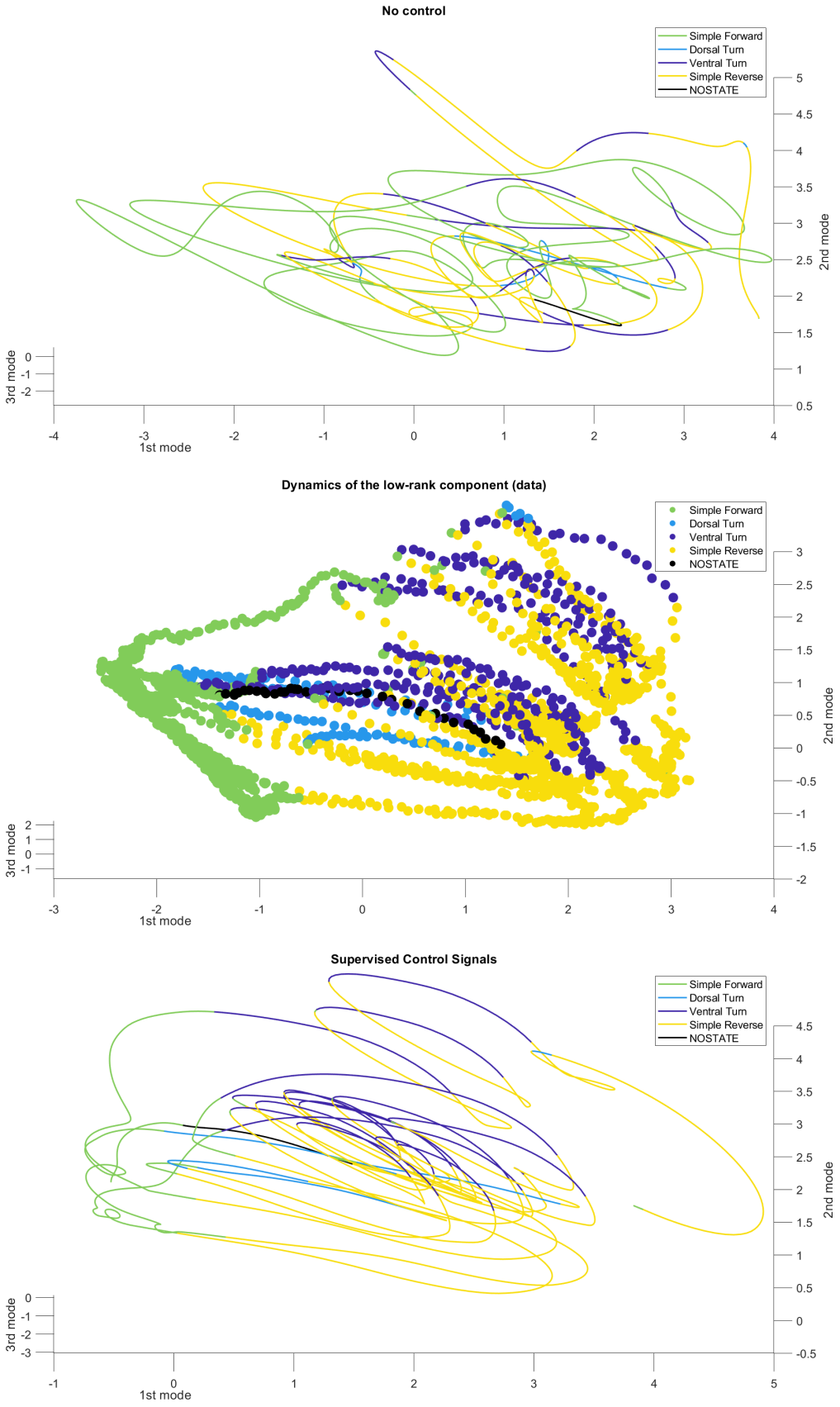

Figure A.19: PCA projections for individual 4.