Supporting Information for: Discovering governing equations from data:

Sparse identification of nonlinear dynamical systems

Steven L. Brunton[1], Joshua L. Proctor[2], J. Nathan Kutz[3]

[1] Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, United States

[2]Institute for Disease Modeling, Intellectual Ventures Laboratory, Bellevue, WA 98004, United States

[3] Department of Applied Mathematics, University of Washington, Seattle, WA 98195, United States

Contents

1 Technical introduction

There is a long and fruitful history of modeling dynamics from data, resulting in powerful techniques for system identification [1]. Many of these methods arose out of the need to understand complex flexible structures, such as the Hubble space telescope or the international space station. The resulting models have been widely applied in nearly every branch of engineering and applied mathematics, most notably for model-based feedback control. However, methods for system identification typically require assumptions on the form of the model, and most often result in linear dynamics, limiting their effectiveness to small amplitude transient perturbations around a fixed point of the dynamics [2].

A recent breakthrough in nonlinear system identification has resulted in a new approach to determine the underlying structure of a nonlinear dynamical system from data [3]. This method uses symbolic regression to determine dynamics and conservation laws, and it balances the complexity of the model (measured in the number of model terms) with the agreement with data. The resulting identification algorithm realizes a long-sought goal of the physics and engineering communities to discover dynamical systems from data. However, the symbolic regression problem is expensive, does not clearly scale well to large-scale dynamical systems of interest, and may be prone to over-fitting unless care is taken explicitly balance model complexity with predictive power. In [3], the Pareto front is used to isolate parsimonious models from a large family of candidate models. There are a host of additional techniques for modeling emergent behavior [4] and the discovery of governing equations from time-series data [5]. These include statistical methods of automated inference of dynamics [6, 7, 8], and equation-free modeling [9], including empirical dynamic modeling [10, 11].

In the present work, we re-envision the dynamical system discovery problem from the perspective of sparse regression [12, 13, 14] and compressive sensing [15, 16, 17, 18, 19, 20]. In particular, we leverage the fact that most physical systems have only a few nonlinear terms in the dynamics, making the right hand side of the equations *sparse* in a high-dimensional nonlinear function space. Before the advent of compressive sampling, and related sparsity-promoting methods, determining the few non-zero terms in a nonlinear dynamical system would have involved a combinatorial brute-force search, meaning that the methods would not scale to larger problems with Moore's law. However, powerful new theory guarantees that the sparse solution may be determined with high-probability using convex methods that do scale favorably with problem size. The resulting nonlinear model identification inherently balances model complexity (i.e., sparsity of right hand side dynamics) with accuracy, and the underlying convex optimization algorithms ensure that the method will be applicable to large-scale problems.

The method described here shares some similarity to the recent dynamic mode decomposition (DMD), which is a linear dynamic regression [21, 22]. DMD is an example of an equation-free method [9], since it only relies on measurement data, but not on knowledge of the governing equations. Recent advances in the extended DMD have developed rigorous connections between DMD built on nonlinear observable functions and the Koopman operator theory for nonlinear dynamical systems [21, 23]. However, there is currently no theory for which nonlinear observable functions to use, so that assumptions must be made on the form of the dynamical system. In contrast, the method developed here results in a *sparse, nonlinear* regression that automatically determines the relevant terms in the dynamical system. The trend to exploit sparsity in dynamical systems is recent but growing [24, 25, 26, 27, 28, 29, 30]. In this work, promoting sparsity in the dynamics results in parsimonious natural laws.

2 Background

This work combines methods from symbolic regression and sparse representation. Symbolic regression is used to find nonlinear functions describing the relationships between variables and measured dynamics (i.e., time derivatives). Traditionally, model complexity is balanced with describing capability using parsimony arguments such as the Pareto front. Here, we use sparse representation to determine the relevant model terms in an efficient and scalable framework.

2.1 Symbolic regression and machine learning

Symbolic regression involves the determination of a function that relates input–output data, and it may be viewed as a form of machine learning. Typically, the function is determined using genetic programming, which is an evolutionary algorithm that builds and tests candidate functions out of simple building blocks [31]. These functions are then modified according to a set of evolutionary rules and generations of functions are tested until a pre-determined accuracy is achieved.

Recently, symbolic regression has been applied to data from *dynamical* systems, and ordinary differential equations were discovered from measurement data [3]. Because it is possible to overfit with symbolic regression and genetic programming, a parsimony constraint must be imposed, and in [3], they accept candidate equations that are at the Pareto front of complexity.

2.2 Sparse representation and compressive sensing

In many regression problems, only a few terms in the regression are important, and a *sparse feature selection* mechanism is required. For example, consider data measurements $\mathbf{y} \in \mathbb{R}^m$ that may be a linear combination of columns from a feature library $\boldsymbol{\Theta} \in \mathbb{R}^{m \times p}$; the linear combination of columns is given by entries of the vector $\boldsymbol{\xi} \in \mathbb{R}^p$ so that:

$$\mathbf{y} = \boldsymbol{\Theta}\boldsymbol{\xi}. \tag{1}$$

Performing a standard regression to solve for $\boldsymbol{\xi}$ will result in a solution with nonzero contributions in each element. However, if *sparsity* of $\boldsymbol{\xi}$ is desired, so that most of the entries are zero, then it is possible to add an L^1 regularization term to the regression, resulting in the LASSO [12, 13, 14]:

$$\boldsymbol{\xi} = \underset{\boldsymbol{\xi}'}{\operatorname{argmin}} \|\boldsymbol{\Theta}\boldsymbol{\xi}' - \mathbf{y}\|_2 + \lambda\|\boldsymbol{\xi}'\|_1. \tag{2}$$

The parameter λ weights the sparsity constraint. This formulation is closely related to the compressive sensing framework, which allows for the sparse vector $\boldsymbol{\xi}$ to be determined from relatively few *incoherent* random measurements [15, 16, 17, 18, 19, 20]. The sparse solution $\boldsymbol{\xi}$ to Eq. 1 may also be used for sparse classification schemes. Importantly, the compressive sensing and sparse representation architectures are convex and scale well to large problems, as opposed to brute-force combinatorial alternatives.

3 Nonlinear system identification using sparse representation

In this work, we are concerned with identifying the governing equations that underly a physical system based on data that may be realistically collected in simulations or experiments. Generically, we seek to represent the system as a nonlinear dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)). \tag{3}$$

The vector $\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_n(t) \end{bmatrix}^T \in \mathbb{R}^n$ represents the state of the system at time t, and the nonlinear function $\mathbf{f}(\mathbf{x}(t))$ represents the dynamic constraints that define the equations of motion of the system. In the following sections, we will generalize Eq. (3) to allow the dynamics $\mathbf{f}$ to vary in time, and also with respect to a set of bifurcation parameters $\boldsymbol{\mu} \in \mathbb{R}^q$.

The key observation in this paper is that for many systems of interest, the function $\mathbf{f}$ often consists of only a few terms, making it sparse in the space of possible functions. For example, the Lorenz system in Eq. (22c) has very few terms in the space of polynomial functions. Recent advances in compressive sensing and sparse regression make this viewpoint of sparsity favorable, since it is now possible to determine *which* right hand side terms are non-zero without performing a computationally intractable brute-force search.

To determine the function $\mathbf{f}$ from data, we collect a time-history of the state $\mathbf{x}(t)$ and either measure the derivative $\dot{\mathbf{x}}(t)$ or approximate it numerically from $\mathbf{x}$. The data is sampled at several times $t_1, t_2, \cdots, t_m$ and arranged into two large matrices:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \overbrace{\begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix}}^{\text{state}} \Bigg\downarrow \text{time} \tag{4a}$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix}. \tag{4b}$$

Next, we construct an augmented library $\boldsymbol{\Theta}(\mathbf{X})$ consisting of candidate nonlinear functions of the columns of $\mathbf{X}$. For example, $\boldsymbol{\Theta}(\mathbf{X})$ may consist of constant, polynomial and trigonometric terms:

$$\boldsymbol{\Theta}(\mathbf{X}) = \begin{bmatrix} | & | & | & | & & | & | & | & | & \\ \mathbf{1} & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \sin(2\mathbf{X}) & \cos(2\mathbf{X}) & \cdots \\ | & | & | & | & & | & | & | & | & \end{bmatrix}. \tag{5}$$

Here, higher polynomials are denoted as $\mathbf{X}^{P_2}, \mathbf{X}^{P_3}$, etc. For example, $\mathbf{X}^{P_2}$ denotes the quadratic nonlinearities in the state variable $\mathbf{x}$, given by:

$$\mathbf{X}^{P_2} = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^2(t_1) & x_2(t_1)x_3(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_2^2(t_2) & x_2(t_2)x_3(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^2(t_m) & x_2(t_m)x_3(t_m) & \cdots & x_n^2(t_m) \end{bmatrix}. \tag{6}$$

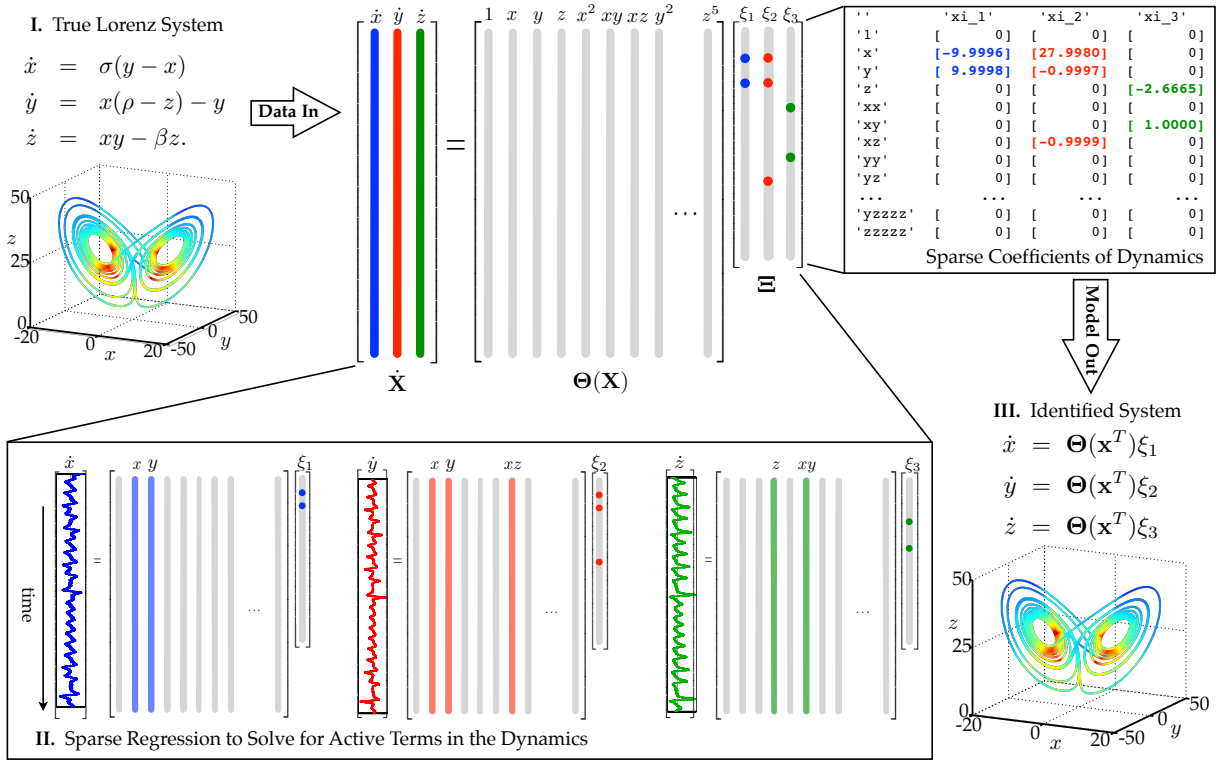

Figure 1: Schematic of sparse dynamic representation, demonstrated on the Lorenz equations. Data is collected from measurements of a complex system, including states $\mathbf{X} = (x, y, z)$ and derivatives $\dot{\mathbf{X}} = (\dot{x}, \dot{y}, \dot{z})$. Next, a large collection of nonlinear functions of the states, $\mathbf{\Theta}(\mathbf{X})$ is constructed. This nonlinear feature library is used to find the fewest dynamics terms to satisfy $\dot{\mathbf{X}} = \mathbf{\Theta}(\mathbf{X})\mathbf{\Xi}$, resulting in a sparse model.

Each column of $\mathbf{\Theta}(\mathbf{X})$ represents a candidate function for the right hand side of Eq. (3). There is tremendous freedom of choice in constructing the entries in this matrix of nonlinearities. Since we believe that only a few of these nonlinearities are active in each row of $\mathbf{f}$, we may set up a sparse regression problem to determine the sparse vectors of coefficients $\mathbf{\Xi} = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \cdots & \boldsymbol{\xi}_n \end{bmatrix}$ that determine which nonlinearities are active, as illustrated in Fig. 1.

$$\dot{\mathbf{X}} = \mathbf{\Theta}(\mathbf{X})\mathbf{\Xi}. \tag{7}$$

Each column $\boldsymbol{\xi}_k$ of $\mathbf{\Xi}$ represents a sparse vector of coefficients determining which terms are active in the right hand side for one of the row equations $\dot{\mathbf{x}}_k = \mathbf{f}_k(\mathbf{x})$ in Eq. (3). Once $\mathbf{\Xi}$ has been determined, a model of each row of the governing equations may be constructed as follows:

$$\dot{\mathbf{x}}_k = \mathbf{f}_k(\mathbf{x}) = \mathbf{\Theta}(\mathbf{x}^T)\boldsymbol{\xi}_k. \tag{8}$$

Note that $\mathbf{\Theta}(\mathbf{x}^T)$ is a vector of symbolic functions of elements of $\mathbf{x}$, as opposed to $\mathbf{\Theta}(\mathbf{X})$, which is a data matrix. This results in the overall model

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \mathbf{\Xi}^T(\mathbf{\Theta}(\mathbf{x}^T))^T. \tag{9}$$

We may solve for $\mathbf{\Xi}$ in Eq. (7) using sparse regression. In many cases, we may need to normalize the columns of $\mathbf{\Theta}(\mathbf{X})$ first to ensure that the restricted isometry property holds [24]; this is especially important when the entries in $\mathbf{X}$ are small, since powers of $\mathbf{X}$ will be minuscule.

3.1 Algorithm for sparse representation of dynamics with noise

There are a number of algorithms to determine sparse solutions $\boldsymbol{\Xi}$ to the regression problem in Eq. (7). Each column of Eq. (7) requires a distinct optimization problem to find the sparse vector of coefficients $\boldsymbol{\xi}_k$ for the k^{th} row equation.

For the examples in this paper, the matrix $\boldsymbol{\Theta}(\mathbf{X})$ has dimensions $m \times p$, where p is the number of candidate nonlinear functions, and where $m \gg p$ since there are more time samples of data than there are candidate nonlinear functions. Realistically, often only the data $\mathbf{X}$ is available, and $\dot{\mathbf{X}}$ must be approximated numerically, as in the examples below. Thus, both $\mathbf{X}$ and $\dot{\mathbf{X}}$ will be contaminated with noise so that Eq. (7) does not hold exactly. Instead,

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi} + \eta\mathbf{Z}, \tag{10}$$

where $\mathbf{Z}$ is a matrix of independent identically distributed Gaussian entries with zero mean, and η is the noise magnitude. Thus we seek a sparse solution to an overdetermined system with noise.

The LASSO [12, 14] from statistics works well with this type of data, providing a sparse regression. However, it may be computationally expensive for very large data sets.

An alternative is to implement the sequential thresholded least-squares algorithm in Code (1). In this algorithm, we start with a least-squares solution for $\boldsymbol{\Xi}$ and then threshold all coefficients that are smaller than some cutoff value λ. Once the indices of the remaining non-zero coefficients are identified, we obtain another least-squares solution for $\boldsymbol{\Xi}$ onto the remaining indices. These new coefficients are again thresholded using λ, and the procedure is continued until the non-zero coefficients converge. This algorithm is computationally efficient, and it rapidly converges to a sparse solution in a small number of iterations. The algorithm also benefits from simplicity, with a single parameter λ required to determine the degree of sparsity in $\boldsymbol{\Xi}$.

Depending on the noise, it may be necessary to filter $\mathbf{X}$ and $\dot{\mathbf{X}}$ before solving for $\boldsymbol{\Xi}$. In many of the examples below, only the data $\mathbf{X}$ is available, and $\dot{\mathbf{X}}$ is obtained by differentiation. To counteract differentiation error, we use the total variation regularized derivative [32] to de-noise the derivative; this is based on the total variation regularization [33]. This works quite well when only state data $\mathbf{X}$ is available, as illustrated on the Lorenz system in Fig. 7. Alternatively, the data $\mathbf{X}$ and $\dot{\mathbf{X}}$ may be filtered, for example using the optimal hard threshold for singular values described in [34]. It is important to note that previous algorithms to identify dynamics from data have been quite sensitive to noise [3, 24]. The algorithm in Code 1 is remarkably robust to noise, even when derivatives must be approximated from noisy data.

Code 1: Sparse representation algorithm in Matlab.

```
%% compute Sparse regression: sequential least squares
Xi = Theta\dXdt;  % initial guess: Least-squares

% lambda is our sparsification knob.
for k=1:10
    smallinds = (abs(Xi)<lambda);   % find small coefficients
    Xi(smallinds)=0;                % and threshold
    for ind = 1:n                   % n is state dimension
        biginds = ~smallinds(:,ind);
        % Regress dynamics onto remaining terms to find sparse Xi
        Xi(biginds,ind) = Theta(:,biginds)\dXdt(:,ind);
    end
end
```

3.2 Cross-validation to determine parsimonious sparse solution on Pareto front

To determine the sparsification parameter λ in the algorithm in Code (1), it is helpful to use the concept of cross-validation from machine learning. It is always possible to hold back some test data apart from the training data to test the validity of models away from training values. In addition, it is important to consider the balance of model complexity (given by the number of nonzero coefficients in $\boldsymbol{\Xi}$) with the model accuracy. There is an "elbow" in the curve of accuracy vs. complexity parameterized by λ, the so-called Pareto front. This value of λ represents a good tradeoff between complexity and accuracy, and it is similar to the approach taken in [3].

3.3 Extensions and Connections

There are a number of extensions to the basic theory above that generalize this approach to a broader set of problems. First, the method is generalized to a discrete-time formulation, establishing a connection with the dynamic mode decomposition (DMD). Next, high-dimensional systems obtained from discretized partial differential equations are considered, extending the method to incorporate dimensionality reduction techniques to handle big data. Finally, the sparse regression framework is modified to include bifurcation parameters, time-dependence, and external forcing.

3.3.1 Discrete-time representation

The aforementioned strategy may also be implemented on discrete-time dynamical systems:

$$\mathbf{x}_{k+1} \quad = \quad \mathbf{f}(\mathbf{x}_k). \tag{11}$$

There are a number of reasons to implement Eq. (11). First, many systems, such as the logistic map in Eq. (26) are inherently discrete-time systems. In addition, it may be possible to recover specific integration schemes used to advance Eq. (3). The discrete-time formulation also foregoes the calculation of a derivative from noisy data. The data collection will now involve two matrices $\mathbf{X}_1^{m-1}$ and $\mathbf{X}_2^m$:

$$\mathbf{X}_1^{m-1} = \begin{bmatrix} \text{———} & \mathbf{x}_1^T & \text{———} \\ \text{———} & \mathbf{x}_2^T & \text{———} \\ & \vdots & \\ \text{———} & \mathbf{x}_{m-1}^T & \text{———} \end{bmatrix}, \qquad \mathbf{X}_2^m = \begin{bmatrix} \text{———} & \mathbf{x}_2^T & \text{———} \\ \text{———} & \mathbf{x}_3^T & \text{———} \\ & \vdots & \\ \text{———} & \mathbf{x}_m^T & \text{———} \end{bmatrix}. \tag{12}$$

The continuous-time sparse regression problem in Eq. (7) now becomes:

$$\mathbf{X}_2^m = \boldsymbol{\Theta}(\mathbf{X}_1^{m-1})\boldsymbol{\Xi} \tag{13}$$

and the function $\mathbf{f}$ is the same as in Eq. (9).

In the discrete setting in Eq. (11), and for linear dynamics, there is a striking resemblance to dynamic mode decomposition. In particular, if $\boldsymbol{\Theta}(\mathbf{x}) = \mathbf{x}$, so that the dynamical system is linear, then Eq. (13) becomes

$$\mathbf{X}_2^m = \mathbf{X}_1^{m-1}\boldsymbol{\Xi} \quad \Longrightarrow \quad \left(\mathbf{X}_2^m\right)^T = \boldsymbol{\Xi}^T \left(\mathbf{X}_1^{m-1}\right)^T. \tag{14}$$

This is equivalent to the DMD, which seeks a dynamic regression onto linear dynamics $\boldsymbol{\Xi}^T$. In particular, $\boldsymbol{\Xi}^T$ is $n \times n$ dimensional, which may be prohibitively large for a high-dimensional state $\mathbf{x}$. Thus, DMD identifies the dominant terms in the eigendecomposition of $\boldsymbol{\Xi}^T$.

3.3.2 High-dimensional systems, partial differential equations, and dimensionality reduction

Often, the physical system of interest may be naturally represented by a partial differential equation (PDE) in a few spatial variables. If data is collected from a numerical discretization or from experimental measurements on a spatial grid, then the state dimension n may be prohibitively large. For example, in fluid dynamics, even simple two-dimensional and three-dimensional flows may require tens of thousands up to billions of variables to represent the discretized system.

The method described above is prohibitive for a large state dimension n, both because of the factorial growth of $\boldsymbol{\Theta}$ in n and because each of the n row equations in Eq. (8) requires a separate optimization. Fortunately, many high-dimensional systems of interest evolve on a low-dimensional manifold or attractor that may be well-approximated using a dimensionally reduced low-rank basis $\boldsymbol{\Psi}$. For example, if data $\mathbf{X}$ is collected for a high-dimensional system as in Eq. (4a), it is possible to obtain a low-rank approximation using the singular value decomposition (SVD):

$$\mathbf{X}^T = \boldsymbol{\Psi}\boldsymbol{\Sigma}\mathbf{V}^*. \tag{15}$$

In this case, the state $\mathbf{x}$ may be well approximated in a truncated modal basis $\boldsymbol{\Psi}_r$, given by the first r columns of $\boldsymbol{\Psi}$ from the SVD:

$$\mathbf{x} \approx \boldsymbol{\Psi}_r \mathbf{a}, \tag{16}$$

where $\mathbf{a}$ is an r-dimensional vector of mode coefficients. We assume that this is a good approximation for a relatively low rank r. Thus, instead of using the original high-dimensional state $\mathbf{x}$, it is possible to obtain a sparse representation of the Galerkin projected dynamics $\mathbf{f}_P$ in terms of the coefficients $\mathbf{a}$:

$$\dot{\mathbf{a}} = \mathbf{f}_P(\mathbf{a}). \tag{17}$$

There are many choices for a low-rank basis, including proper orthogonal decomposition (POD) [35, 36], based on the SVD.

3.3.3 External forcing, bifurcation parameters, and normal forms

It is also possible to identify normal forms associated with a bifurcation parameter $\boldsymbol{\mu}$ by suspending the parameter as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu}) \tag{18a}$$

$$\dot{\boldsymbol{\mu}} = 0. \tag{18b}$$

Here we consider the bifurcation parameter $\boldsymbol{\mu}$ as a variable with zero time derivative. It is then possible to identify the right hand side $\mathbf{f}(\mathbf{x}; \boldsymbol{\mu})$ as a sparse combination of functions of components in $\mathbf{x}$ as well as the bifurcation parameter $\boldsymbol{\mu}$. This idea is illustrated on two examples, the one-dimensional logistic map and the two-dimensional Hopf normal form. This is an important generalization, since it is now possible to identify a normal form from data, allowing for the prediction of bifurcation phenomena and dynamic unfoldings [2].

Similarly, time-dependence may be added to the vector field by suspending the time variable:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \tag{19a}$$

$$\dot{t} = 1. \tag{19b}$$

This includes both time-varying vector fields as well as external forcing terms.

4 Results

The methods described in Sec. 3 to identify governing equations from data are now demonstrated on a number of example systems of varying complexity. The first example illustrates the method on simple systems including a comparison of two-dimensional linear vs. nonlinear damped oscillators, as well as a three-dimensional stable linear system. In the second example, the chaotic Lorenz system is investigated, and the sparse identification algorithm accurately reproduces the form of the governing equations, and hence the attractor dynamics. The third example demonstrates the extension of this method to nonlinear partial differential equations (PDEs) by investigating the fluid flow past a circular cylinder at Reynolds number 100. In this example, data from direct numerical simulation of the Navier-Stokes equations are used to obtain a low-dimensional proper orthogonal decomposition (POD) subspace. In POD coordinates, the identified system accurately captures limit cycle dynamics as well as transients. Importantly, the identified nonlinear terms are quadratic, which is consistent with the form of the Navier-Stokes equations; thus the mean-field model captures the subtle slow-manifold dynamics of the system. In the fourth example, normal forms are identified for both the logistic map and the Hopf normal form; in both cases, the model is correctly parameterized. The examples in Sections 4.2-4.4 approximate derivatives $\dot{\mathbf{x}}$ from noisy state measurements $\mathbf{x}$ using the total-variation regularized derivative [32].

4.1 Example 1: Simple illustrative systems

4.1.1 Example 1a: Two-dimensional damped oscillator (linear vs. nonlinear)

In this example, we consider the two-dimensional damped harmonic oscillator with either linear dynamics, as in Eq. (20b), or with cubic dynamics, as in Eq. (20b). The dynamic data and the sparse identified model are shown in Fig. 2. The correct form of the nonlinearity is obtained in each case; the augmented nonlinear library $\mathbf{\Theta}(\mathbf{x})$ includes polynomials in $\mathbf{x}$ up to fifth order. The sparse identified model and algorithm parameters are shown in the Appendix in Tables 6 and 7.

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -0.1 & 2 \\ -2 & -0.1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \tag{20a}$$

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -0.1 & 2 \\ -2 & -0.1 \end{bmatrix}\begin{bmatrix} x^3 \\ y^3 \end{bmatrix} \tag{20b}$$

4.1.2 Example 1b: Three-dimensional linear system

A linear system with three variables and the sparse approximation are shown in Fig. 3. In this case, the dynamics are given by

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -0.1 & -2 & 0 \\ 2 & -0.1 & 0 \\ 0 & 0 & -0.3 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{21}$$

The sparse identification algorithm correctly identifies the system in the space of polynomials up to second or third order, and the sparse model is given in Table 8. Interestingly, including polynomial terms of higher order (e.g. orders 4 or 5) introduces a degeneracy in the sparse identification algorithm, because linear combinations of powers of $e^{\lambda t}$ may approximate other exponential rates. This unexpected degeneracy motivates a hierarchical approach to identification, where subsequently higher order terms are included until the algorithm either converges or diverges.

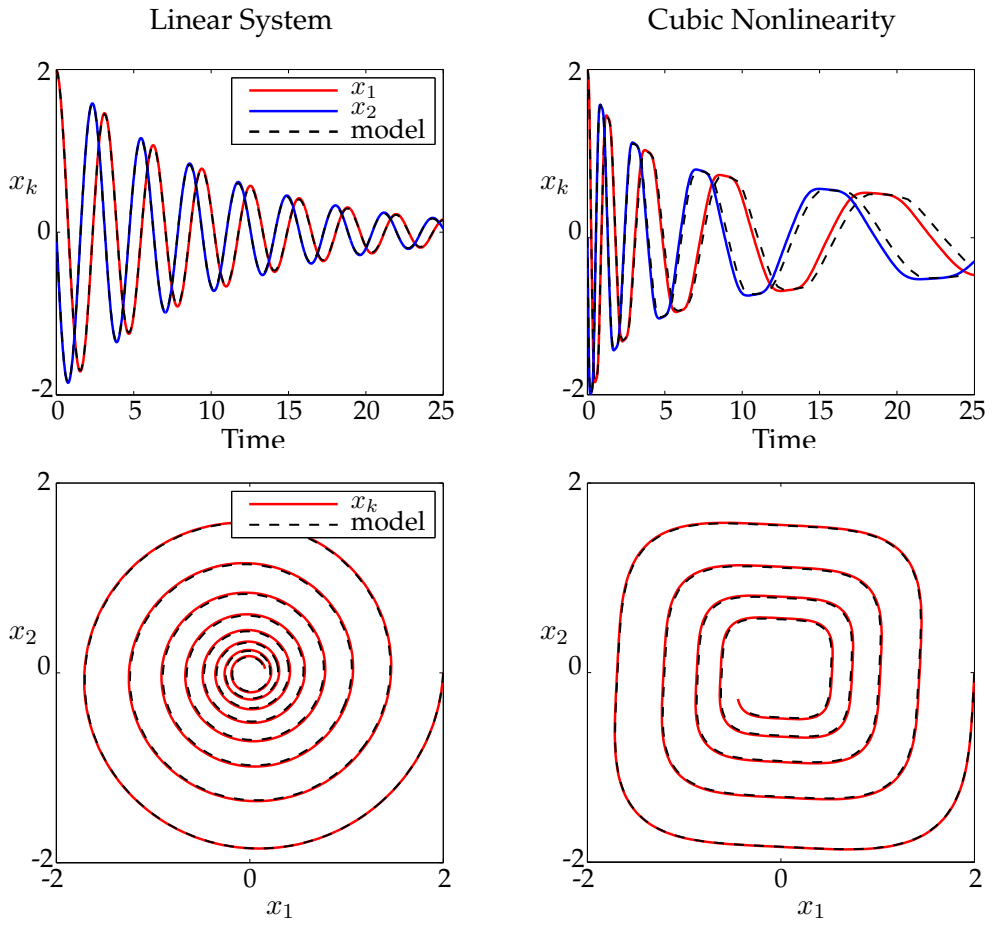

Figure 2: Comparison of linear system (left) against system with cubic nonlinearity (right). The sparse identified system correctly identifies the form of the dynamics and accurately reproduces the phase portraits.

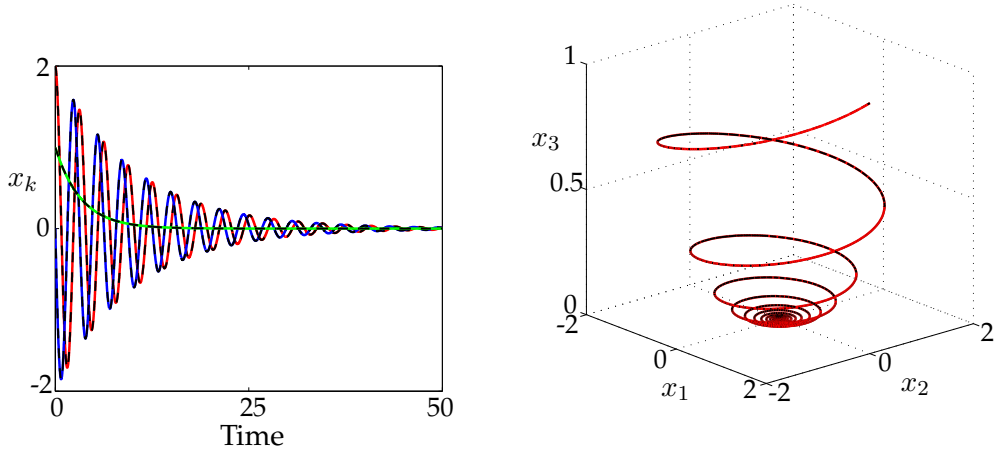

Figure 3: Three-dimensional linear system (solid colored lines) is well-captured by sparse identified system (dashed black line).

4.2 Example 2: Lorenz system (Nonlinear ODE)

Here, we consider the nonlinear Lorenz system [37] to explore the identification of chaotic dynamics evolving on an attractor:

$$\dot{x} = \sigma(y - x) \tag{22a}$$
$$\dot{y} = x(\rho - z) - y \tag{22b}$$
$$\dot{z} = xy - \beta z. \tag{22c}$$

For this example, we use the standard parameters $\sigma = 10, \beta = 8/3, \rho = 28$, with an initial condition $\begin{bmatrix} x & y & z \end{bmatrix}^T = \begin{bmatrix} -8 & 7 & 27 \end{bmatrix}^T$. Data is collected from $t = 0$ to $t = 100$ with a time-step of $\Delta t = 0.001$. The system is identified in the space of polynomials in (x, y, z) up to fifth order:

$$\mathbf{\Theta}(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & | & | & | & | & | & | \\ \mathbf{x}(t) & \mathbf{y}(t) & \mathbf{z}(t) & \mathbf{x}(t)^2 & \mathbf{x}(t)\mathbf{y}(t) & \mathbf{x}(t)\mathbf{z}(t) & \mathbf{y}(t)^2 & \mathbf{y}(t)\mathbf{z}(t) & \mathbf{z}(t)^2 & \cdots & \mathbf{z}(t)^5 \\ | & | & | & | & | & | & | & | & | & | & | \end{bmatrix}. \tag{23}$$

To explore the effect of noisy derivatives in a controlled setting, we add zero-mean Gaussian measurement noise with variance η to the exact derivatives. The short-time ($t = 0$ to $t = 20$) and long-time ($t = 0$ to $t = 250$) system reconstruction is shown in Fig. 4 for two different noise values, $\eta = 0.01$ and $\eta = 10$. The trajectories are also shown in dynamo view in Fig. 5, and the ℓ_2 error vs. time for increasing noise η is shown in Fig. 6. Although the ℓ_2 error increases for large noise values η, the form of the equations, and hence the attractor dynamics, are accurately captured. The system has a positive Lyapunov exponent, and small differences in model coefficients or initial conditions grow exponentially, until saturation, even though the attractor remains intact.

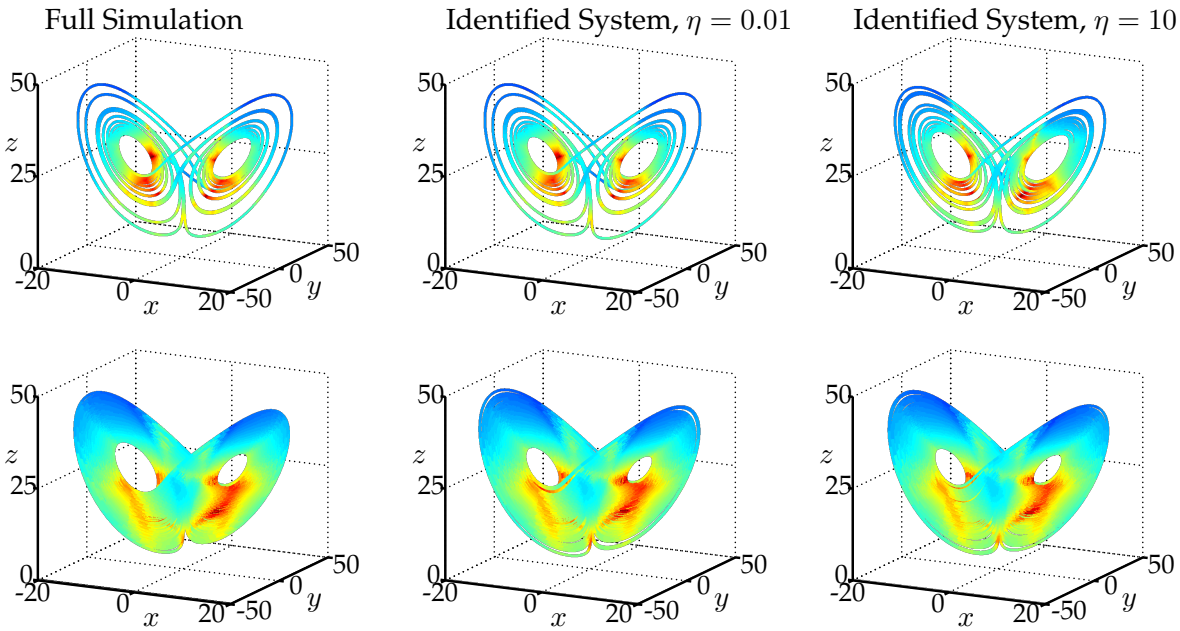

Figure 4: Trajectories of the Lorenz system for short-time integration from $t = 0$ to $t = 20$ (top) and long-time integration from $t = 0$ to $t = 250$ (bottom). The full dynamics (left) are compared with the sparse identified systems (middle, right) for various additive noise, assuming measurements of $\mathbf{x}$ and $\dot{\mathbf{x}}$. The trajectories are colored by Δt, the adaptive Runge-Kutta time step. This color is a proxy for local sensitivity.

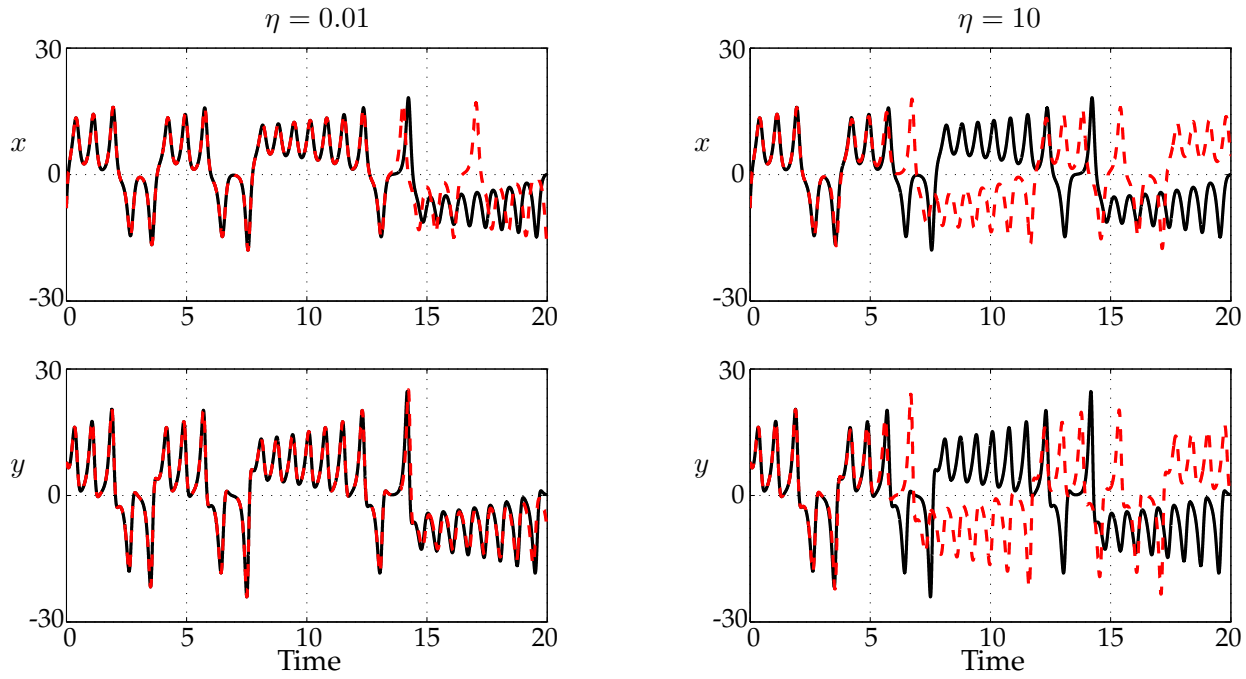

Figure 5: Dynamo view of trajectories of the Lorenz system for the illustrative case where $\mathbf{x}$ and $\dot{\mathbf{x}}$ are measured with noise. The exact system is shown in black ($-$) and the sparse identified system is shown in the dashed red arrow ($--$).

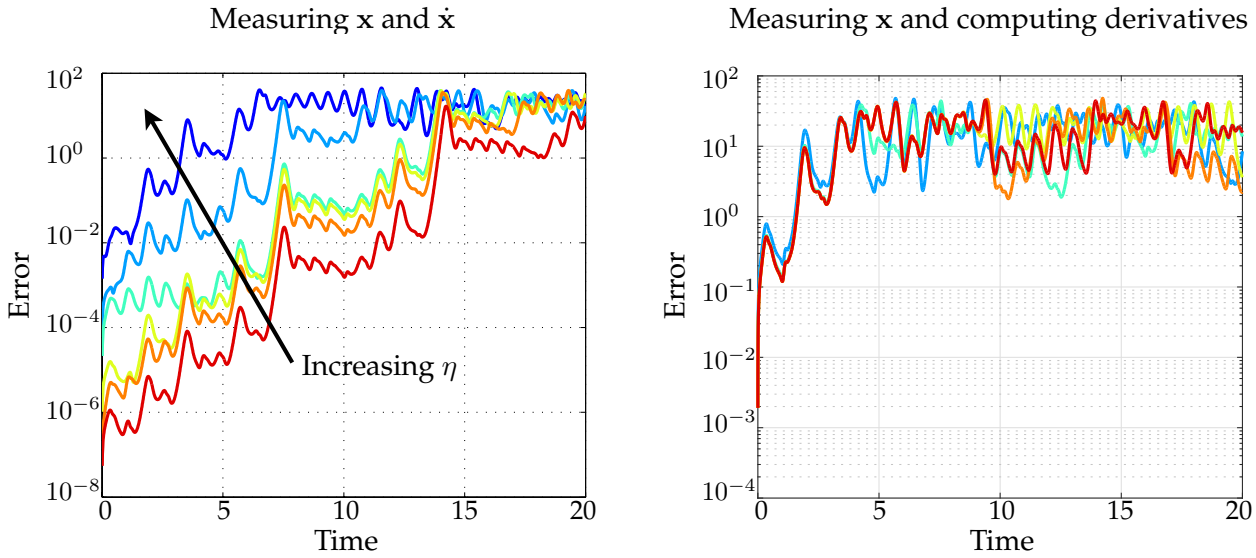

Figure 6: Error vs. time for sparse identified systems generated from data with increasing noise magnitude η. When $\mathbf{x}$ and $\dot{\mathbf{x}}$ are measured, then noise is added to the derivative (left). This error corresponds to the difference between solid black and dashed red curves in Fig. 5. Sensor noise values are $\eta \in \{0.0001, 0.001, 0.01, 0.1, 1.0, 10.0\}$. When only $\mathbf{x}$ is measured, noise is added to the state, and the derivatives $\dot{\mathbf{x}}$ are computed using the total variation regularized derivative [32] (right). In this case, the largest noise magnitude $\eta = 10.0$ is omitted, because the approximation fails.

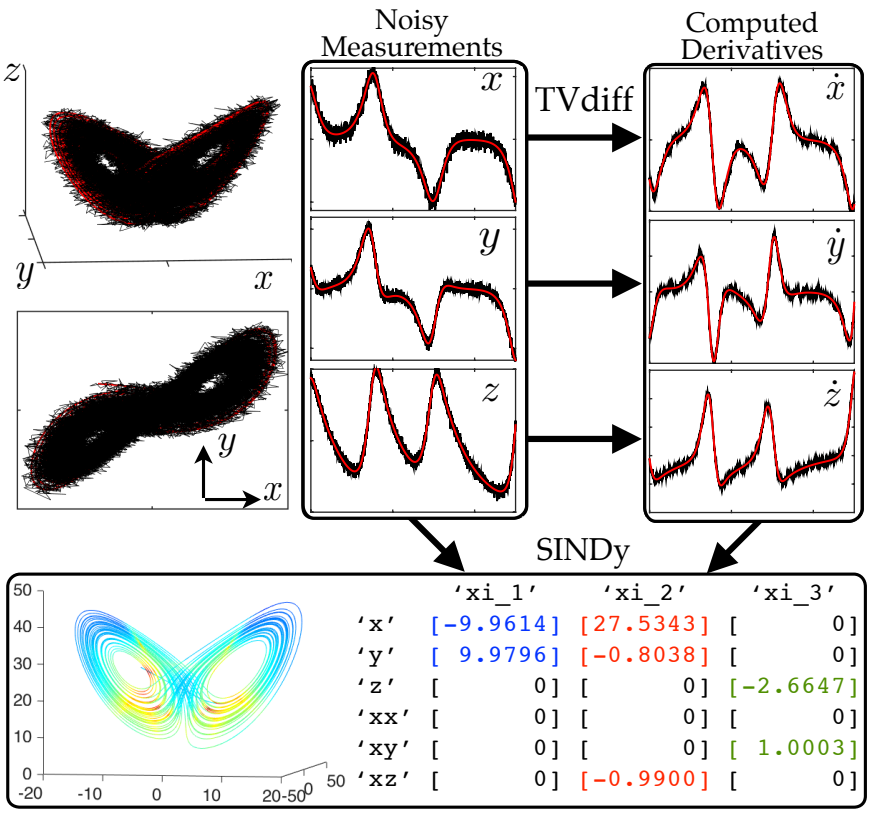

Figure 7: SINDy procedure when only noisy state measurements are available for the Lorenz system. Gaussian noise with $\sigma = 1$ is added to the state, and derivatives are computed using the total variation derivative [32]. The exact system without noise is shown in red, and the noisy measurements and approximated derivatives are shown in black.

Next, we explore the SINDy algorithm on the Lorenz equation when only noisy measurements of the state $\mathbf{x}$ are available. Gaussian noise with variance η is added to the state $\mathbf{x}$, and derivatives $\dot{\mathbf{x}}$ are computed using the total-variation regularized derivative [32]. This procedure is illustrated for a relatively large noise magnitude $\eta = 1.0$ in Fig. 7. The correct terms are identified, and the attractor is captured by these sparse identified dynamics. A systematic investigation for varying η is shown in the right panel of Fig. 6. Again, the ℓ_2 error rapidly grows because of the chaotic nature of the Lorenz attractor, so that individual trajectories rapidly diverge. However, even for large noise magnitudes, the attractor dynamics are captured.

We also explore the ability to capture the attractor dynamics using time-delay coordinates when incomplete measurements are taken. This extension is presented in Section 4.5.

4.3 Example 3: Fluid wake behind a cylinder (Nonlinear PDE)

Here we demonstrate the generalization of the sparse dynamics method to partial differential equations (PDEs). Data is collected for the fluid flow past a cylinder at Reynolds number 100 using direct numerical simulations of the two-dimensional Navier-Stokes equations [38, 39]. Then, the dynamic relationship between low-rank coherent structures is determined.

The low-Reynolds number flow past a cylinder is a particularly interesting example because of its rich history in fluid mechanics and dynamical systems. It has long been theorized that turbulence may be the result of a sequence of Hopf bifurcations that occur as the Reynolds number of the flow increases [40]. The Reynolds number is a rough measure of the ratio of inertial and viscous forces, and an increasing Reynolds number may correspond, for example, to increasing flow velocity, giving rise to more rich and intricate structures in the fluid.

It took roughly 15 years to find the first Hopf bifurcation in fluid mechanics, in the transition from a laminar steady wake to laminar periodic vortex shedding behind a cylinder at Reynolds number 47 [41, 42]. This discovery led to another long-standing debate about how a Hopf bifurcation, with cubic nonlinearity, can be exhibited in a Navier-Stokes fluid with quadratic nonlinearities. After 15 more years, this issue was finally resolved using a separation of time-scales argument and a mean-field model [43]. It was demonstrated that coupling between oscillatory wake modes with the base flow gives rise to a slow manifold (see Fig. 8), and this slow manifold produces algebraic terms that approximate cubic nonlinearities on slow timescales.

This example provides a compelling test-case for the proposed algorithm, since the underlying form of the dynamics took nearly three decades to uncover. Indeed, the sparse dynamics algorithm identifies the on-attractor and off-attractor dynamics using quadratic nonlinearities and reproduces a parabolic slow manifold. It is interesting to note that when the off-attractor trajectories are not included in the system identification, the algorithm incorrectly identifies the dynamics using cubic nonlinearities, and fails to correctly identify the dynamics associated with the shift mode, which connects the mean flow to the unstable steady state.

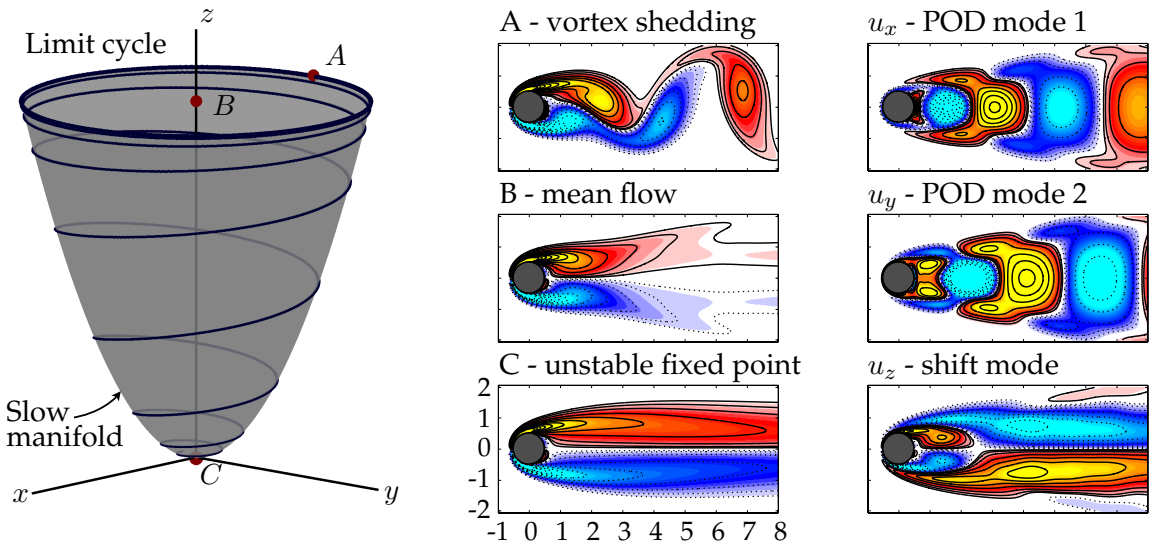

Figure 8: Illustration of the low-rank dynamics underlying the periodic vortex shedding behind a circular cylinder at low Reynolds number, Re $= 100$.

4.3.1 Direct numerical simulation

The direct numerical simulation involves a fast multi-domain immersed boundary projection method [38, 39]. Four grids are used, each with a resolution of 450×200, with the finest grid having dimensions of 9×4 cylinder diameters and the largest grid having dimensions of 72×32 diameters. The finest grid has 90,000 points, and each subsequent coarser grid has 67,500 distinct points. Thus, if the state includes the vorticity at each grid point, then the state dimension is 292,500. The vorticity field on the finest grid is shown in Fig. 8. The code is non-dimensionalized so that the cylinder diameter and free-stream velocity are both equal to one: $D = 1$ and $U_\infty = 1$, respectively. The simulation time-step is $\Delta t = 0.02$ non dimensional time units.

4.3.2 Mean field model

To develop a mean-field model for the cylinder wake, first we must reduce the dimension of the system. The proper orthogonal decomposition (POD) [36], provides a low-rank basis that is optimal in the L^2 sense; for fluid velocity fields, the POD results in a hierarchy of orthonormal modes that, when truncated, capture the most energy of the original system for the given rank truncation. The first two most energetic POD modes capture a significant portion of the energy; the steady-state vortex shedding is a limit cycle in these coordinates. An additional mode, called the shift mode, is included to capture the transient dynamics connecting the unstable steady state with the mean of the limit cycle [43].

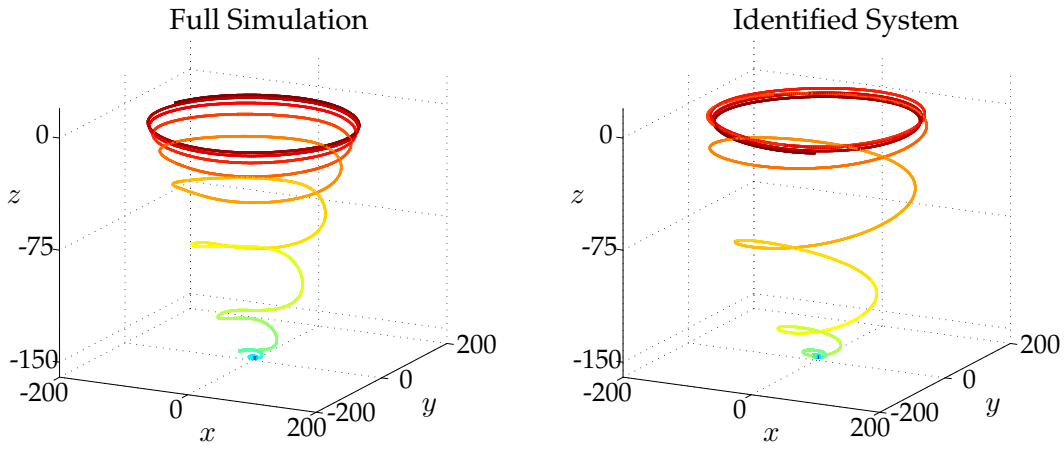

Figure 9: Evolution of the cylinder wake trajectory in reduced coordinates. The full simulation (left) comes from direct numerical simulation of the Navier-Stokes equations, and the identified system (right) captures the dynamics on the slow manifold. Color indicates simulation time.

In the three-dimensional coordinate system described above, the mean-field model for the cylinder dynamics are given by:

$$\dot{x} = \mu x - \omega y + Axz \tag{24a}$$
$$\dot{y} = \omega x + \mu y + Ayz \tag{24b}$$
$$\dot{z} = -\lambda(z - x^2 - y^2). \tag{24c}$$

If λ is large, so that the z-dynamics are fast, then the mean flow rapidly corrects to be on the (slow) manifold $z = x^2 + y^2$ given by the amplitude of vortex shedding. When substituting this algebraic relationship into Eqs. 24a and 24b, we recover the Hopf normal form on the slow manifold.

Remarkably, similar dynamics are discovered by the sparse dynamics algorithm, purely from data collected from simulations. The identified model coefficients, shown in Table 10, only include quadratic nonlinearity, consistent with the Navier-Stokes equations. Moreover, the transient behavior, shown in Figs. 10 and 11, is captured qualitatively for solutions that do not start on the slow manifold. When the off-attractor dynamics in Fig. 10 are not included in the training data, then the model only recovers a simple Hopf normal form in x and y with cubic terms, but does not correctly identify the slow-manifold with quadratic nonlinearities. Note that time derivatives of the POD coefficients are approximated numerically using a fourth order central difference scheme.

The data from Fig. 11 was not included in the training data, and although qualitatively similar, the identified model does not exactly reproduce the transients. Since this initial condition had twice the fluctuation energy in the x and y directions, the slow manifold approximation may not be valid here. Finally, reducing the sparsifying parameter λ, it is possible to obtain models that agree almost perfectly with the data in Figs. 9-11, although the model will then include higher order nonlinearities; this is discussed in Sec. 4.3.3.

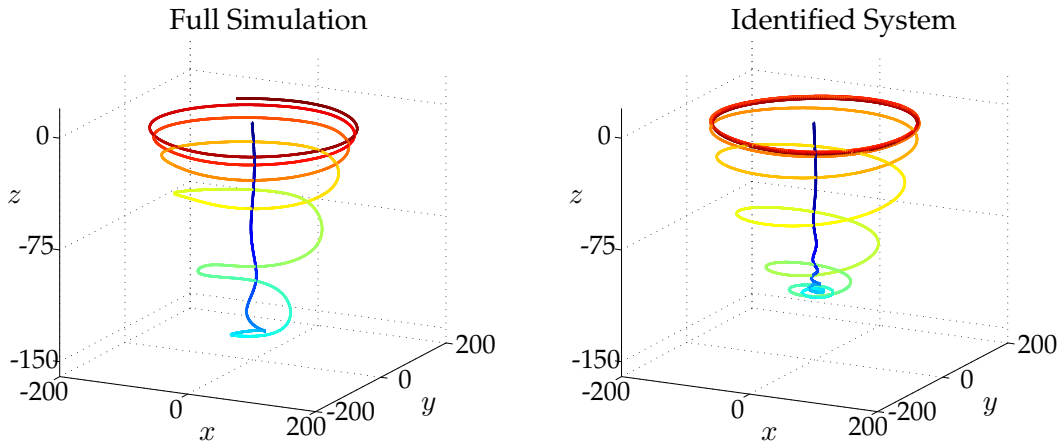

Figure 10: Evolution of the cylinder wake trajectory starting from a flow state initialized at the mean of the steady-state limit cycle. Both the full simulation and sparse model capture the off-attractor dynamics, characterized by rapid attraction of the trajectory onto the slow manifold.

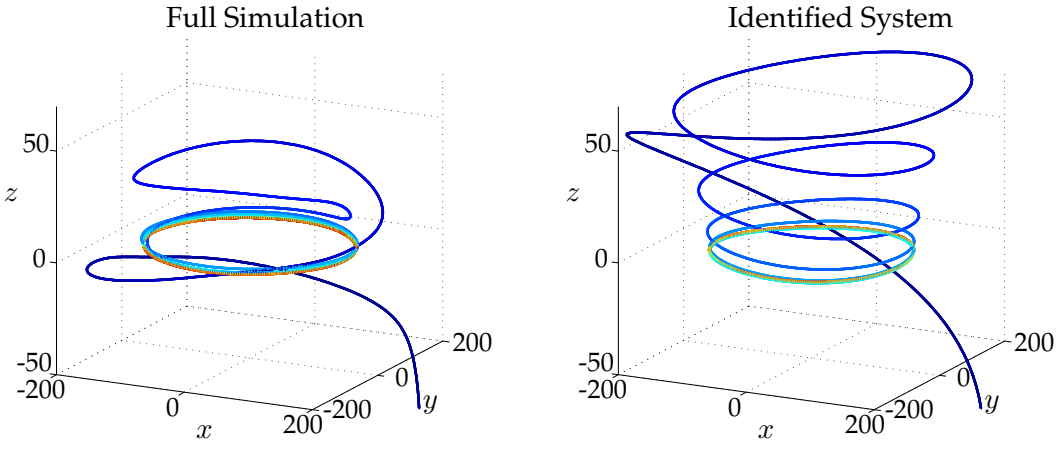

Figure 11: This simulation corresponds to an initial condition obtained by doubling the magnitude of the limit cycle behavior. This data was not included in the training of the sparse model.

4.3.3 Cubic nonlinearities

It is important to note that although the nonlinear system in Figs. 9 and 10 is identified using only quadratic nonlinearities, they also contain constant forcing terms, which introduce an extra spurious fixed point in the z direction. If we decrease the sparsifying parameter λ, so that we obtain a model that also includes cubic nonlinearities, the identified system is more accurate in terms of the dynamic response and does not posses this spurious extra fixed point. This is not entirely surprising that the quadratic response has limitations when only using three POD modes, since there is additional energy captured by higher POD pairs.

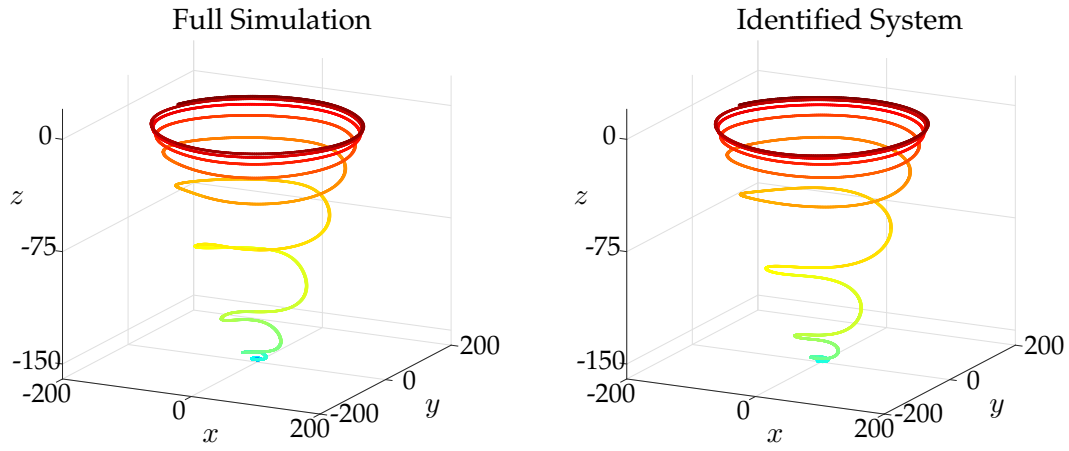

Figure 12: Evolution of the cylinder wake trajectory in reduced coordinates. The full simulation (left) comes from direct numerical simulation of the Navier-Stokes equations, and the identified system (right) captures the dynamics on the slow manifold. Color indicates simulation time.

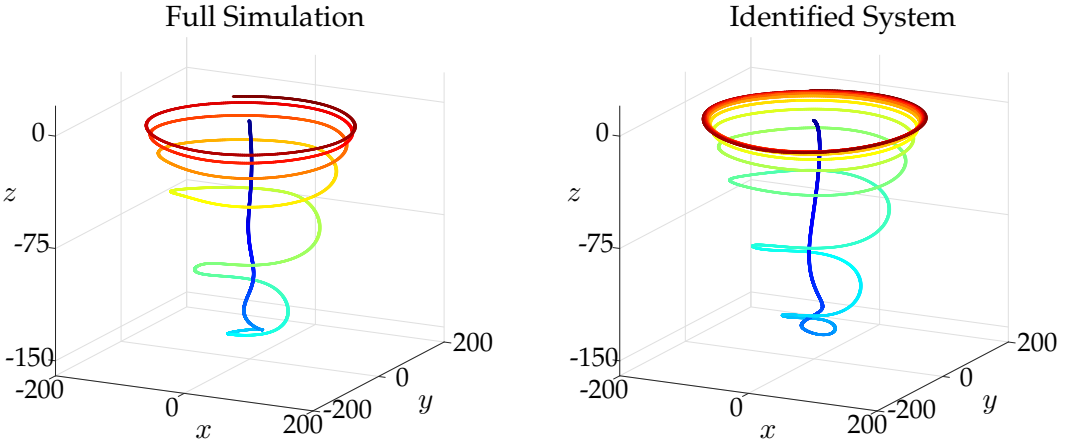

Figure 13: Evolution of the cylinder wake trajectory starting from a flow state initialized at the mean of the steady-state limit cycle. Both the full simulation and sparse model capture the off-attractor dynamics, characterized by rapid attraction of the trajectory onto the slow manifold.

4.4 Example 4: Bifurcations and normal forms

It is also possible to identify normal forms associated with a bifurcation parameter μ by suspending it in the dynamics as a variable:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \mu) \tag{25a}$$

$$\dot{\mu} = 0. \tag{25b}$$

It is then possible to identify the right hand side $\mathbf{f}(\mathbf{x}; \mu)$ as a sparse combination of functions of components in $\mathbf{x}$ as well as the bifurcation parameter μ. This idea is illustrated on two examples, the one-dimensional logistic map and the two-dimensional Hopf normal form.

4.4.1 Logistic map

The logistic map is a classical model that exhibits a cascade of bifurcations, leading to chaotic trajectories. The dynamics with stochastic forcing η_k and parameter r are given by

$$x_{k+1} = r x_k (1 - x_k) + \eta_k. \tag{26}$$

Sampling the stochastic system at ten parameter values of r, the algorithm correctly identifies the underlying parameterized dynamics, shown in Fig. 14 and Table 12.

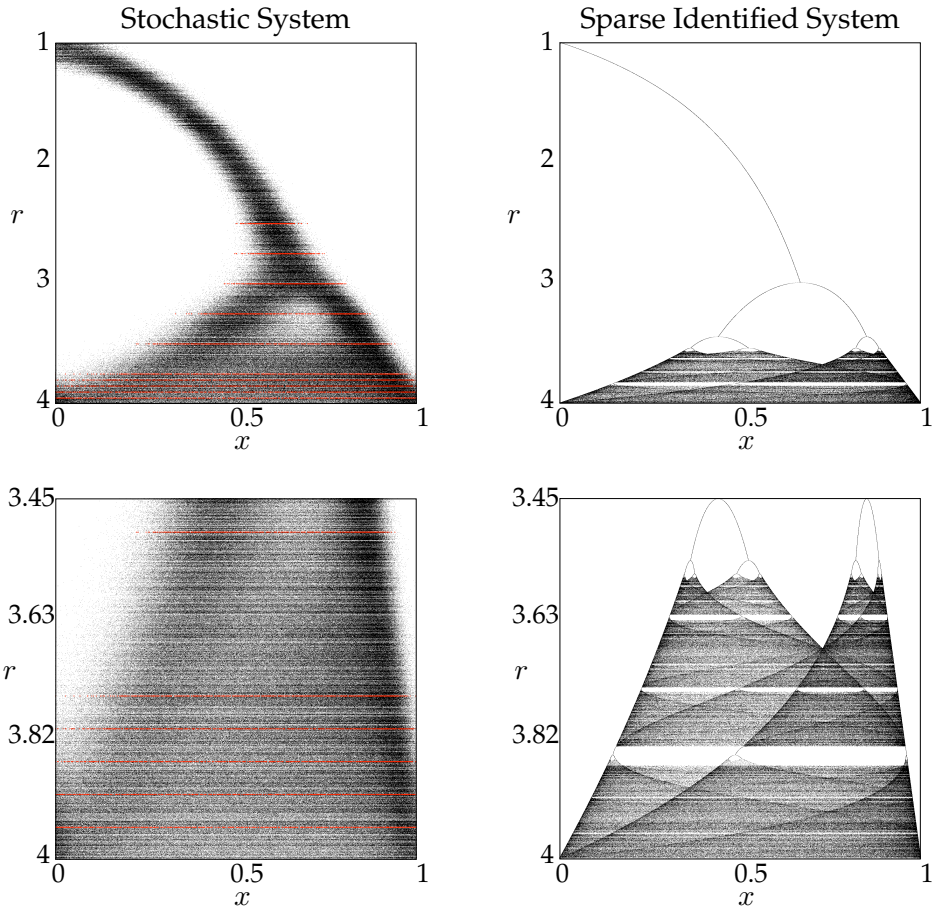

Figure 14: Attracting sets of the logistic map vs. the parameter r. (left) Data from stochastically forced system and (right) the sparse identified system. Data is sampled at rows indicated in red for $r \in \{2.5, 2.75, 3, 3.25, 3.5, 3.75, 3.8, 3.85, 3.9, 3.95\}$. The forcing η_k is Gaussian with magnitude 0.025.

4.4.2 Hopf normal form

The final example illustrating the ability of the sparse dynamics method to identify parameterized normal forms is the Hopf normal form [44]. Noisy data is collected from the Hopf system

$$\dot{x} = \mu x + \omega y - Ax(x^2 + y^2) \tag{27a}$$
$$\dot{y} = -\omega x + \mu y - Ay(x^2 + y^2) \tag{27b}$$

for various values of the parameter μ. Data is collected on the blue and red trajectories in Fig. 15, and noise is added to simulate sensor noise. The total variation derivative [32] is used to de-noise the derivative for use in the algorithm.

The sparse model identification algorithm correctly identifies the Hopf normal form, with model parameters given in Table 13. The noise-free model reconstruction is shown in Fig. 16. Note that with noise in the training data, although the model terms are correctly identified, the actual values of the cubic terms are off by almost 8%. Collecting more training data or reducing the noise magnitude both improve the model agreement.

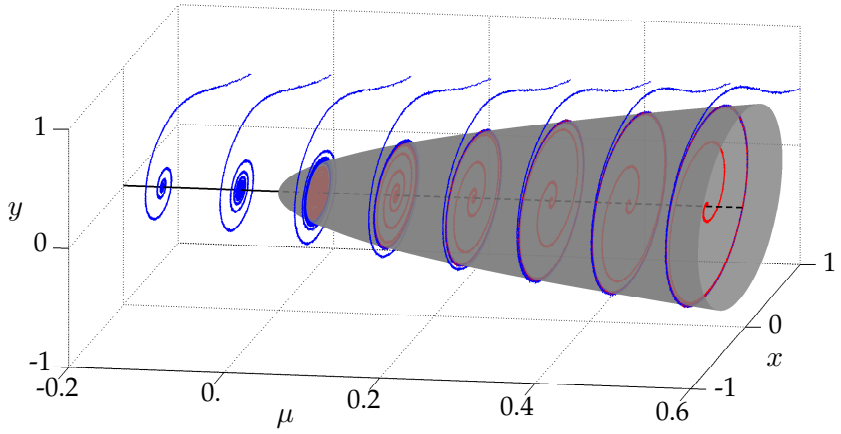

Figure 15: Training data to identify Hopf normal form. Blue trajectories denote solutions that start outside of the fixed point for $\mu < 0$ or the limit cycle for $\mu > 0$, and red trajectories denote solutions that start inside of the limit cycle.

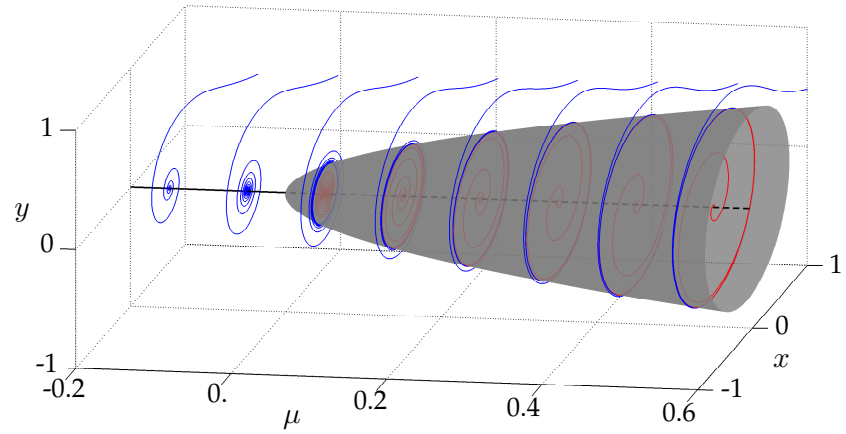

Figure 16: Sparse model captures the Hopf normal form. Initial conditions are the same as in Fig. 15

4.5 Sparse identification of the Lorenz system with time-delay coordinates

It is not always clear what measurements of a dynamical system to take, and even if we did know, these measurements may be prohibitively expensive to collect. Here, we explore the ability to extract dynamics in the Lorenz system if only the first variable $x(t)$ is measured. It is well-known that time-delay coordinates allow us to synthesize additional dynamic variables using a time-series measurement from a single variable $x(t)$ [11]. The dynamics in these time-delay coordinates produce a new attractor with the same topology, according to Takens' theorem [45] . In particular, we construct a Hankel matrix by stacking delayed time-series of x as rows:

$$\mathbf{H} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_p \\ x_2 & x_3 & x_4 & \cdots & x_{p+1} \\ x_3 & x_4 & x_5 & \cdots & x_{p+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_q & x_{q+1} & x_{q+2} & \cdots & x_{p+q-1} \end{bmatrix} \tag{28}$$

Taking the singular value decomposition (SVD), we obtain

$$\mathbf{H} = \mathbf{\Psi \Sigma V}^*, \tag{29}$$

where we may think of columns of $\mathbf{V}$ as a hierarchical set of *eigen*-time-series. For this example, we collect measurements from $t = 0$ to $t = 100$ with $\Delta t = 0.001$, and we stack $q = 10$ rows in $\mathbf{H}$; it is possible to stack more rows in $\mathbf{H}$, although this is not relevant for this discussion.

We choose the first three dominant eigen-time-series given by the first three columns of $\mathbf{V}$, and we denote these coordinates as u, v, and w for convenience. The new time-delay embedding is shown, for short time up to $t = 5$, in Fig. 17.

Using these time-delay coordinates, it is possible to compute the derivatives $\dot{u}$, $\dot{v}$, and $\dot{w}$ numerically using a fourth-order central difference; in cases with noise, we recommend the total-variation regularized derivative. Next, we use our time-delay coordinates and derivatives as inputs to the SINDy algorithm, and the resulting model coefficients identified up to cubic order are shown in Table 1. These coefficients have been identified after normalizing the columns of $\mathbf{\Theta}(\mathbf{V})$. We use a third order polynomial basis since increasing the polynomial order results in over-fitting for this case. Since we determine time-delay coordinates using the SVD, there is a small amount of information missing from the three coordinates chosen that are captured in lower energy columns

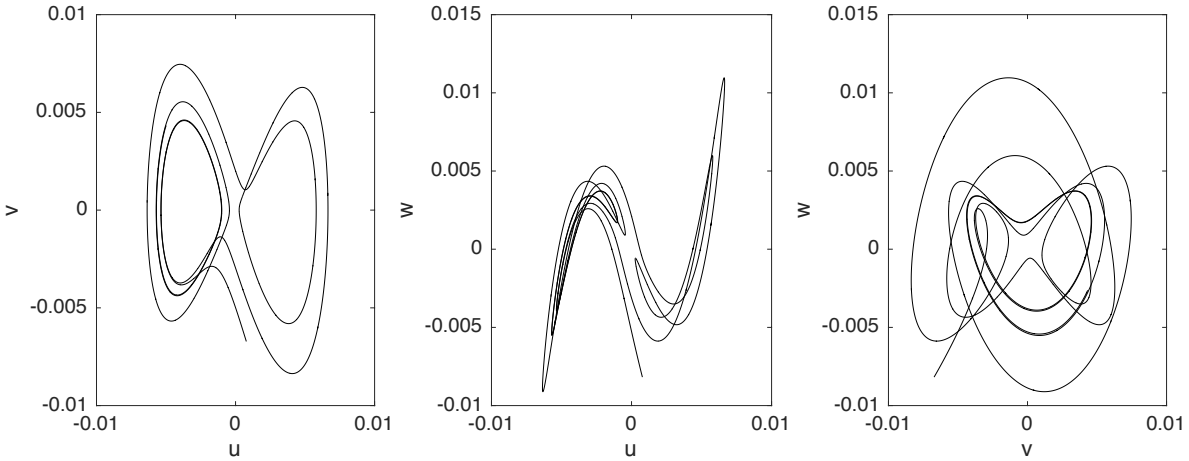

Figure 17: Lorenz attractor in time-delay coordinates for short time.

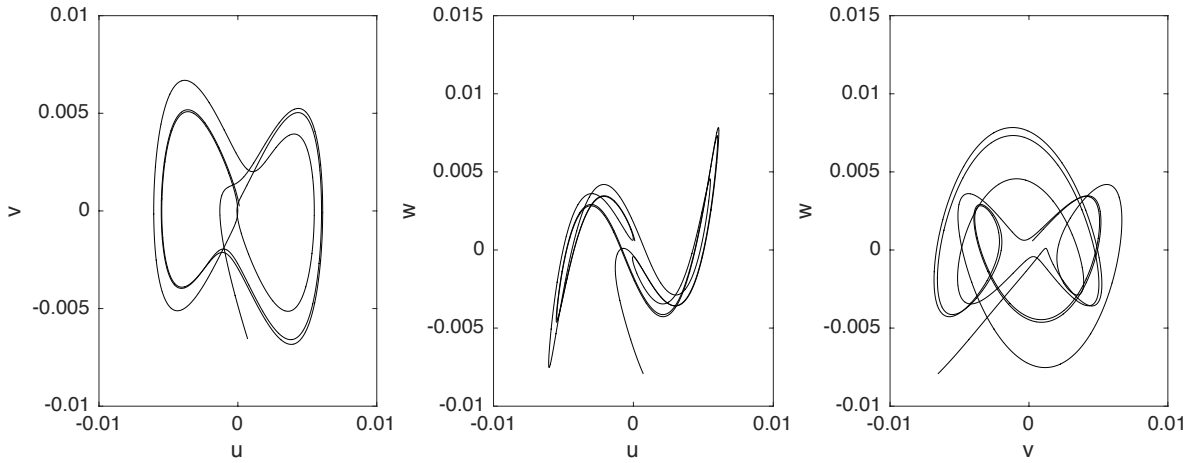

Figure 18: SINDy reconstruction of Lorenz attractor in time-delay coordinates.

of $\mathbf{V}$. Thus, when selecting higher order polynomials, we may encounter overfit models, when in reality there is a small amount of missing information in the lower energy coordinates. It would be interesting to investigate the interplay between the number and energy of the time-delay coordinates, the polynomial order, and the attainable model fidelity.

The reconstruction using the sparse identified dynamics from Table 1 is shown in Fig. 18. For short times, the identified dynamics are qualitatively quite similar to the true time-delay embedding, capturing the *skeleton* of the attractor. For longer times, the identified dynamics do not have the same measure on the attractor as the time-delay embedding, but instead attract onto a complicated quasi-periodic orbit on the attractor. We believe that this behavior in the identified system is the result of small numerical errors introduced by truncating the SVD and the nonlinear terms. However, the short-time behavior is extremely promising, indicating that it is possible to capture very similar dynamics even without collecting the right measurements up-front. Finally, Fig. 19 shows the accuracy of the identified dynamics in satisfying the equations $\dot{\mathbf{V}} = \mathbf{\Theta}(\mathbf{V})\mathbf{\Xi}$, and Fig. 20 shows the correlation between measured and approximated derivatives for various λ.

Table 1: SINDy coefficients for Lorenz system in time-delay coordinates.

	'udot'	'vdot'	'wdot'
'1'	[0]	[0]	[0]
'u'	[0]	[-5.1787]	[-12.0796]
'v'	[5.2307]	[0]	[-7.2660]
'w'	[0]	[-10.3782]	[-5.1039]
'uu'	[0]	[0]	[0]
'uv'	[0]	[0]	[0]
'uw'	[0]	[0]	[0]
'vv'	[0]	[0]	[0]
'vw'	[0]	[0]	[0]
'ww'	[0]	[0]	[0]
'uuu'	[0]	[0]	[17.3748]
'uuv'	[0]	[0]	[20.0657]
'uuw'	[0]	[0]	[0]
'uvv'	[0]	[0]	[-4.9360]
'uvw'	[0]	[0]	[0]
'uww'	[0]	[0]	[0]
'vvv'	[0]	[0]	[0]
'vvw'	[0]	[0]	[0]
'vww'	[0]	[0]	[0]
'www'	[0]	[0]	[0]

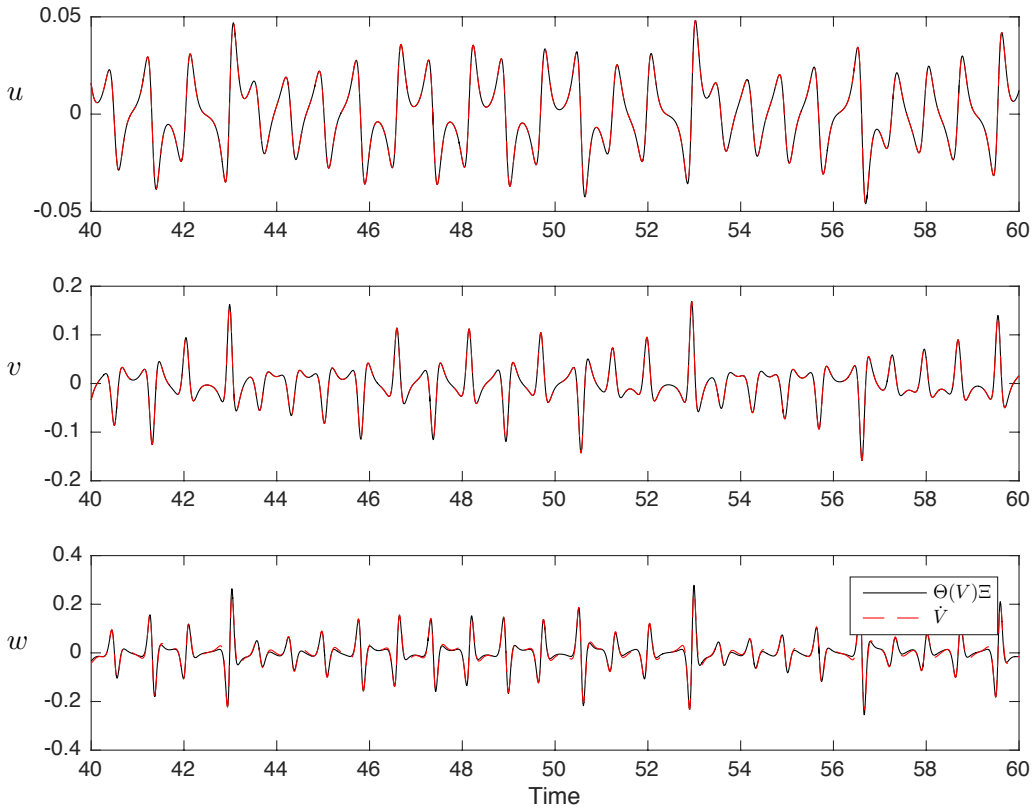

Figure 19: Accuracy of sparse dynamics coefficients in capturing $\dot{\mathbf{V}} = \boldsymbol{\Theta}(\mathbf{V})\boldsymbol{\Xi}$.

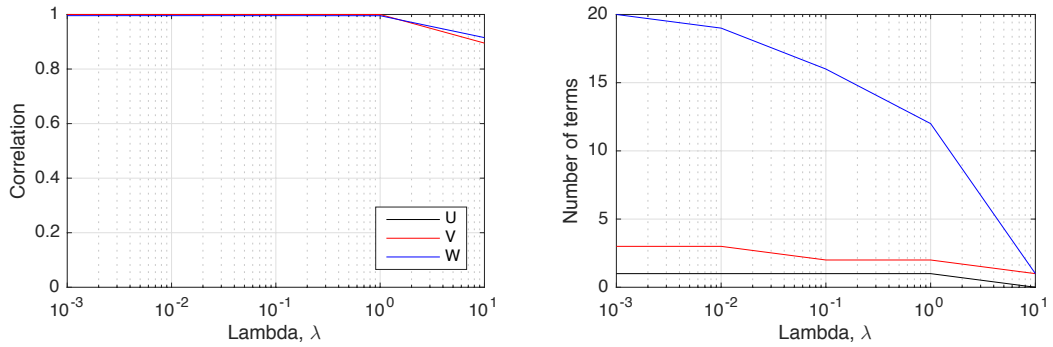

Figure 20: (left) Correlation of computed derivatives $\boldsymbol{\Theta}(\mathbf{V})\boldsymbol{\Xi}$ with measured derivatives $\dot{\mathbf{V}}$ and (right) number of terms in the differential equations as a function of the sparsifying parameter λ.

5 Discussion

In this work, we have demonstrated a powerful new technique to identify nonlinear dynamical systems from data without assumptions on the form of the nonlinearity. This builds on prior work in symbolic regression but with innovations related to sparse representation, which allow our algorithms to scale to high-dimensional complex systems. The new method is demonstrated on a number of example systems exhibiting chaos, big data with low-rank coherence, and parameterized dynamics. The identification of *sparse* nonlinearities and parameterizations mark a significant step toward the long-held goal of intelligent, unassisted identification of dynamical systems.

There are numerous fields where this method may be applied, where there is ample data and

the absence of governing equations. These applications include neuroscience, climate science, epidemiology, and financial markets. As shown in the Lorenz example, the ability to predict a specific trajectory, may be less important than the ability to capture the attractor dynamics. The method also generalizes to partial differential equations, as demonstrated on an example from fluid mechanics. Finally, normal forms may be discovered by including parameters in the optimization, as shown on two examples.

In each of the examples shown, we have investigated the robustness of the sparse dynamics algorithm to measurement noise and the unavailability of derivative measurements. In each case, the sparse regression framework appears well-suited to measurement and process noise, especially when derivatives are smoothed using the total-variation regularized derivative. We also find that larger noise magnitude increases the data required for accurate model identification.

There are significant implications of this method for fields that are already using symbolic regression or genetic programming. The inclusion of genetic programming and symbolic regression in a convex framework may allow these methods to generalize to much larger systems.

A number of open problems remain surrounding the dynamical systems aspects of this procedure. For example, many systems possess dynamical symmetries and conserved quantities that may alter the form of the identified dynamics. For example, the degenerate identification of a linear system in a space of high-order polynomial nonlinearities suggest a connection with near-identity transformations and dynamic similarity. We believe that this may be a fruitful line of research. A significant outstanding issue in the above approach is the correct choice of measurement coordinates and the choice of sparsifying function basis for the dynamics. There is no simple solution to this challenge, and there must be a coordinated effort to incorporate expert knowledge, feature extraction, and inference based methods to tackle this in general. However, in practice, there may be some hope of obtaining the correct coordinate system and function basis without knowing the solution ahead of time, since we often know something about the physics that guide the choice of function space. In the case that we have few measurements, these may be augmented using time delay coordinates, and when we have too many measurements, we may extract coherent structures using advanced methods from dimensionality reduction and machine learning. It may also be possible to make the dynamics more sparse through subsequent coordinate transformations [2]. We hope that this connection between sparsity methods, machine learning, and dynamical systems will spur developments to automate and improve these choices.

Appendix A: Choice of basis functions

In real-world systems, the correct choice of basis functions to sparsely represent the dynamics might not be clear, although physical intuition may be leveraged in many systems. Here, we explore a simple ODE where the right hand side dynamics are given by a trigonometric function:

$$\frac{d}{dt}x = -\sin(x) \tag{30a}$$

$$= -x + \frac{1}{3!}x^3 - \frac{1}{5!}x^5 + \mathcal{O}(x^7). \tag{30b}$$

On this test problem, we investigate the SINDy algorithm with different bases, including polynomial basis functions, trigonometric functions, and a combination of polynomial and trigonometric functions. In the case of a polynomial basis, shown in Table 2, the SINDy algorithm identifies the correct terms in the Taylor expansion of $f(x) = -\sin(x)$. The dynamic reconstruction, shown in Fig. 21 is excellent. In both of the other cases of a purely trigonometric basis or a basis consisting of polynomials and trigonometric functions, the correct term $f(x) = -\sin(x)$ is identified.

Table 2: Polynomial basis.

	'xdot'
'1'	0
'x'	-1.0000
'xx'	0
'xxx'	0.1664
'xxxx'	0
'xxxxx'	-0.0079

Table 3: Trigonometric basis.

	'xdot'
'sin(x)'	-1.0000
'cos(x)'	0
'sin(2x)'	0
'cos(2x)'	0

Table 4: Polynomial and trigonometric basis.

	'xdot'
'1'	0
'x'	0
'xx'	0
'xxx'	0
'xxxx'	0
'xxxxx'	0
'sin(x)'	-1.0000
'cos(x)'	0
'sin(2x)'	0
'cos(2x)'	0

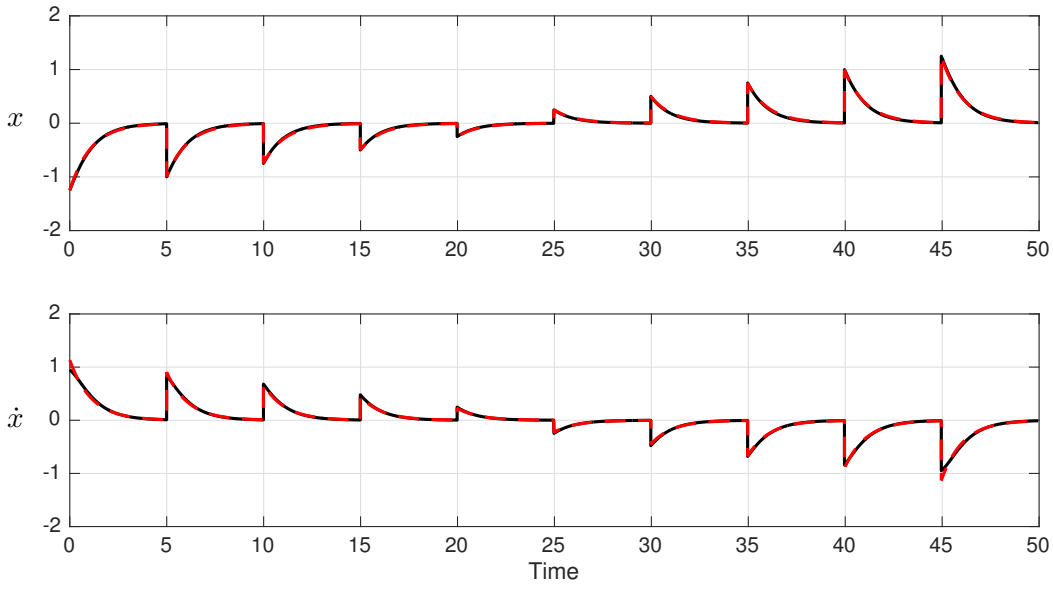

Figure 21: Data (black) and sparse dynamics reconstruction (red) for a sequence of initial conditions initialized every 5 time units. Initial conditions are chosen from -1.25 to 1.25 in increments of 0.25 (excluding $x_0 = 0$).

Appendix B: Limitations of the sparse identification framework

The sparse identification algorithm above relies on having measurements in a sensible coordinate system where the dynamics are sparse in the chosen function basis. Here, we explore the limitations of this modeling framework when the coordinates or function basis are not amenable to sparse representation of the dynamics.

B-1: Lorenz system in nonlinear coordinates

The limitations of the method are clear when we transform the simple Lorenz example from the natural coordinates (x, y, z) nonlinearly into the new coordinates (A, B, C) according to the map:

$$A = x\sin(x) \tag{31a}$$
$$B = y\cos(y) \tag{31b}$$
$$C = z\sin(z). \tag{31c}$$

In these new coordinates the Lorenz system has complicated nonlinear behavior that is not well approximated by a dynamical system with polynomial nonlinearities. The system response in (A, B, C) coordinates is shown in Fig. 22.

The sparse identification algorithm fails to identify a model that agrees with the measured derivatives for any value of the sparsity promoting parameter λ, as shown in the correlation plot in Fig. 23. For this problem, various search spaces were explored, including polynomial nonlinearities up to fifth order as well as trigonometric functions.

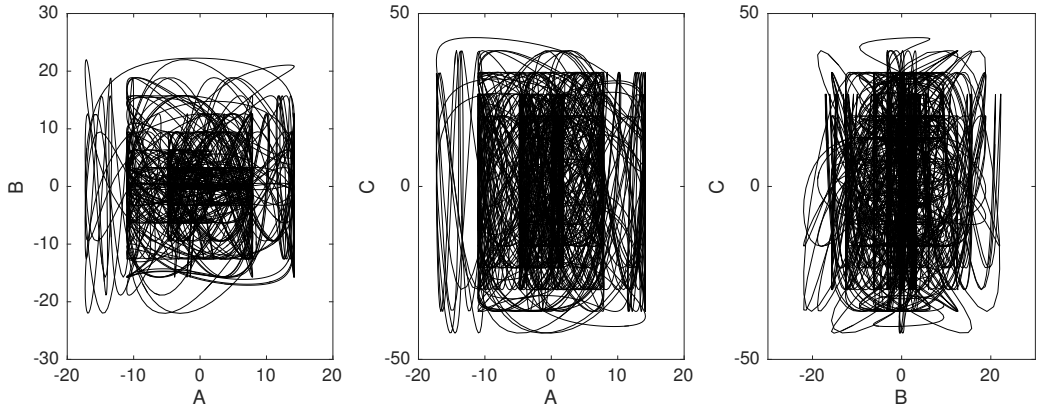

Figure 22: Lorenz attractor in nonlinear coordinates (A, B, C) from Eq. (31).

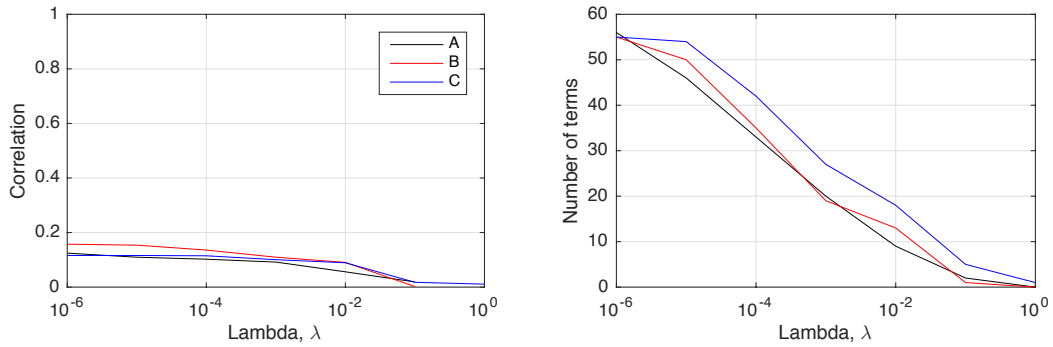

Figure 23: Correlation of sparse prediction of derivatives $\boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi}$ and measured derivatives $\dot{\mathbf{X}}$ of Lorenz system in nonlinear coordinates (A, B, C) from Eq. (31).

As in Sec. 4.5, the use of generalized eigen-time-delay coordinates present a promising technique to find a natural coordinate system. Figure 24 shows the first three time-delay coordinates obtained from the eigen-decomposition of the Hankel matrix in Eq. (28); for this example, the number of rows is $q = 100$ and the Lorenz system is simulated from $t = 0$ to $t = 100$ with $\Delta t = 0.001$. Using these time-delay coordinates results in much better correlation between measured and modeled derivatives, even for relatively sparse models, as shown in Fig. 25. Increasing the number of time-delay coordinates to the leading four results in improved correlation, shown in Fig. 26. However, the resulting dynamical systems for each of these cases does not accurately reproduce the attractor dynamics, motivating further research to identify natural coordinate systems to measure in and natural function bases to represent dynamics sparsely.

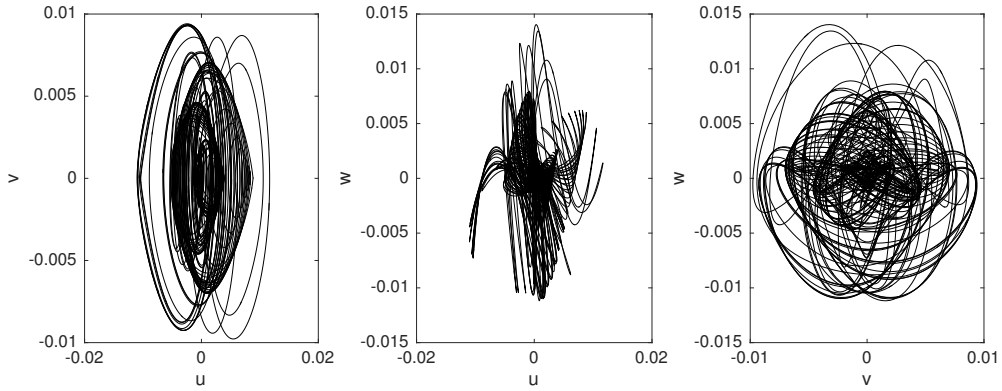

Figure 24: Lorenz attractor in time-delay coordinates obtained from measurements (A, B, C) from Eq. (31).

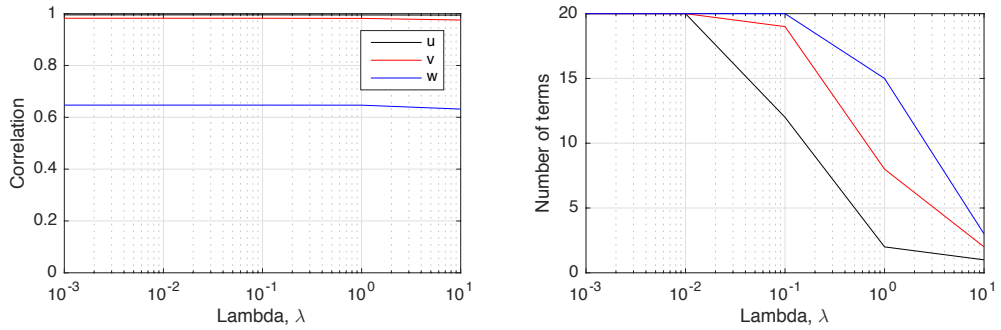

Figure 25: Correlation of sparse model prediction $\mathbf{\Theta}(\mathbf{X})\mathbf{\Xi}$ and measured derivatives $\dot{\mathbf{X}}$ of Lorenz system using the first three time-delay coordinates obtained from measurements (A, B, C) from Eq. (31).

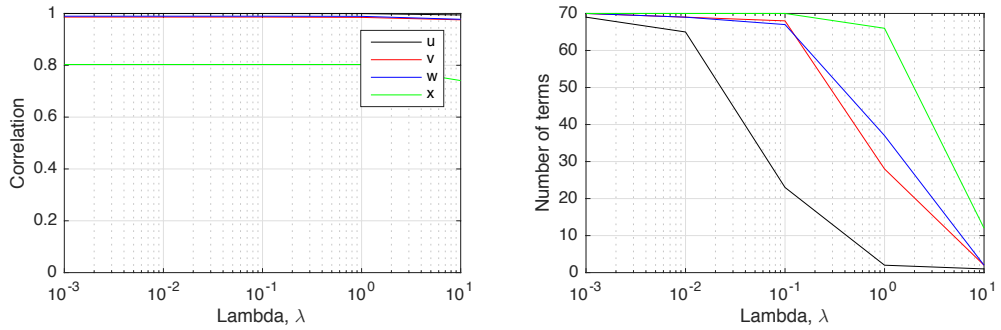

Figure 26: Correlation of sparse model prediction $\mathbf{\Theta}(\mathbf{X})\mathbf{\Xi}$ and measured derivatives $\dot{\mathbf{X}}$ of Lorenz system using the first four time-delay coordinates obtained from measurements (A, B, C) from Eq. (31).

B-2: Glycolytic oscillator model

The glycolytic oscillator model is a standard benchmark problem for model prediction, system identification and automatic inference [6, 8, 7]. We simulate the system presented in Daniels and Nemenman [8] (Eq. (19) in [8]) :

$$\frac{dS_1}{dt} = J_0 - \frac{k_1 S_1 S_6}{1 + (S_6/K_1)^q}, \tag{32a}$$

$$\frac{dS_2}{dt} = 2\frac{k_1 S_1 S_6}{1 + (S_6/K_1)^q} - k_2 S_2 (N - S_5) - k_6 S_2 S_5, \tag{32b}$$

$$\frac{dS_3}{dt} = k_2 S_2 (N - S_5) - k_3 S_3 (A - S_6), \tag{32c}$$

$$\frac{dS_4}{dt} = k_3 S_3 (A - S_6) - k_4 S_4 S_5 - \kappa (S_4 - S_7), \tag{32d}$$

$$\frac{dS_5}{dt} = k_2 S_2 (N - S_5) - k_4 S_4 S_5 - k_6 S_2 S_5, \tag{32e}$$

$$\frac{dS_6}{dt} = -2\frac{k_1 S_1 S_6}{1 + (S_6/K_1)^q} + 2k_3 S_3 (A - S_6) - k_5 S_6, \tag{32f}$$

$$\frac{dS_7}{dt} = \psi \kappa (S_4 - S_7) - k S_7. \tag{32g}$$

Daniels and Nemenman [8] provide the various parameters (Table 1 in [8]) and initial conditions (Table 2 in [8]) to match yeast glycolysis. Data from a simulation of Eq. (32) using these parameters and initial conditions and a time step of $\Delta t = 0.001$ minutes is shown in Fig. 27.

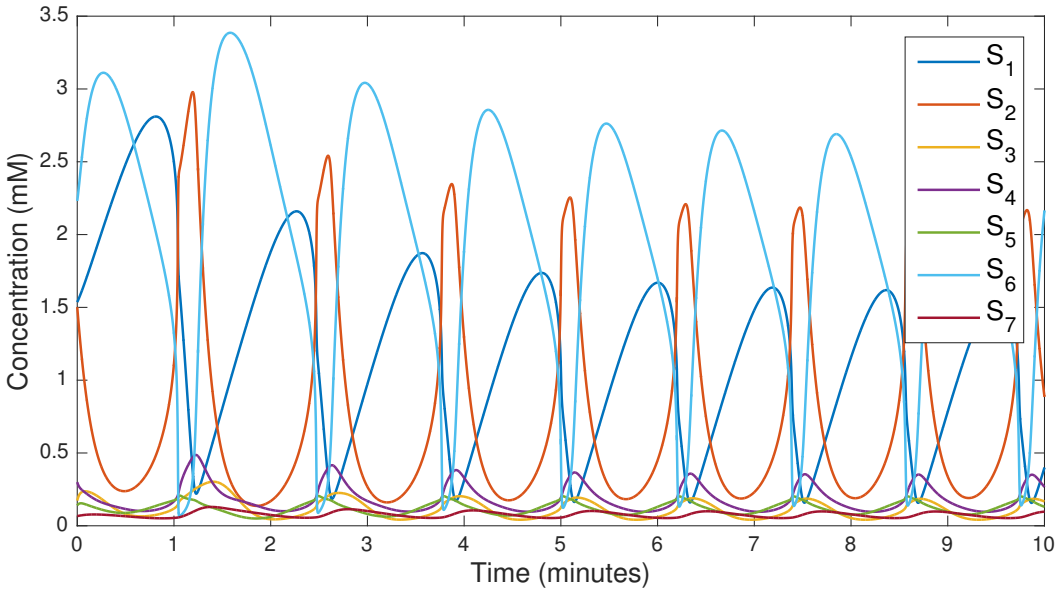

Figure 27: Glycolytic oscillator network dynamics for random initial conditions chosen from the ranges provided in Table 2 of Daniels and Nemenman [8].

The results of the sparse identification of nonlinear dynamics algorithm are shown in Tab. 5. The algorithm accurately identifies the dynamics for S_3, S_4, S_5, and S_7, since each of these variables have dynamics that are sparse in the polynomial search basis. However, the algorithm does not identify sparse dynamics for the S_1, S_2, and S_6 terms, which each have a rational function in their dynamics. Although the identified model in Tab. 5 produces derivatives that accurately match the measured derivatives, as seen in the correlation plot in Fig. 28, the dynamic model does not agree with the true system, except for a very short time at the beginning of the simulation.

The fact that the algorithm produces accurate sparse dynamics in some of the variables (S_3, S_4, S_5, and S_7) is a good indication that the measurement coordinates are correct. The fact that the dynamics are not sparse in the remaining equations indicates that the function basis is not appropriate for sparse representation of the dynamics for the remaining equations (S_1, S_2, and S_6). Investigating how to generalize the SINDy algorithm to include a broader function search space is an important area of current and future work.

Table 5: Identified dynamics for metabolic network.

''	'S1dot'	'S2dot'	'S3dot'	'S4dot'	'S5dot'	'S6dot'	'S7dot'
'1'	-780.181	1.565e+03	0	0	0	-1.565e+03	0
'S1'	-82.317	164.633	0	0	0	-164.633	0
'S2'	-70.328	134.657	6.00	0	6.00	-140.657	0
'S3'	-1.578e+04	3.156e+04	-64.00	64.00	0	-3.143e+04	0
'S4'	-1.044e+03	2.087e+03	0	-13.00	0	-2.087e+03	1.3000
'S5'	1.309e+04	-2.618e+04	0	0	0	2.618e+04	0
'S6'	-439.231	878.461	0	0	0	-879.741	0
'S7'	3.982e+04	-7.964e+04	0	13.00	0	7.964e+04	-3.1000
'S1S1'	18.308	-36.615	0	0	0	36.615	0
'S1S2'	253.763	-507.527	0	0	0	507.526	0
'S1S3'	3.706e+03	-7.412e+03	0	0	0	7.412e+03	0
'S1S4'	-607.006	1.214e+03	0	0	0	-1.214e+03	0
'S1S5'	-1.752e+03	3.505e+03	0	0	0	-3.505e+03	0
'S1S6'	311.284	-622.568	0	0	0	622.568	0
'S1S7'	-9.857e+03	1.972e+04	0	0	0	-1.972e+04	0
'S2S2'	231.996	-463.993	0	0	0	463.993	0
'S2S3'	1.107e+04	-2.213e+04	0	0	0	2.213e+04	0
'S2S4'	-242.407	484.813	0	0	0	-484.813	0
'S2S5'	-8.786e+03	1.757e+04	-6.00	0	-18.00	-1.757e+04	0
'S2S6'	434.818	-869.636	0	0	0	869.636	0
'S2S7'	-2.041e+04	4.082e+04	0	0	0	-4.082e+04	0
'S3S3'	-1.086e+03	2.171e+03	0	0	0	-2.171e+03	0
'S3S4'	-3.535e+04	7.071e+04	0	0	0	-7.071e+04	0
'S3S5'	2.056e+04	-4.112e+04	0	0	0	4.112e+04	0
'S3S6'	5.193e+03	-1.039e+04	16.00	-16.00	0	1.035e+04	0
'S3S7'	4.116e+03	-8.232e+03	0	0	0	8.232e+03	0
'S4S4'	7.587e+03	-1.517e+04	0	0	0	1.517e+04	0
'S4S5'	1.342e+04	-2.684e+04	0	-100.00	-100.00	2.684e+04	0
'S4S6'	1.267e+03	-2.533e+03	0	0	0	2.533e+03	0
'S4S7'	-1.320e+04	2.640e+04	0	0	0	-2.640e+04	0
'S5S5'	-7.955e+03	1.591e+04	0	0	0	-1.591e+04	0
'S5S6'	-5.507e+03	1.101e+04	0	0	0	-1.101e+04	0
'S5S7'	2.174e+04	-4.348e+04	0	0	0	4.348e+04	0
'S6S6'	214.779	-429.558	0	0	0	429.558	0
'S6S7'	-1.561e+04	3.122e+04	0	0	0	-3.122e+04	0
'S7S7'	8.259e+04	-1.652e+05	0	0	0	1.652e+05	0

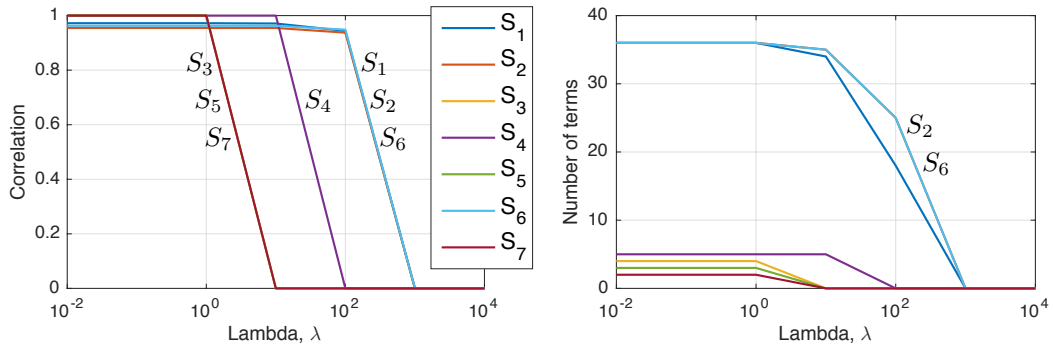

Figure 28: Correlation of sparse model prediction $\boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi}$ and measured derivatives $\dot{\mathbf{X}}$ for glycolytic oscillator model.

Appendix C: Identified coefficients of dynamics

Table 6: Damped harmonic oscillator with linear terms.

	'xdot'	'ydot'
'1'	[0]	[0]
'x'	[-0.1015]	[-1.9990]
'y'	[2.0027]	[-0.0994]
'xx'	[0]	[0]
'xy'	[0]	[0]
'yy'	[0]	[0]
'xxx'	[0]	[0]
'xxy'	[0]	[0]
'xyy'	[0]	[0]
'yyy'	[0]	[0]
'xxxx'	[0]	[0]
'xxxy'	[0]	[0]
'xxyy'	[0]	[0]
'xyyy'	[0]	[0]
'yyyy'	[0]	[0]
'xxxxx'	[0]	[0]
'xxxxy'	[0]	[0]
'xxxyy'	[0]	[0]
'xxyyy'	[0]	[0]
'xyyyy'	[0]	[0]
'yyyyy'	[0]	[0]

Table 7: Damped harmonic oscillator with cubic nonlinearity.

''	'xdot'	'ydot'
'1'	[0]	[0]
'x'	[0]	[0]
'y'	[0]	[0]
'xx'	[0]	[0]
'xy'	[0]	[0]
'yy'	[0]	[0]
'xxx'	[-0.0996]	[-1.9994]
'xxy'	[0]	[0]
'xyy'	[0]	[0]
'yyy'	[1.9970]	[-0.0979]
'xxxx'	[0]	[0]
'xxxy'	[0]	[0]
'xxyy'	[0]	[0]
'xyyy'	[0]	[0]
'yyyy'	[0]	[0]
'xxxxx'	[0]	[0]
'xxxxy'	[0]	[0]
'xxxyy'	[0]	[0]
'xxyyy'	[0]	[0]
'xyyyy'	[0]	[0]
'yyyyy'	[0]	[0]

Table 8: Three-dimensional linear system.

''	'xdot'	'ydot'	'zdot'
'1'	[0]	[0]	[0]
'x'	[-0.0996]	[-1.9997]	[0]
'y'	[2.0005]	[-0.0994]	[0]
'z'	[0]	[0]	[-0.3003]
'xx'	[0]	[0]	[0]
'xy'	[0]	[0]	[0]
'xz'	[0]	[0]	[0]
'yy'	[0]	[0]	[0]
'yz'	[0]	[0]	[0]
'zz'	[0]	[0]	[0]

Table 9: Lorenz system identified using SINDy, assuming measurements of $\mathbf{x}$ and $\dot{\mathbf{x}}$, with $\eta = 1.0$.

	'xdot'	'ydot'	'zdot'
'1'	[0]	[0]	[0]
'x'	[-9.9996]	[27.9980]	[0]
'y'	[9.9998]	[-0.9997]	[0]
'z'	[0]	[0]	[-2.6665]
'xx'	[0]	[0]	[0]
'xy'	[0]	[0]	[1.0000]
'xz'	[0]	[-0.9999]	[0]
'yy'	[0]	[0]	[0]
'yz'	[0]	[0]	[0]
'zz'	[0]	[0]	[0]
'xxx'	[0]	[0]	[0]
'xxy'	[0]	[0]	[0]
'xxz'	[0]	[0]	[0]
'xyy'	[0]	[0]	[0]
'xyz'	[0]	[0]	[0]
'xzz'	[0]	[0]	[0]
'yyy'	[0]	[0]	[0]
'yyz'	[0]	[0]	[0]
'yzz'	[0]	[0]	[0]
'zzz'	[0]	[0]	[0]
'xxxx'	[0]	[0]	[0]
'xxxy'	[0]	[0]	[0]
'xxxz'	[0]	[0]	[0]
'xxyy'	[0]	[0]	[0]
'xxyz'	[0]	[0]	[0]
'xxzz'	[0]	[0]	[0]
'xyyy'	[0]	[0]	[0]
'xyyz'	[0]	[0]	[0]
'xyzz'	[0]	[0]	[0]
'xzzz'	[0]	[0]	[0]
'yyyy'	[0]	[0]	[0]
'yyyz'	[0]	[0]	[0]
'yyzz'	[0]	[0]	[0]
'yzzz'	[0]	[0]	[0]
'zzzz'	[0]	[0]	[0]
'xxxxx'	[0]	[0]	[0]
'xxxxy'	[0]	[0]	[0]
'xxxxz'	[0]	[0]	[0]
'xxxyy'	[0]	[0]	[0]
'xxxyz'	[0]	[0]	[0]
'xxxzz'	[0]	[0]	[0]
'xxyyy'	[0]	[0]	[0]
'xxyyz'	[0]	[0]	[0]
'xxyzz'	[0]	[0]	[0]
'xxzzz'	[0]	[0]	[0]
'xyyyy'	[0]	[0]	[0]
'xyyyz'	[0]	[0]	[0]
'xyyzz'	[0]	[0]	[0]
'xyzzz'	[0]	[0]	[0]
'xzzzz'	[0]	[0]	[0]
'yyyyy'	[0]	[0]	[0]
'yyyyz'	[0]	[0]	[0]
'yyyzz'	[0]	[0]	[0]
'yyzzz'	[0]	[0]	[0]
'yzzzz'	[0]	[0]	[0]
'zzzzz'	[0]	[0]	[0]

Table 10: Identified dynamics of cylinder wake modes. Notice that quadratic terms are identified.

''	'xdot'	'ydot'	'zdot'
'1'	[-0.1225]	[-0.0569]	[-20.8461]
'x'	[-0.0092]	[1.0347]	[-4.6476e-04]
'y'	[-1.0224]	[0.0047]	[2.4057e-04]
'z'	[-9.2203e-04]	[-4.4932e-04]	[-0.2968]
'xx'	[0]	[0]	[0.0011]
'xy'	[0]	[0]	[0]
'xz'	[2.1261e-04]	[0.0022]	[0]
'yy'	[0]	[0]	[8.6432e-04]
'yz'	[-0.0019]	[-0.0018]	[0]
'zz'	[0]	[0]	[-0.0010]
'xxx'	[0]	[0]	[0]
'xxy'	[0]	[0]	[0]
'xxz'	[0]	[0]	[0]
'xyy'	[0]	[0]	[0]
'xyz'	[0]	[0]	[0]
'xzz'	[0]	[0]	[0]
'yyy'	[0]	[0]	[0]
'yyz'	[0]	[0]	[0]
'yzz'	[0]	[0]	[0]
'zzz'	[0]	[0]	[0]
'xxxx'	[0]	[0]	[0]
'xxxy'	[0]	[0]	[0]
'xxxz'	[0]	[0]	[0]
'xxyy'	[0]	[0]	[0]
'xxyz'	[0]	[0]	[0]
'xxzz'	[0]	[0]	[0]
'xyyy'	[0]	[0]	[0]
'xyyz'	[0]	[0]	[0]
'xyzz'	[0]	[0]	[0]
'xzzz'	[0]	[0]	[0]
'yyyy'	[0]	[0]	[0]
'yyyz'	[0]	[0]	[0]
'yyzz'	[0]	[0]	[0]
'yzzz'	[0]	[0]	[0]
'zzzz'	[0]	[0]	[0]
'xxxxx'	[0]	[0]	[0]
'xxxxy'	[0]	[0]	[0]
'xxxxz'	[0]	[0]	[0]
'xxxyy'	[0]	[0]	[0]
'xxxyz'	[0]	[0]	[0]
'xxxzz'	[0]	[0]	[0]
'xxyyy'	[0]	[0]	[0]
'xxyyz'	[0]	[0]	[0]
'xxyzz'	[0]	[0]	[0]
'xxzzz'	[0]	[0]	[0]
'xyyyy'	[0]	[0]	[0]
'xyyyz'	[0]	[0]	[0]
'xyyzz'	[0]	[0]	[0]
'xyzzz'	[0]	[0]	[0]
'xzzzz'	[0]	[0]	[0]
'yyyyy'	[0]	[0]	[0]
'yyyyz'	[0]	[0]	[0]
'yyyzz'	[0]	[0]	[0]
'yyzzz'	[0]	[0]	[0]
'yzzzz'	[0]	[0]	[0]
'zzzzz'	[0]	[0]	[0]

Table 11: Identified dynamics of cylinder wake modes with smaller λ, resulting in cubic nonlinearities.

''	'xdot'	'ydot'	'zdot'
'1'	0	0	0
'x'	0	0	0
'y'	-1.0420	0.0062	2.5451e-04
'z'	1.9812e-05	-3.5585e-05	0.4750
'xx'	0	0	6.0153e-05
'xy'	0	0	-1.9444e-04
'xz'	0.0014	-0.0074	0
'yy'	0	0	-5.7268e-05
'yz'	-0.0037	-0.0037	0
'zz'	0	0	0.0053
'xxx'	0	4.5311e-05	0
'xxy'	0	0	0
'xxz'	0	0	-3.0965e-05
'xyy'	0	4.9559e-05	0
'xyz'	0	0	-2.3562e-05
'xzz'	1.0918e-05	-2.1442e-05	0
'yyy'	0	0	0
'yyz'	0	0	-2.4035e-05
'yzz'	-1.5787e-05	-1.6271e-05	0
'zzz'	0	0	1.4677e-05
'xxxx'	0	0	0
'xxxy'	0	0	0
'xxxz'	0	0	0
'xxyy'	0	0	0
'xxyz'	0	0	0
'xxzz'	0	0	0
'xyyy'	0	0	0
'xyyz'	0	0	0
'xyzz'	0	0	0
'xzzz'	0	0	0
'yyyy'	0	0	0
'yyyz'	0	0	0
'yyzz'	0	0	0
'yzzz'	0	0	0
'zzzz'	0	0	0
'xxxxx'	0	0	0
'xxxxy'	0	0	0
'xxxxz'	0	0	0
'xxxyy'	0	0	0
'xxxyz'	0	0	0
'xxxzz'	0	0	0
'xxyyy'	0	0	0
'xxyyz'	0	0	0
'xxyzz'	0	0	0
'xxzzz'	0	0	0
'xyyyy'	0	0	0
'xyyyz'	0	0	0
'xyyzz'	0	0	0
'xyzzz'	0	0	0
'xzzzz'	0	0	0
'yyyyy'	0	0	0
'yyyyz'	0	0	0
'yyyzz'	0	0	0
'yyzzz'	0	0	0
'yzzzz'	0	0	0
'zzzzz'	0	0	0

Table 12: Logistic map identified using SINDy.

	x_{k+1}	r_{k+1}
1	0	0
x	0	0
r	0	1.0000
xx	0	0
xr	0.9993	0
rr	0	0
xxx	0	0
xxr	-0.9989	0
xrr	0	0
rrr	0	0
xxxx	0	0
xxxr	0	0
xxrr	0	0
xrrr	0	0
rrrr	0	0
xxxxx	0	0
xxxxr	0	0
xxxrr	0	0
xxrrr	0	0
xrrrr	0	0
rrrrr	0	0

Table 13: Hopf normal form identified with SINDy. Here u represents the bifurcation parameter μ.

	'xdot'	'ydot'	'udot'
'1'	[0]	[0]	[0]
'x'	[0]	[0.9914]	[0]
'y'	[-0.9920]	[0]	[0]
'u'	[0]	[0]	[0]
'xx'	[0]	[0]	[0]
'xy'	[0]	[0]	[0]
'xu'	[0.9269]	[0]	[0]
'yy'	[0]	[0]	[0]
'yu'	[0]	[0.9294]	[0]
'uu'	[0]	[0]	[0]
'xxx'	[-0.9208]	[0]	[0]
'xxy'	[0]	[-0.9244]	[0]
'xxu'	[0]	[0]	[0]
'xyy'	[-0.9211]	[0]	[0]
'xyu'	[0]	[0]	[0]
'xuu'	[0]	[0]	[0]
'yyy'	[0]	[-0.9252]	[0]
'yyu'	[0]	[0]	[0]
'yuu'	[0]	[0]	[0]
'uuu'	[0]	[0]	[0]
'xxxx'	[0]	[0]	[0]
'xxxy'	[0]	[0]	[0]
'xxxu'	[0]	[0]	[0]
'xxyy'	[0]	[0]	[0]
'xxyu'	[0]	[0]	[0]
'xxuu'	[0]	[0]	[0]
'xyyy'	[0]	[0]	[0]
'xyyu'	[0]	[0]	[0]
'xyuu'	[0]	[0]	[0]
'xuuu'	[0]	[0]	[0]
'yyyy'	[0]	[0]	[0]
'yyyu'	[0]	[0]	[0]
'yyuu'	[0]	[0]	[0]
'yuuu'	[0]	[0]	[0]
'uuuu'	[0]	[0]	[0]
'xxxxx'	[0]	[0]	[0]
'xxxxy'	[0]	[0]	[0]
'xxxxu'	[0]	[0]	[0]
'xxxyy'	[0]	[0]	[0]
'xxxyu'	[0]	[0]	[0]
'xxxuu'	[0]	[0]	[0]
'xxyyy'	[0]	[0]	[0]
'xxyyu'	[0]	[0]	[0]
'xxyuu'	[0]	[0]	[0]
'xxuuu'	[0]	[0]	[0]
'xyyyy'	[0]	[0]	[0]
'xyyyu'	[0]	[0]	[0]
'xyyuu'	[0]	[0]	[0]
'xyuuu'	[0]	[0]	[0]
'xuuuu'	[0]	[0]	[0]
'yyyyy'	[0]	[0]	[0]
'yyyyu'	[0]	[0]	[0]
'yyyuu'	[0]	[0]	[0]
'yyuuu'	[0]	[0]	[0]
'yuuuu'	[0]	[0]	[0]
'uuuuu'	[0]	[0]	[0]

References

[1] Ljung L (1999) *System Identification: Theory for the User* (Prentice Hall).

[2] Holmes P, Guckenheimer J (1983) *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, Applied Mathematical Sciences (Springer-Verlag, Berlin) Vol. 42.

[3] Schmidt M, Lipson H (2009) Distilling free-form natural laws from experimental data. *Science* 324:81–85.

[4] Roberts AJ (2014) *Model emergent dynamics in complex systems* (SIAM).

[5] Crutchfield JP, McNamara BS (1987) Equations of motion from a data series. *Complex systems* 1:417–452.

[6] Schmidt MD, et al. (2011) Automated refinement and inference of analytical models for metabolic networks. *Physical biology* 8:055011.

[7] Daniels BC, Nemenman I (2015) Automated adaptive inference of phenomenological dynamical models. *Nature communications* 6.

[8] Daniels BC, Nemenman I (2015) Efficient inference of parsimonious phenomenological models of cellular dynamics using s-systems and alternating regression. *PloS one* 10:e0119821.

[9] Kevrekidis IG, et al. (2003) Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Science* 1:715–762.

[10] Sugihara G, et al. (2012) Detecting causality in complex ecosystems. *Science* 338:496–500.

[11] Ye H, et al. (2015) Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling. *PNAS* 112:E1569–E1576.

[12] Hastie T, et al. (2009) *The elements of statistical learning* (Springer) Vol. 2.

[13] James G, Witten D, Hastie T, Tibshirani R (2013) *An introduction to statistical learning* (Springer).

[14] Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J. of the Royal Statistical Society B* pp 267–288.

[15] Donoho DL (2006) Compressed sensing. *IEEE Trans. Information Theory* 52:1289–1306.

[16] Candès EJ, Romberg J, Tao T (2006) Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* 52:489–509.

[17] Candès EJ, Romberg J, Tao T (2006) Stable signal recovery from incomplete and inaccurate measurements. *Communications in Pure and Applied Mathematics* 59:1207–1223.

[18] Candès EJ (2006) Compressive sensing. *Proc. International Congress of Mathematics*.

[19] Baraniuk RG (2007) Compressive sensing. *IEEE Signal Processing Magazine* 24:118–120.

[20] Tropp JA, Gilbert AC (2007) Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory* 53:4655–4666.

[21] Rowley CW, Mezić I, Bagheri S, Schlatter P, Henningson D (2009) Spectral analysis of nonlinear flows. *J. Fluid Mech.* 645:115–127.

[22] Schmid PJ (2010) Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics* 656:5–28.

[23] Mezic I (2013) Analysis of fluid flows via spectral properties of the koopman operator. *Annual Review of Fluid Mechanics* 45:357–378.

[24] Wang WX, Yang R, Lai YC, Kovanis V, Grebogi C (2011) Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *PRL* 106:154101.

[25] Schaeffer H, Caflisch R, Hauck CD, Osher S (2013) Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences USA* 110:6634–6639.

[26] Ozoliņš V, Lai R, Caflisch R, Osher S (2013) Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences* 110:18368–18373.

[27] Mackey A, Schaeffer H, Osher S (2014) On the compressive spectral method. *Multiscale Modeling & Simulation* 12:1800–1827.

[28] Brunton SL, Tu JH, Bright I, Kutz JN (2014) Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM Journal on Applied Dynamical Systems* 13:1716–1732.

[29] Proctor JL, Brunton SL, Brunton BW, Kutz JN (2014) Exploiting sparsity and equation-free architectures in complex systems (invited review). *The European Physical Journal Special Topics* 223:2665–2684.

[30] Bai Z, et al. (2014) Low-dimensional approach for reconstruction of airfoil data via compressive sensing. *AIAA Journal* pp 1–14.

[31] Koza JR (1992) *Genetic programming: on the programming of computers by means of natural selection* (MIT press) Vol. 1.

[32] Chartrand R (2011) Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics* 2011.

[33] Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* 60:259–268.

[34] Gavish M, Donoho DL (2014) The optimal hard threshold for singular values is $4/\sqrt{3}$. *ArXiv e-prints*.

[35] Berkooz G, Holmes P, Lumley JL (1993) The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics* 23:539–575.

[36] Holmes PJ, Lumley JL, Berkooz G, Rowley CW (2012) *Turbulence, coherent structures, dynamical systems and symmetry*, Cambridge Monographs in Mechanics (Cambridge University Press, Cambridge, England), 2nd edition.

[37] Lorenz EN (1963) Deterministic nonperiodic flow. *J. Atmos. Sciences* 20:130–141.

[38] Taira K, Colonius T (2007) The immersed boundary method: a projection approach. *Journal of Computational Physics* 225:2118–2137.

[39] Colonius T, Taira K (2008) A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering* 197:2131–2146.

[40] Ruelle D, Takens F (1971) On the nature of turbulence. *Comm. Math. Phys.* 20:167–192.

[41] Jackson CP (1987) A finite-element study of the onset of vortex shedding in flow past variously shaped bodies. *Journal of Fluid Mechanics* 182:23–45.

[42] Zebib Z (1987) Stability of viscous flow past a circular cylinder. *Journal of Engineering Mathematics* 21:155–165.

[43] Noack BR, Afanasiev K, Morzynski M, Tadmor G, Thiele F (2003) A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics* 497:335–363.

[44] Marsden JE, McCracken M (1976) *The Hopf bifurcation and its applications* (Springer-Verlag) Vol. 19.

[45] Takens F (1981) Detecting strange attractors in turbulence. *Lecture Notes in Mathematics* 898:366–381.