

Sparse Identification of Nonlinear Dynamics for Rapid Model Recovery

Markus Quade,^{1,2, a)} Markus Abel,^{1,2} J. Nathan Kutz,³ and Steven L. Brunton⁴

¹⁾ *Universität Potsdam, Institut für Physik und Astronomie, Karl-Liebknecht-Straße 24/25, 14476 Potsdam, Germany*

²⁾ *Ambrosys GmbH, David-Gilly Straße 1, 14469 Potsdam, Germany*

³⁾ *Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA*

⁴⁾ *Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA*

Big data has become a critically enabling component of emerging mathematical methods aimed at the automated discovery of dynamical systems, where first principles modeling may be intractable. However, in many engineering systems, abrupt changes must be rapidly characterized based on limited, incomplete, and noisy data. Many leading automated learning techniques rely on unrealistically large data sets and it is unclear how to leverage prior knowledge effectively to re-identify a model after an abrupt change. In this work, we propose a conceptual framework to recover parsimonious models of a system in response to abrupt changes in the low-data limit. First, the abrupt change is detected by comparing the estimated Lyapunov time of the data with the model prediction. Next, we apply the sparse identification of nonlinear dynamics (SINDy) regression to update a previously identified model with the fewest changes, either by addition, deletion, or modification of existing model terms. We demonstrate this sparse model recovery on several examples for abrupt system change detection in periodic and chaotic dynamical systems. Our examples show that sparse updates to a previously identified model perform better with less data, have lower runtime complexity, and are less sensitive to noise than identifying an entirely new model. The proposed abrupt-SINDy architecture provides a new paradigm for the rapid and efficient recovery of a system model after abrupt changes.

PACS numbers: 02.60.-x, 02.60.Ed, 05.45.-a, 05.45.Pq, 05.45.Tp, 07.05.Kf

Keywords: Dynamical systems, Chaos, Data-driven models, Machine learning, Sparse optimization

Dynamical systems modeling is a cornerstone of modern mathematical physics and engineering. The dynamics of many complex systems (e.g., neuroscience, climate, epidemiology, etc.) may not have first-principles derivations, and researchers are increasingly using data-driven methods for system identification and the discovery of dynamics. Related to discovery of dynamical systems models from data is the *recovery* of these models following abrupt changes to the system dynamics. In many domains, such as aviation, model recovery is mission critical, and must be achieved rapidly and with limited noisy data. This paper leverages recent advances in sparse optimization to identify the fewest terms required to recover a model, introducing the concept of *parsimony of change*. In other words, many abrupt system changes, even catastrophic bifurcations, may be characterized with relatively few changes to the terms in the underlying model. In this work, we show that sparse optimization enables rapid model recovery that is faster, requires less data, is more accurate, and has higher noise robustness than the alternative approach of re-characterizing a model from scratch.

I. INTRODUCTION

The data-driven discovery of physical laws and dynamical systems is poised to revolutionize how we model, predict, and control physical systems. Advances are driven by the confluence of big data, machine learning, and modern perspectives on dynamics and control. However, many modern techniques in machine learning (e.g., neural networks) often rely on access to massive data sets, have limited ability to generalize beyond the attractor where data is collected, and do not readily incorporate known physical constraints. These various limitations are framing many state-of-the-art research efforts around learning algorithms¹, especially as it pertains to generalizability, limited data and *one-shot learning*²⁻⁴. Such limitations also frame the primary challenges and limitations associated with data-driven discovery for real-time control of strongly nonlinear, high-dimensional, multi-scale systems with abrupt changes in the dynamics. Whereas traditional methods often require unrealistic amounts of training data to produce a viable model, this work focuses on methods that take advantage of prior experience and knowledge of the physics to dramatically reduce the data and time required to characterize dynamics. Our methodology is similar in philosophy to the machine learning technique of *transfer learning*⁵, which allows networks trained on one task to be efficiently adapted to another task. Our architecture is designed around the goal of rapidly extracting parsimonious, nonlinear dynamical models that identify only the fewest important interac-

^{a)} Electronic mail: mquade@uni-potsdam.de;

All code for this work is publicly available on Github: <https://github.com/Ohjeah/sparsereg>

tion terms so as to avoid overfitting.

There are many important open challenges associated with data-driven discovery of dynamical systems for real-time tracking and control. When abrupt changes occur in the system dynamics, an effective controller must rapidly characterize and compensate for the new dynamics, leaving little time for recovery based on limited data⁶. The primary challenge in real-time model discovery is the reliance on large quantities of training data. A secondary challenge is the ability of models to generalize beyond the training data, which is related to the ability to incorporate new information and quickly modify the model. Machine learning algorithms often suffer from overfitting and a lack of interpretability, although the application of these algorithms to physical systems offers a unique opportunity to enforce known symmetries and physical constraints (e.g. conservation of mass). Inspired by biological systems, which are capable of extremely fast adaptation and learning based on very few trials of new information^{7–9}, we propose model discovery techniques that leverage an *experiential framework*, where known physics, symmetries, and conservation laws are used to rapidly infer model changes with limited data.

A. Previous work in system identification

There are a wealth of regression techniques for the characterization of system dynamics from data, with varying degrees of generality, accuracy, data requirements, and computational complexity. Classical linear model identification algorithms include Kalman filters^{10–12}, the eigensystem realization algorithm (ERA)¹³, dynamic mode decomposition (DMD)^{14–17}, and autoregressive moving average (ARMA) models^{18,19}, to name only a few. The resulting linear models are ideal for control design, but are unable to capture the underlying nonlinear dynamics or structural changes. Increasingly, machine learning is being used for nonlinear model discovery. Neural networks have been used for decades to identify nonlinear systems²⁰, and are experiencing renewed interest because of the ability to train deeper networks with more data^{1,21–25} and the promise of transformations that linearize dynamics via the Koopman operator^{26,27}. Neural networks show good capacity to recover the dynamics in a so-called “model-free” way^{28,29}. These methods are also known as “reservoir computers”, “liquid state machines”, or “echo state networks”, depending on the context. However, a real-time application is unrealistic, and the output is generally not analytically interpretable. In another significant vein of research, genetic programming^{30,31} is a powerful bio-inspired method that has successfully been applied to system identification^{32–35}, time-series prediction^{36,37} and control^{38,39}. However, evolutionary methods in their pure form, including genetic programming, are computationally complex and thus are not suitable for real-time tracking.

Recently, *interpretability* and *parsimony* have become important themes in nonlinear system identification^{32,33}.

A common goal now is to identify the fewest terms required to represent the nonlinear structure of a dynamical system model while avoiding overfitting⁴⁰. Symbolic regression methods^{33,40–42} are generally appealing for system identification of structural changes, although they may need to be adapted to the low-data limit and for faster processing time. Nonparametric additive regression models^{43–45} require a backfitting loop which allows general transformations, but may be prohibitively slow for real-time applications. Generalized linear regression methods are slightly less general but can be brought to a fast evaluation and sparse representation^{40,42}. These leading approaches to identify dynamical equations from data usually rely on past data and aim at reliable reproduction of a stationary system, i.e. when the underlying equations do not change in the course of time^{33,40,45}.

B. Contributions of this work

In this work, we develop an adaptive modification of the sparse identification of nonlinear dynamics (SINDy) algorithm⁴⁰ for real-time recovery of a model following abrupt changes to the system dynamics. We refer to this modeling framework as *abrupt-SINDy*. Although this is not the only approach for real-time change detection and recovery, parsimony and sparsity are natural concepts to track abrupt changes, focusing on the fewest modifications to an existing model. SINDy already requires relatively small amounts of data⁴⁶, is based on fast regression techniques, and has been extended to identify PDEs^{47,48}, to include known constraints and symmetries⁴⁹, to work with limited measurements⁵⁰ and highly corrupted and noisy data^{51,52}, to include control inputs^{46,53}, and to incorporate information criteria to assess the model quality⁵⁴, which will be useful in abrupt model recovery.

Here, we demonstrate that the abrupt-SINDy architecture is capable of rapidly identifying sparse changes to an existing model to recover the new dynamics following an abrupt change to the system. The first step in the adaptive identification process is to detect a system change using divergence of the prediction from measurements. Next, an existing model is updated with sparse corrections, including parameter variations, deletions, and additions of terms. We show that identifying sparse model changes from an existing model requires less data, less computation, and is more robust to noise than identifying a new model from scratch. The overarching framework is illustrated in Fig. 1.

II. STATE OF THE ART

Recently, sparse regression in a library of candidate nonlinear functions has been used for sparse identification of nonlinear dynamics (SINDy) to efficiently identify a sparse model structure from data⁴⁰. The SINDy architecture bypasses an intractable brute-force search through all possible models, leveraging the fact that many dy-

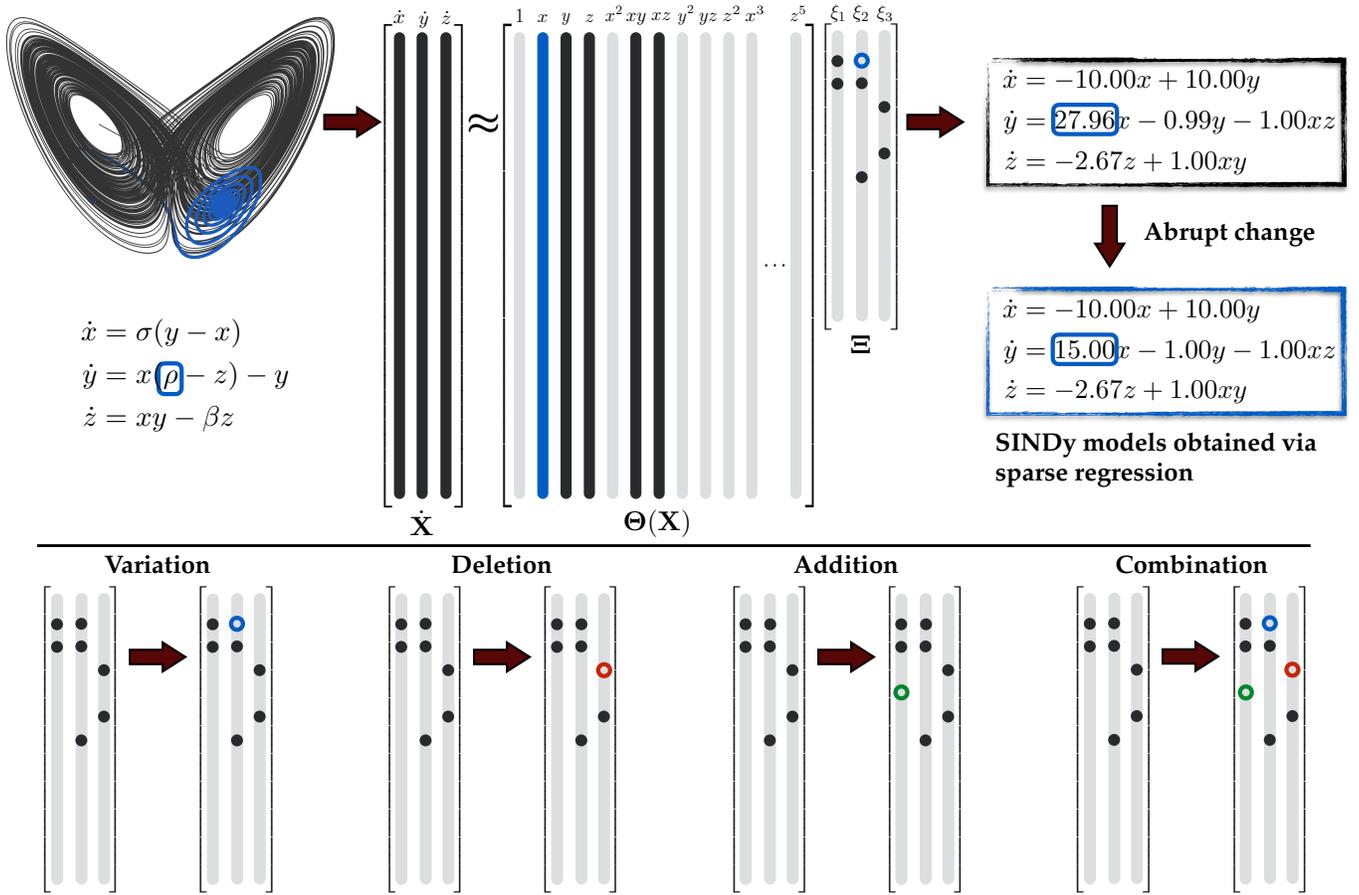


FIG. 1. Schematic overview of abrupt-SINDy method. (top) Illustration of single variation in a term, giving rise to an abrupt change in the dynamics, from black to blue. (bottom) Canonical sparse changes, including parameter variation, addition, deletion, or a combination. The data and results for the top panel are from the example in Sec. IV A below.

namical systems of the form

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}) \quad (1)$$

have dynamics \mathbf{f} that are sparse in the state variable $\mathbf{x} \in \mathbb{R}^n$. Such models may be identified using a sparsity-promoting regression^{55–57} that penalizes the number of nonzero terms ξ_{ij} in a generalized linear model:

$$\hat{f}_i(\mathbf{x}) = \sum_{j=1}^p \xi_{ij} \theta_j(\mathbf{x}), \quad (2)$$

where $\theta_j(\mathbf{x})$ form a set of nonlinear candidate functions. The candidate functions may be chosen to be polynomials, trigonometric functions, or a more general set of functions^{40,42}. With poor choice of the candidate functions θ_j , i.e. if library functions are non-orthogonal and/or overdetermined, the SINDy approach may fail to identify the correct model.

Sparse models may be identified from time-series data, which are collected and formed into the data matrix

$$\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_m]^T. \quad (3)$$

Shifting the data by one time-step yields

$$\mathbf{X}' = [\mathbf{x}_2 \ \mathbf{x}_3 \ \cdots \ \mathbf{x}_{m+1}]^T. \quad (4)$$

Next, we consider a library of candidate nonlinear functions $\Theta(\mathbf{X})$, of the form

$$\Theta(\mathbf{X}) = [1 \ \mathbf{X} \ \mathbf{X}^2 \ \cdots \ \mathbf{X}^d \ \cdots \ \sin(\mathbf{X}) \ \cdots]. \quad (5)$$

Here, the matrix \mathbf{X}^d denotes a matrix with column vectors given by all possible time-series of d -th degree polynomials in the state \mathbf{x} . The terms in Θ can be functional forms motivated by knowledge of the physics. Within the proposed work, they may parameterize a piecewise-affine dynamical model. To preprocess, we mean-subtract and normalize each column of Θ to have unit variance. The dynamical system can now be represented in terms of the data matrices as

$$\mathbf{X}' \approx \Theta(\mathbf{X})\Xi. \quad (6)$$

The coefficients in the column Ξ_k of Ξ determine the active terms in the k -th row of Eq. (2). A parsimonious model has the fewest terms in Ξ required to explain the data. One option to obtain a sparse model is via convex

ℓ_1 -regularized regression:

$$\Xi = \operatorname{argmin}_{\Xi} \|\mathbf{X}' - \Theta(\mathbf{X})\Xi'\|_2 + \gamma\|\Xi'\|_1. \quad (7)$$

The hyper parameter γ balances complexity and sparsity of the solution. Sparse regression, such as LASSO⁵⁵ and sequential thresholded least-squares⁴⁰, improves the robustness of identification for noisy overdetermined data, in contrast to earlier methods⁵⁸ using compressed sensing^{59,60}. Other regularization schemes may be used to improve performance, such as the elastic net regression⁶¹.

In this paper we use the sequentially thresholded ridge regression⁴⁷, which iteratively solves the ridge regression

$$\Xi = \operatorname{argmin}_{\Xi} \|\mathbf{X}' - \Theta(\mathbf{X})\Xi'\|_2 + \alpha\|\Xi'\|_2. \quad (8)$$

and then thresholds any coefficient that is smaller than γ . The procedure is repeated on the non-zero entries of Ξ until the model converges. After a sparse model structure has been identified in normalized coordinates, it is necessary to regress onto this sparse structure in the original unnormalized coordinates. Otherwise, non-physical constant terms appear when transforming back from normalized coordinates due to the mean-subtraction.

In La Cava et al.³⁵ the authors pursue a complementary although more computationally intensive idea of adaptive modeling in the context of generalized linear models. Starting from an initial guess for the model, a brute force search is conducted to scan a larger set of candidate functions $\theta \rightarrow \theta\theta'^\gamma$, where θ' are multiplicative extensions to the initial set of candidate functions and γ are real valued exponents. The intended use of this method is the refinement of first-principle based models by discovery of coupling terms. It is possible to combine this refinement with our proposed scheme for dealing with abrupt changes. In addition, sparse sensors⁶² and randomized algorithms⁶³ may improve speed.

III. METHODS

The viewpoint of sparsity extends beyond model discovery, and we propose to extend SINDy to identify systems undergoing abrupt changes. It may be the case that abrupt model changes will only involve the addition, deletion, or modification of a few terms in the model. This is a statement of the *parsimony of change*, and indicates that we can use sparse regression to efficiently identify the new or missing terms with considerably less data than required to identify a new model from scratch. In general, each additional term that must be identified requires additional training data to distinguish between joint effects. Thus, having only a few changes reduces the amount of data required, making the model recovery more rapid. This section will describe a procedure that extends SINDy to handle three basic types of model changes:

i) Variation of a term. If the structure of the model is unchanged and only the parameters vary, we will perform least-squares regression on the known structure to

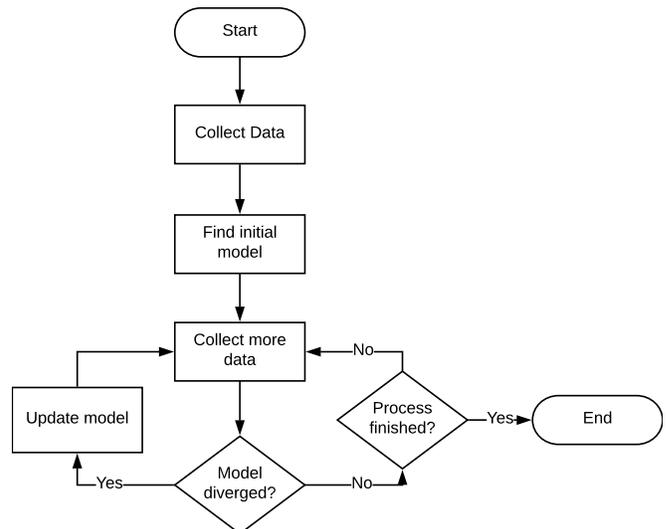


FIG. 2. Adaptive SINDy flow chart. For an initial model and hyper parameter selection, a gridsearch is conducted. Next, we apply a predictor corrector scheme checking every t_{error} for model divergence using estimated Lyapunov time, and eventually update the model in a two step fashion.

identify the new parameters. This is computationally fast, and it is easy to check if the model explains the new dynamics, or if it is necessary to explore possible additions or deletions of terms.

ii) Deletion of a term. If the model changes by the removal of a few terms, then SINDy regression can be applied on the sparse coefficients in order to identify which terms have dropped out.

iii) Addition of a term. If a term is added, then SINDy regression will find the sparsest combination of inactive terms that explain the model error. Since least squares regression scales asymptotically $\mathcal{O}(p^3)$, with p the number of columns in the library, this is computationally less expensive than regression in the entire library.

Combinations of these changes, such as a simultaneous addition and deletion, are more challenging and will also be explored. This approach is known as *abrupt-SINDy*, and it is depicted schematically in Fig. 2.

A. Baseline model

First, we must identify a baseline SINDy model, and we use a gridsearch to determine the optimal hyper parameter selection. In gridsearch, all combinations of hyper parameters are tested and the best performing set is selected. This search is only performed once, locking in hyper parameters for future updates. The baseline model is characterized by the sparse coefficients in Ξ_0 .

B. Detecting model divergence

It is essential to rapidly detect any change in the model, and we employ a classical predictor-corrector scheme¹². The predictor step is performed over a time τ_{pred} in the interval $t, t + \tau_{\text{pred}}$ using the model valid at time t . The divergence of the predicted and measured state is computed at $t + \tau$ as $\|\Delta\mathbf{x}\| = \|\hat{\mathbf{x}}(t + \tau) - \mathbf{x}(t + \tau)\|$, where $\hat{\mathbf{x}}$ is the prediction and \mathbf{x} is the measurement. The idea is to identify when the model and the measurement diverge faster than predicted by the dynamics of the system. For a chaotic system, the divergence of a trajectory is measured by the largest Lyapunov exponent of the system⁶⁴, although a wealth of similar measures have been suggested⁶⁵. The Lyapunov exponent is defined as

$$\lambda = \lim_{\tau \rightarrow \infty} \lim_{\Delta\mathbf{x}(t_0) \rightarrow 0} \frac{\left\langle \log \left(\frac{\Delta\mathbf{x}(t_0 + \tau)}{\Delta\mathbf{x}(t_0)} \right) \right\rangle}{\tau}, \quad (9)$$

and its inverse sets the fastest time scale. Here, the analogy of ensemble and time average is used, more precisely the local, finite-time equivalent^{66,67}. An improvement can be achieved by exploiting an ensemble, e.g. by adding noise to the state \mathbf{x} that corresponds to the given measurement accuracy. Since we know the dynamical system for the prediction step, the Lyapunov exponent is determined by evolving the tangent space with the system^{68,69}.

In our detection algorithm, we fix a fluctuation tolerance $\Delta\mathbf{x}$ and measure if the divergence time we find deviates from the expectation. Formally, the model and measurements diverge if the time-scale given by the local Lyapunov exponent and prediction horizon disagree. The local Lyapunov exponent is computed directly from the eigenvalues of the dynamical system^{68,69}. The prediction horizon $T(t)$ is the first passage time where prediction $\hat{\mathbf{x}}(t + \Delta t)$ with initial condition $\mathbf{x}(t)$ and measurement $\mathbf{x}(t + \Delta t)$ differ by more than $\Delta\mathbf{x}$:

$$T(t) = \operatorname{argmax}_{\Delta t} \|\hat{\mathbf{x}}(t + \Delta t) - \mathbf{x}(t + \Delta t)\| < \|\Delta\mathbf{x}\|. \quad (10)$$

Analogous to the local Lyapunov exponent, we compute the ratio $\log \|\Delta\mathbf{x}(t_0 + \tau) / \Delta\mathbf{x}(t_0)\|$ as a measure for the divergence based on the measurement. For the model, we compute the local Lyapunov exponent as the average maximum eigenvalue $\bar{\lambda}(t) = \langle \lambda(t') \rangle_{t' \in [t, t+T]}$ with $\lambda(t) = \max(\lambda_i(t))$ and $\lambda_i v_i(t) = \partial f_j / \partial \mathbf{x}_k |_{\mathbf{x}(t)} v_i(t)$. Thus we compare the expected and observed trajectory divergence. Model and measurement have diverged at time t if the model time scale and the measured one differ:

$$\bar{\lambda}(t) > \alpha \frac{\log(\Delta\mathbf{x}) - \log(\bar{\Delta}(t))}{T(t)}. \quad (11)$$

If the model is not chaotic, but the measurement is chaotic, one must invert the inequality, as in Fig. 3. The empirical factor α accounts for finite-time statistics.

This method depends heavily on the particular system under investigation, including the dynamics, time scales,

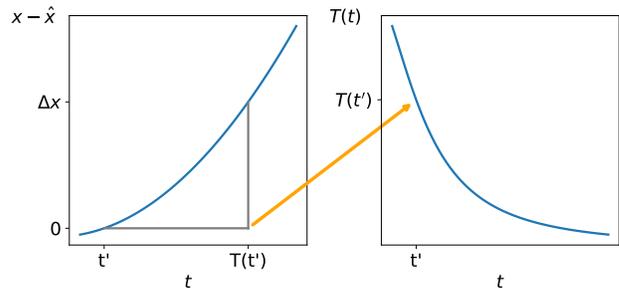


FIG. 3. Sketch of the prediction horizon estimation. We use the observation $\mathbf{x}(t)$ as initial condition for the current model. Integration gives $\hat{\mathbf{x}}(t)$. The prediction horizon $T(t)$ is calculated according to Eq. (10). The prediction horizon is a function of time and the current model. It indicates divergence of model and observation. For details see text.

and sampling rate. In a practical implementation, these considerations must be handled carefully and automatically. It is important to note that we are able to formulate the divergence in terms of dynamical systems theory, because our model *is* a dynamical system, in other cases, such as artificial neural networks, this is not possible due to the limited mathematical framework.

C. Adaptive model fitting

After a change is detected, the following procedure is implemented to rapidly recover the model:

1. First, the new data is regressed onto the existing sparse structure Ξ_0 to identify varying parameters.
2. Next, we identify deletions of terms by performing the sparse regression on the sparse columns of Θ that correspond to nonzero rows in Ξ_0 . This is more efficient than identifying a new model, as we only seek to delete existing terms from the model.
3. Finally, if there is still a residual error, then a sparse model is fit for this error in the inactive columns of Θ that correspond to zero rows in Ξ_0 . In this way, new terms may be added to the model.

If the residual is sufficiently small after any step, the procedure ends. Alternatively, the procedure may be iterated until convergence. We are solving smaller regression problems by restricting our attention to subsets of the columns of Θ . These smaller regressions require less data and are less computationally expensive⁶¹, compared to fitting a new model. The deletion-addition procedure is performed after a model divergence is detected, using new transient data collected in an interval of size t_{update} .

IV. RESULTS

In this section, we describe the results of the abrupt-SINDy framework on dynamical systems with abrupt

Parameter	Value
α	0, 0.2, 0.4, 0.6, 0.8, 0.95
γ	0.1, 0.2, 0.4
n_{degree}	2, 3
n_{fold}	5
Seed	42
CV	k-fold
Score	explained variance score

TABLE I. Parameters for the grid search.

changes, including parameter variation, deletion of terms, and addition of terms. The proposed algorithm is compared against the original SINDy algorithm, which is used to identify a new model from scratch, in terms of data required, computational time, and model accuracy.

In each case, we begin by running a gridsearch algorithm⁷⁰⁷¹ to identify the main parameters: α , the ridge regression regularization parameter; γ , the thresholding parameter; n_{degree} , the maximum degree of the polynomial feature transformation; and n_{fold} , the number of cross-validation runs. For scoring we use the explained variance score and conduct a five-fold cross validation for each point in the $(\alpha, \gamma, n_{\text{degree}})$ parameter grid.

A. Lorenz system

The Lorenz system is a well-studied, and highly simplified, conceptual model for atmospheric circulation⁷²:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - xz - y \\ \dot{z} &= xy - \beta z\end{aligned}\quad (12)$$

where the parameter ρ represents the heating of the atmosphere, corresponding to the Rayleigh number, σ corresponds to Prandtl number, and β to the aspect ratio⁷³. The parameters are set to $\rho = 28$, $\beta = 8/3$, $\sigma = 10$.

In the following we integrate the system numerically to produce a reference data set. We deliberately change the parameter ρ at $t = 40$ to $\rho = 15$ and at $t = 80$ back to $\rho = 28$, as shown in Figs. 4 and 5. These parametric changes lead to a bifurcation in the dynamics, and they are detected quickly. The subsequent adapted parameters are accurately detected up to two digits, as shown in Table II. Because we are identifying the sparse model structure on a normalized library Θ , with zero mean and unit variance, we must de-bias the parameter estimates by computing a least-squares regression onto the sparse model structure in the original unnormalized variables. Otherwise, computing the least-squares regression in the normalized library, as is typically recommended in machine learning, would result in non-physical constant terms in the original unnormalized coordinates.

Abrupt changes to the system parameters are detected using the prediction horizon from Eq. (10). When the system changes, the prediction horizon of the system should decrease, with smaller horizon corresponding to

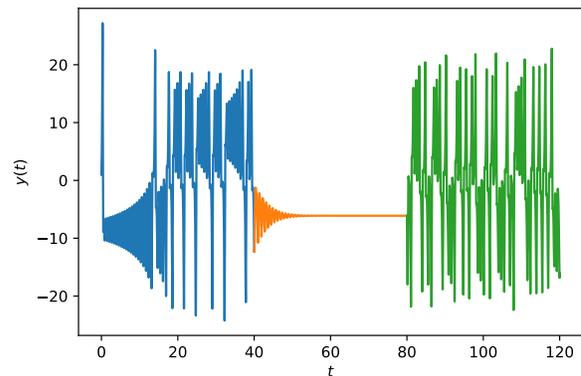


FIG. 4. Time-series of the y coordinate of the Lorenz system. The blue and green segments correspond to the system parameters $\sigma = 10, \rho = 28, \beta = \frac{8}{3}$. The orange segment from $t = 40$ and $t = 80$ corresponds to the modified parameter $\rho = 15$. The initial condition is $\mathbf{x}_0 = (1, 1, 1)$.

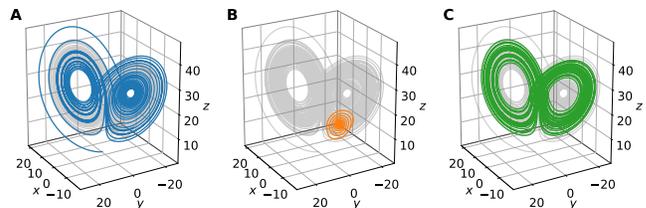


FIG. 5. Lorenz system: Colors and parameters as in Fig. 4. In **A**, **B**, and **C** we show the first, second, and third segments of the trajectory in color with the concatenated trajectory in grey. The system changes from a butterfly attractor to a stable fixed point and back to a butterfly attractor.

a more serious change. Conversely, the inverse time, corresponding to the Lyapunov exponent, should diverge. Figure 6 exhibits this expected behavior. After a change is detected the model is rapidly recovered as shown in Table II. It is important to confirm that the updated model accurately represents the structure of the true dynamics. Figure 6 shows the norm of the model coefficients, $\|\xi - \hat{\xi}\|$, which is a measure of the distance between the estimated and true systems. Except for a short time ($t_{\text{update}} = 1$) after the abrupt change, the identified model closely agrees with the true model.

1. Effects of noise and data volume

An important set of practical considerations include how noise and the amount of data influence the speed of change detection and the accuracy of subsequent model recovery. Both the noise robustness and the amount of data required will change for a new problem, and here we report trends for this specific case. In addition, the amount of data required is also related to the sampling rate, which is the subject of ongoing investigation; in some cases, higher sampling time may even degrade model performance due to numerical effects⁷⁴.

Figure 7 shows the model fit following the abrupt change, comparing both the abrupt-SINDy method, which uses information about the existing model struc-

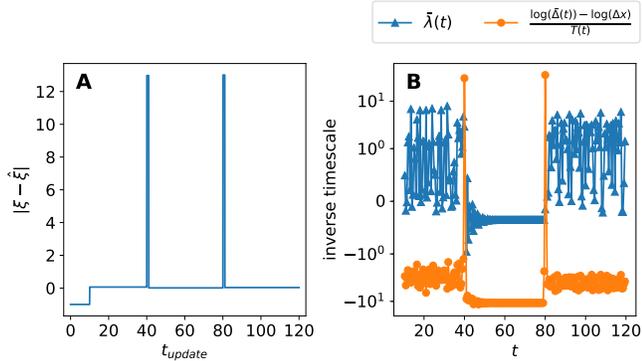


FIG. 6. Lorenz system: **A** We show the model accuracy over time. For coefficients, see Table II. For $t \leq 10$ no model is available and $\|\xi - \hat{\xi}\| = -1$. At both switch points, $t = 40$ and $t = 80$, $t_{\text{update}} = 1$ is needed to update the model. During this interval, a fallback solution, e.g. DMD could be implemented. Note that the accuracy metric requires knowledge about the ground truth and thus is only available in a hindcast scenario. **B** Evaluation of Eq. (10). At both switch points, $t = 40$ and $t = 80$, we quickly detect the divergence of model and measurement. The parameters are $t_{\text{model}} = 10$, $t_{\text{update}} = 1$, $t_{\text{error}} = 0.5$, and $\Delta \mathbf{x} = 1.0$.

t_{detected}	t_{update}	Equations
0.00	10.0	$\dot{x} = -10.0x + 10.0y$ $\dot{y} = 27.96x - 0.99y - 1.0xz$ $\dot{z} = -2.67z + 1.0xy$
40.01	41.0	$\dot{x} = -10.0x + 10.0y$ $\dot{y} = 15.0x - 1.0y - 1.0xz$ $\dot{z} = -2.67z + 1.0xy$
80.02	81.0	$\dot{x} = -10.0x + 10.0y$ $\dot{y} = 27.98x - 1.0y - 1.0xz$ $\dot{z} = -2.67z + 1.0xy$

TABLE II. Lorenz system: detection and update times, along with identified equations. The detection time coincides up to the second digit with the true switching time. The rapidly identified model agrees well with the true model structure and parameters. Coefficients are rounded to the second digit.

ture, and the standard SINDy method, which re-identifies the model from scratch following a detected change. In this figure, the model quality is shown as a function of the amount of data collected after the change.

The abrupt-SINDy model is able to identify more accurate models in a very short amount of time, given by $t_{\text{update}} \approx 0.1$. At this point, the standard SINDy method shows comparable error, however for even smaller times, the data are no longer sufficient for the conventional method. Since the adaptive method starts near the optimal solution, larger data sets do not degrade the model, which was an unexpected additional advantage.

Figure 8 explores the effect of noise on the abrupt-SINDy and standard SINDy algorithms. Abrupt-SINDy has considerably higher noise tolerance than the standard algorithm, as it must identify fewer unknown coefficients. In fact, it is able to handle approximately an order of

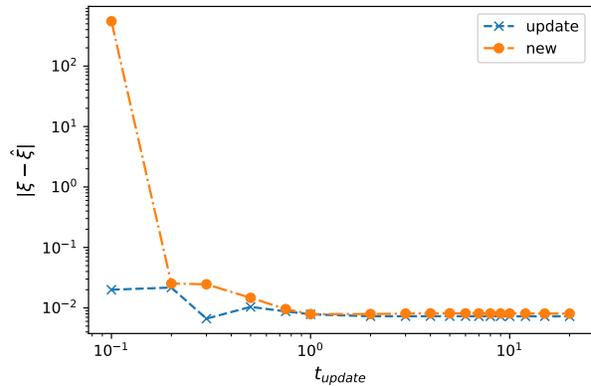


FIG. 7. Lorenz system: We show the model accuracy versus the amount of data used to update (blue \times) or re-fit (orange dot) respectively. Data is collected from in the interval $[40, 40 + t_{\text{update}}]$ just after the first change of the system dynamics. The number of data points for $t_{\text{update}} = 0.1$ are $N = 25$, for $t_{\text{update}} = 10$ we have 2500 points. At $t_{\text{update}} \simeq 1$, updating and re-fitting methods become comparable. However, for smaller update times, or less data, respectively, the fraction of transient data becomes too small for identifying the exact model from scratch. Updating the model needs less data for the same accuracy or achieves higher accuracy with the same amount of data.

magnitude more noise before failing to identify a model. Generally, increasing the volume of data collection improves the model. The critical point in the abrupt-SINDy curves corresponds to when small but dynamically important terms are mis-identified as a result of insufficient signal-to-noise. Although the noise and chaotic signal cannot be easily distinguished for small signal-to-noise, it may be possible to distinguish between them using a spectral analysis, since chaos yields red noise in contrast to the white additive noise.

B. Van der Pol

As a second example, we consider the famous nonlinear Van der Pol oscillator⁷⁵. We include additional quadratic nonlinearities αx^2 and αy^2 to study the ability of our method to capture structural changes when these terms are added and removed abruptly. This example focuses on the important class of periodic phenomena, in contrast to the chaotic Lorenz dynamics. The modified Van der Pol oscillator is described by the following equations:

$$\begin{aligned} \dot{x} &= y - \alpha y^2 \\ \dot{y} &= \mu(1 - x^2)y - x + \alpha x^2, \end{aligned} \quad (13)$$

where $\mu > 0$ is a parameter controlling the nonlinear damping, and α parameterizes the additional quadratic nonlinearity. The reference data set is shown in Fig. 9, with $\mu = 7.5$ and $\alpha = 0$ for $t \in [0, 100]$, which results in a canonical periodic orbit. At $t = 100$ we introduce a structural change, switching on the quadratic nonlin-

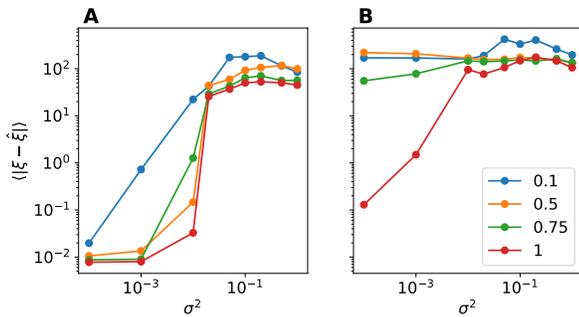


FIG. 8. Lorenz system: We show the noise robustness of model accuracy. In A) we use the previous knowledge and update the model; in B) we make a new fit only re-using the previously discovered hyper-parameters. The curves are parametrized by t_{update} , c.f. Fig. 7. The accuracy measure is very noise sensitive, as distinction between library functions gets lost. At a signal to noise ratio of approximately 1, no accurate model can be obtained with either model. At lower noise ratios, updating the model achieves higher accuracy (the library is smaller). In both cases, accuracy scales approximately logarithmically with t_{update} .

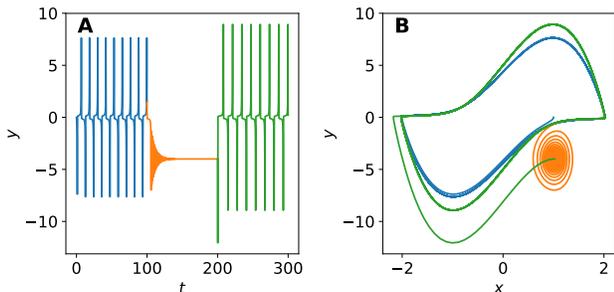


FIG. 9. Van der Pol system with parameters $\mu = 5, \alpha = 0$ (blue), $\mu = 7.5, \alpha = -0.25$ (orange), $\mu = 6.0, \alpha = 0$ (green). **A**: time evolution of the y -coordinate. **B** phase-space-trajectory x, y .

erity ($\alpha = -0.25$), and driving the system to a stable fixed point. We also modify the parameter μ , setting it to $\mu = 6.0$. Finally, at $t = 200$, we switch off the additional nonlinearity ($\alpha = 0$) and keep $\mu = 6$.

Table III shows the corresponding models recovered using the abrupt-SINDy method. The change is detected using the Lyapunov time defined in Eq. (11), as shown in Fig. 10. Again, the estimated Lyapunov time (Fig. 10 B) captures the changes in the model, which correspond to peaks in structural model error (Fig. 10 A). While the first and third stage are indeed identified correctly, the term $-1.25x$ is preferred over $-x - 0.25x^2$ in the sparse estimate for \dot{y} in the orange trajectory. However, since both terms look similar near the fixed point at $x \sim 1$, this describes the dynamics well. This type of mis-identification often occurs in data mining when features are highly correlated⁶¹ and is more related to sparse regression in general than the proposed abrupt-SINDy. For dynamic system identification, the correct nonlinearity could be resolved by obtaining more transient data, i.e. by perturbing the system through actu-

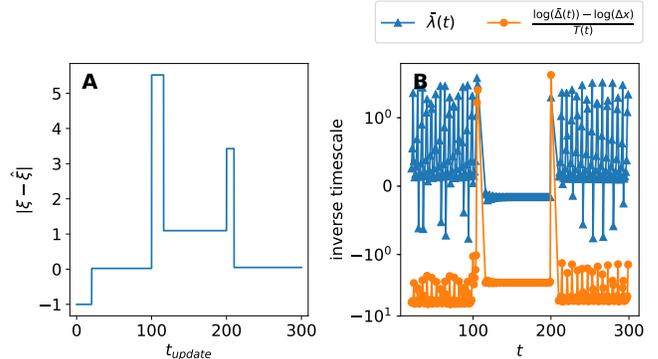


FIG. 10. Van der Pol system: Evaluation of Eq. (10). Parameters: $t_{\text{model}} = 20, t_{\text{update}} = 10, t_{\text{error}} = 1, \Delta x = 1.5$.

t_{detected}	t_{update}	Equations
0.00	20.01	$\dot{x} = 1.0y$ $\dot{y} = -1.0x + 4.99y - 4.99x^2y$
106.39	116.00	$\dot{x} = 0.99y + 0.25y^2$ $\dot{y} = -1.26x + 7.46y - 7.46x^2y$
200.12	210.00	$\dot{x} = 1.0y$ $\dot{y} = -1.0x + 5.98y - 5.98x^2y$

TABLE III. Van der Pol system: Summary of the discovered equations. Coefficients are rounded to the second digit.

ation. However, this model may be sufficient for control while a more accurate model is identified.

V. CONCLUSIONS

In this work, we develop an adaptive nonlinear system identification strategy designed to rapidly recover nonlinear models from limited data following an abrupt change to the system dynamics. The sparse identification of nonlinear dynamics (SINDy) framework is ideal for change detection and model recovery, as it relies on parsimony to select the fewest active terms required to model the dynamics. In our adaptive abrupt-SINDy method, we rely on previously identified models to identify the fewest *changes* required to recover the model. This modified algorithm is shown to be highly effective at model recovery following an abrupt change, requiring less data, less computation time, and having improved noise robustness over identifying a new model from scratch. The abrupt-SINDy method is demonstrated on several numerical examples exhibiting chaotic dynamics and periodic dynamics, as well as parametric and structural changes, enabling real-time model recovery.

There are several promising directions that may be pursued to improve the abrupt-SINDy method:

1. **Fallback models:** In the current implementation, after a change has been detected, the old model will be used until enough data is collected to identify a new model. The dynamic mode decomposition¹⁷

provides an alternative fallback model, that may be identified rapidly with even less data. Additionally, instead of relying on a sparse update to the current model, it is sensible to also maintain a library of past models for rapid characterization⁷⁶.

2. **Hyperparameterization:** In the initial prototype, the hyper-parameters Δ_x and t_{update} are fixed. Over time, an improved algorithm may learn and adapt optimal hyper-parameters.
3. **Comprehensive Lyapunov time estimation:** According to Eq.(10), the Lyapunov time $T(t|\Delta\mathbf{x})$ is estimated for a fixed $\Delta\mathbf{x}$. Estimating the time for a range of values, i.e. $\Delta\mathbf{x} \in (0, \Delta\mathbf{x}_{\text{max}}]$, will be more robust and may provide a richer analysis without requiring additional data.
4. **Advanced optimization and objectives:** Looking forward, advanced optimization techniques may be used to further improve the adaptation to system changes. Depending on the system, other objectives may be optimized, either by including regularization or in a multi-objective optimization.

The proposed abrupt-SINDy framework is promising for the real-time recovery of nonlinear models following abrupt changes. Future work will be required to demonstrate this method on more sophisticated engineering problems and to incorporate it in controllers. The abrupt-SINDy modeling framework may also help inform current rapid learning strategies in neural network architectures²⁻⁴, potentially allowing dynamical systems methods to inform rapid training paradigms in deep learning.

ACKNOWLEDGEMENTS

MQ was supported by a fellowship within the FITweltweit program of the German Academic Exchange Service (DAAD). MQ and MA acknowledge support by the European Erasmus SME/HPC project (588372-EPP-1-2017-1-IE-EPPKA2-KA). SLB acknowledges support from the ARO and AFOSR Young Investigator Programs (ARO W911NF-17-1-0422 and AFOSR FA9550-18-1-0200). SLB and JNK acknowledge support from DARPA (HR0011-16-C-0016). We also acknowledge valuable discussions related to abrupt model recovery and programming wisdom with Dennis Bernstein, Karthik Duraisamy, Thomas Isele, Eurika Kaiser, and Hod Lipson.

REFERENCES

- ¹I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, Vol. 1 (MIT press Cambridge, 2016).
- ²L. Fei-Fei, R. Fergus, and P. Perona, IEEE transactions on pattern analysis and machine intelligence **28**, 594 (2006).
- ³O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, in *Advances in Neural Information Processing Systems* (2016) pp. 3630–3638.
- ⁴C. B. Delahunt and J. N. Kutz, arXiv preprint arXiv:1802.05405 (2018).
- ⁵S. J. Pan and Q. Yang, IEEE Transactions on knowledge and data engineering **22**, 1345 (2010).
- ⁶S. L. Brunton and B. R. Noack, Applied Mechanics Reviews **67**, 050801 (2015).
- ⁷C. H. Rankin, Curr. Biol. **14**, R617 (2004).
- ⁸J. R. Whitlock, Science **313**, 1093 (2006).
- ⁹J. P. Johansen, C. K. Cain, L. E. Ostroff, and J. E. LeDoux, Cell **147**, 509 (2011).
- ¹⁰R. E. Kalman, Journal of Fluids Engineering **82**, 35 (1960).
- ¹¹B. L. Ho and R. E. Kalman, in *Proc. of the 3rd Ann. Allerton Conference on Circuit and System Theory* (1965) pp. 449–459.
- ¹²N. A. Gershenfeld, *The Nature of Mathematical Modeling* (Cambridge University Press, 1999).
- ¹³J. N. Juang and R. S. Pappa, Journal of Guidance, Control, and Dynamics **8**, 620 (1985).
- ¹⁴P. J. Schmid, Journal of Fluid Mechanics **656**, 5 (2010).
- ¹⁵C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson, J. Fluid Mech. **645**, 115 (2009).
- ¹⁶J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, Journal of Computational Dynamics **1**, 391 (2014).
- ¹⁷J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition* (Society for Industrial and Applied Mathematics, 2016).
- ¹⁸H. Akaike, Annals of the institute of Statistical Mathematics **21**, 243 (1969).
- ¹⁹P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, Springer Texts in Statistics (Springer, 1996).
- ²⁰R. Gonzalez-Garcia, R. Rico-Martinez, and I. Kevrekidis, Computers & chemical engineering **22**, S965 (1998).
- ²¹E. Yeung, S. Kundu, and N. Hodas, arXiv preprint arXiv:1708.06850 (2017).
- ²²N. Takeishi, Y. Kawahara, and T. Yairi, in *Advances in Neural Information Processing Systems* (2017) pp. 1130–1140.
- ²³C. Wehmeyer and F. Noé, arXiv preprint arXiv:1710.11239 (2017).
- ²⁴A. Mardt, L. Pasquali, H. Wu, and F. Noé, Nature Communications **9** (2018).
- ²⁵B. Lusch, J. N. Kutz, and S. L. Brunton, arXiv preprint arXiv:1712.09707 (2017).
- ²⁶B. O. Koopman, Proceedings of the National Academy of Sciences **17**, 315 (1931).
- ²⁷I. Mezić, Nonlinear Dyn **41**, 309 (2005).
- ²⁸M. Lukoševičius and H. Jaeger, Comp. Sci. Rev. **3**, 127 (2009).
- ²⁹Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, Chaos **27**, 041102 (2017).
- ³⁰G. B. Dantzig, R. Cottle, and Y. P. Aneja, *Mathematical programming: Essays in honor of George B. Dantzig*, Mathematical Programming: Essays in Honor of George B. Dantzig No. Bd. 1 (North-Holland, 1985).
- ³¹J. R. Koza, *Genetic programming: On the programming of computers by means of natural selection*, Vol. 1 (MIT press, 1992).
- ³²J. Bongard and H. Lipson, Proceedings of the National Academy of Sciences **104**, 9943 (2007).
- ³³M. Schmidt and H. Lipson, Science **324**, 81 (2009).
- ³⁴M. D. Schmidt, R. R. Vallabhajosyula, J. W. Jenkins, J. E. Hood, A. S. Soni, J. P. Wikswo, and H. Lipson, Phys. Biol. **8**, 055011 (2011).
- ³⁵W. G. La Cava and K. Danai, Int. J. Syst. Sci. **47**, 249 (2015).
- ³⁶W. La Cava, K. Danai, L. Spector, P. Fleming, A. Wright, and M. Lackner, Renewable Energy **87**, 892 (2016).
- ³⁷M. Quade, M. Abel, K. Shafi, R. K. Niven, and B. R. Noack, Phys. Rev. E **94** (2016), 10.1103/physreve.94.012214.
- ³⁸J. Gout, M. Quade, K. Shafi, R. K. Niven, and M. Abel, Nonlinear Dyn **91**, 1001 (2017).

- ³⁹T. Duriez, S. L. Brunton, and B. R. Noack, *Machine Learning Control – Taming Nonlinear Dynamics and Turbulence*, Fluid Mechanics and Its Applications, Vol. 116 (Springer, 2017).
- ⁴⁰S. L. Brunton, J. L. Proctor, and J. N. Kutz, Proceedings of the National Academy of Sciences **113**, 3932 (2016).
- ⁴¹H. Voss, M. Büchner, and M. Abel, Phys. Rev. E **57**, 2820 (1998).
- ⁴²T. McConaghy, in *Genetic and Evolutionary Computation* (Springer New York, 2011) pp. 235–260.
- ⁴³M. Abel, K. Ahnert, J. Kurths, and S. Mandelj, Phys. Rev. E **71**, 015203 (2005).
- ⁴⁴H. U. Voss, P. Kolodner, M. Abel, and J. Kurths, Phys. Rev. Lett. **83**, 3422 (1999).
- ⁴⁵M. Abel, Int. J. Bifurcation Chaos **14**, 2027 (2004).
- ⁴⁶E. Kaiser, J. N. Kutz, and S. L. Brunton, arXiv preprint arXiv:1711.05501 (2017).
- ⁴⁷S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Sci. Adv. **3**, e1602614 (2017).
- ⁴⁸H. Schaeffer, in *Proc. R. Soc. A*, Vol. 473 (The Royal Society, 2017) p. 20160446.
- ⁴⁹J.-C. Loiseau and S. L. Brunton, Journal of Fluid Mechanics **838**, 42 (2018).
- ⁵⁰S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, Nat Commun **8** (2017), 10.1038/s41467-017-00030-8.
- ⁵¹G. Tran and R. Ward, arXiv preprint arXiv:1607.01067 (2016).
- ⁵²H. Schaeffer and S. G. McCalla, Physical Review E **96**, 023302 (2017).
- ⁵³S. L. Brunton, J. L. Proctor, and J. N. Kutz, IFAC NOLCOS **49**, 710 (2016).
- ⁵⁴N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, Proc. R. Soc. A **473**, 20170009 (2017).
- ⁵⁵R. Tibshirani, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **73**, 273 (2011).
- ⁵⁶T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning*, Vol. 2 (Springer New York, 2001).
- ⁵⁷G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning* (Springer New York, 2013).
- ⁵⁸W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, Phys. Rev. Lett. **106**, 154101 (2011).
- ⁵⁹D. Donoho, IEEE Trans. Inform. Theory **52**, 1289 (2006).
- ⁶⁰E. J. Candès, Proceedings of the International Congress of Mathematics (2006).
- ⁶¹J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, CSUR **50**, 1 (2017).
- ⁶²K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, Invited for IEEE Control Systems Magazine (arXiv eprint arXiv:1701.07569) (2017).
- ⁶³N. B. Erichson, S. Voronin, S. L. Brunton, and J. N. Kutz, To appear in Journal of Statistical Software (arXiv preprint arXiv:1608.02148) (2018).
- ⁶⁴E. Ott, *Chaos in Dynamical Systems* (Cambridge University Press, 2002).
- ⁶⁵E. Schöll and H. G. Schuster, *Handbook of Chaos Control* (Wiley-VCH Verlag GmbH & Co. KGaA, 2007).
- ⁶⁶Y.-C. Lai and T. Tél, *Transient Chaos*, Vol. 173 (Springer New York, 2011).
- ⁶⁷R. Ding and J. Li, Phys. Lett. A **364**, 396 (2007).
- ⁶⁸H. Kantz, Phys. Lett. A **185**, 77 (1994).
- ⁶⁹A. Pikovsky and A. Politi, Nonlinearity **11**, 1049 (1998).
- ⁷⁰A. Abraham, F. Pedregosa, M. Eickenberg, P. Gervais, A. Mueller, J. Kossaifi, A. Gramfort, B. Thirion, and G. Varoquaux, Front. Neuroinform. **8**, 2825 (2014).
- ⁷¹The user manual is located at http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.
- ⁷²E. N. Lorenz, Journal of the atmospheric sciences **20**, 130 (1963).
- ⁷³S. H. Strogatz, *Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering* (Hachette UK, 2014).
- ⁷⁴K. Ahnert and M. Abel, *Computer Physics Communications*, Comput. Phys. Commun. **177**, 764 (2007).
- ⁷⁵B. Van der Pol, Radio Review **1**, 701 (1920).
- ⁷⁶S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz, SIAM Journal on Applied Dynamical Systems **13**, 1716 (2014).