IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 18 July 2024, accepted 8 September 2024, date of publication 20 September 2024, date of current version 3 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3465390

RESEARCH ARTICLE

Learning Nonlinear Dynamics Using Kalman Smoothing

JACOB M. STEVENS-HAAS[1], YASH BHANGALE[2], J. NATHAN KUTZ[1,3], (Senior Member, IEEE), AND ALEKSANDR ARAVKIN[1]

[1]Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA
[2]Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA
[3]Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195, USA

Corresponding author: Jacob M. Stevens-Haas (jacob.stevens.haas@gmail.com)

This work was supported in part by U.S. National Science Foundation (NSF) Artificial Intelligence (AI) Institute for Dynamical Systems (dynamicsai.org) under Grant 2112085, and in part by U.S. Department of Veterans Affairs under the Post-9/11 GI Bill.

ABSTRACT Identifying Ordinary Differential Equations (ODEs) from measurement data requires both fitting the dynamics and assimilating, either implicitly or explicitly, the measurement data. The *Sparse Identification of Nonlinear Dynamics* (SINDy) method does so in two steps: a derivative estimation and smoothing step and a sparse regression step on a library of ODE terms. Previously, the derivative step in SINDy and its python package, pysindy, used finite difference, L1 total variation minimization, or local filters like Savitzky-Golay. We have incorporated Kalman smoothing, along with hyperparameter optimization, into the existing `pysindy` architecture, allowing for rapid adoption of the method. Kalman smoothing is a classical framework for assimilating the measurement data with known noise statistics. As a first SINDy step, it denoises the data by applying a prior belief that the system is an instance of integrated Brownian motion. We conduct numerical experiments on eight dynamical systems show Kalman smoothing to be the best SINDy differentiation/smoothing option in the presence of noise on four of those systems, and tied for three of them. It has particular advantage at preserving problem structure in simulation. The addition of hyperparameter optimization further makes it the most amenable method for generic data. In doing so, it is the first SINDy method for noisy data that requires only a single hyperparameter, and it gives viable results in half of the systems we test.

INDEX TERMS Dynamical systems, machine learning, sparse regression, optimization, Kalman smoothing, SINDy, differential equations.

I. INTRODUCTION

The method of *Sparse Identification of Nonlinear Dynamics* (SINDy) [1], [2] seeks to discover a differential or partial differential equation governing an arbitrary, temporally measured system. The method takes as input some coordinate measurements over time, such as angles between molecular bonds [3] or a spatial field, such as wave heights [4], and returns the best ordinary or partial differential equation (ODE or PDE) from a library of candidate terms. However, the method struggles to accommodate significant measurement noise, which is typical of real-world systems. On the other hand, Kalman theory [5], [6] has a half-century history of assimilating measurement noise to smooth a trajectory, with well-studied and rigorously characterized noise properties [7]. We integrate the mature and well-established theory of Kalman with the emerging SINDy technology and combine with generalized cross validation (GCV) parameter selection for systematic practical applications. The aim is to improve the robustness of SINDy to noisy data. Our Kalman SINDy architecture is shown to be competitive with other combinations of data smoothing and system identification techniques, and has a significant advantage in preservation of problem structure and ease of parameter selection.

Model discovery methods are emerging as a critical component in data-driven engineering design and scientific discovery. Enabled by advancements in computational power, optimization schemes, and machine learning algorithms,

The associate editor coordinating the review of this manuscript and approving it for publication was Halil Ersin Soken.

such techniques are revolutionizing what can be achieved from sensor measurements deployed in a given system. Of interest here is the discovery of dynamic models, which can be constructed from a diversity of techniques, including simple regression techniques such as the *dynamic mode decomposition* (DMD) [8], [9] to neural networks such as *physics-informed neural networks* (PINNs) [10]. In such models, the objective is to construct a proxy model for the observed measurements which an be used to characterize and reconstruct solutions. While DMD provides an interpretable linear model in terms of a modal decomposition, most neural network architectures remain black-box without a clear view of the underlying dynamical processes. Although the number of techniques available are beyond the scope of this paper to review [11], [12], SINDy is perhaps the leading data-driven model discovery method for interpretable and/or explainable dynamic models as it looks to infer the governing equations underlying the observed data. As such, it discovers relationships between spatial and/or temporal derivatives, the underlying mathematical representation of physics and engineering systems since the time of Newton.

The SINDy regression architecture seeks to robustly establish relationships between derivatives. Emerging from [1] and [2], all variants aim to discover a sparse symbolic representation of an autonomous or controlled system, $\dot{x} = f(x)$. As a first step, it estimates derivatives.

This paper introduces Kalman smoothing as the derivative estimation step in SINDy in distinction with the L1 total variation minimization or Savitzky-Golay smoothers common in application. It is not the first to combine Kalman methods with SINDy; [13] utilize Ensemble Kalman Filtering (EKF) to identify a partially-known system as a portion of a multi-step method, and [14] apply Kalman filtering to the ODE coefficients as a way of modeling a non-stationary but separable system. This paper's introduction of Kalman smoothing a continuous process loss for derivative estimation, on the other hand, begins to align the derivative estimation step to the symbolic regression step. It allows engineering applications to incorporate SINDy estimation with a well-established and familiar data assimilation technique whose noise properties are well understood.

Section II describes the individual methods of SINDy and Kalman smoothing, providing some literature review. Section III briefly describes the engineering, rather than methodological, contributions of this work. In section IV, experiments demonstrate that Kalman with GCV is able to remove the need for a hyperparameter and successfully smooth systems and that Kalman smoothing is an effective differentiation step in SINDy. The paper concludes with avenues for future research in section four.

II. BACKGROUND

A. SINDy

SINDy [1] is a family of emerging methods for discovering the underlying dynamics of a system governed by unknown or partially-known [15] differential equations. It can handle

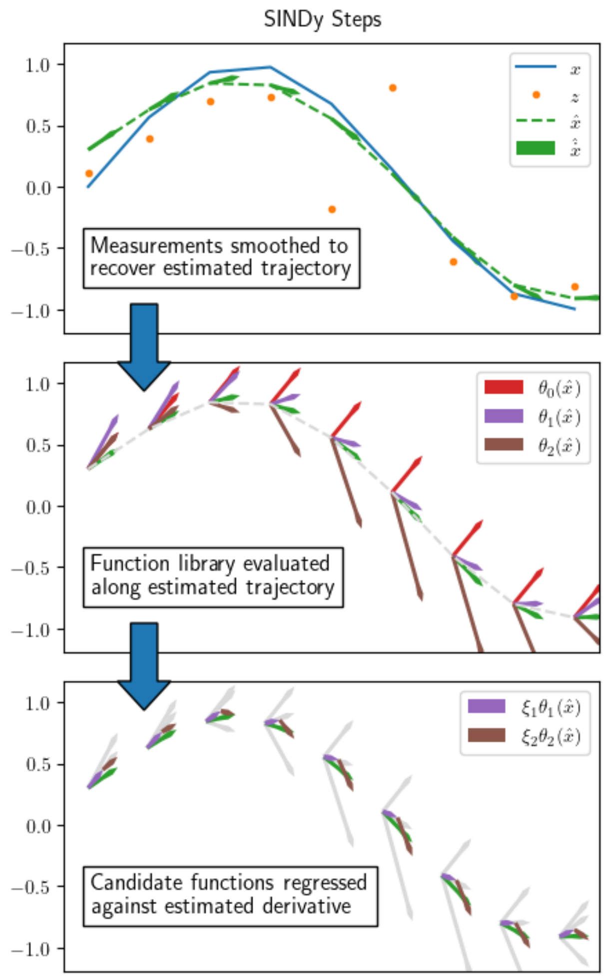


FIGURE 1. The SINDy method, for example, applied to fitting a sinusoid. It takes noisy data and smoothes it in the first panel, using the smoothed values to evaluate a library of functions at each point (second panel). Finally,it identifies the model $\dot{x} = \xi_1\theta_1(x) + \xi_2\theta_2(x)$, rejecting θ_0.

ODEs as well as PDEs [4], and has been used for chemical reaction networks [16], plasma physics [17], and more. Most invocations occur through the `pysindy` Python package, but innovations such as Langevin Regression [18] or [19] exist as independent code.

Given some variable of interest X and a library of functions Θ (including spatial derivatives, when relevant) SINDy seeks to find the coefficients Ξ of the differential equation:

$$\dot{X} = \Xi\Theta(X), \tag{1}$$

where

$X \in \mathbb{R}^{n\times m} = x(t_1)\ldots x(t_m)$: system of n coordinates at m timepoints.
$\Theta(X) \in \mathbb{R}^{p\times m}$: library of p functions evaluated at m timepoints
$\Xi \in \mathbb{R}^{n\times p}$: coefficients for n equations of p functions

The function library, written as a time-independent quantity, refers to the collection $\Theta = [\theta_1, \ldots \theta_p]^T$, where $\theta_i : \mathbb{R}^n \to \mathbb{R}$. Examples include the family of all degree-2 polynomials of n inputs, mixed sines and cosines of certain frequencies, or any user-specified family.

The method generally presumes the measurements (Z) faithfully reflect system state (X) and proceeds in two steps:

1) Estimate the time derivatives of the system $\widehat{\dot{X}} = \mathrm{F}(Z)$ for some smoothing function F.
2) Choosing a sparse regression method, solve the problem $\arg\min_{\text{sparse } \Xi} \left\| \widehat{\dot{X}} - \Xi\Theta(X) \right\|^2$.

This general process is sketched out in a representative scenario in Fig. 1. Each step described there has various innovations and options. Researchers have tried a few different methods for calculating the derivatives, broadly grouped into global methods (e.g. L-1 total variation minimization of [20]) and local methods (e.g. Savitzky-Golay smoothing). Different ways of applying sparsity has attracted more attention, including sequentially thresholding linear regression, nonconvex penalties such as L-0 with a relaxation-based minimization method [15], [21], an L-0 constraint [22], and Bayesian methods for a prior distribution such as spike-slab or regularized horseshoe priors [23], [24]. The latter two papers also demonstrate an interesting line of innovation, eschewing derivatives and using the integral of function library in the loss term. A related approach instead uses the weak form of the differential equation. Most of these methods can benefit from ensembling the data and library terms, as in [25], but others, such as [26] and [27], for identifying Galerkin modes of globally stable fluid flows, require a specific form of function library.

This paper seeks to make SINDy more resilient to noise by taking a data assimilation approach. It instead presents the Kalman SINDy steps:

1) Estimate the state and time derivatives of the system:
$$\widehat{\dot{X}}, \widehat{X} = \arg\min_{X, \dot{X}} \mathrm{L}(X, \dot{X}, Z) = \mathrm{F}(Z) \tag{2}$$
where F applies Kalman smoothing and L is its particular loss function.
2) Choosing a sparse regression method, solve the problem
$$\arg\min_{\text{sparse } \Xi} \left\| \widehat{\dot{X}} - \Xi\Theta(\widehat{X}) \right\|^2. \tag{3}$$

A range of methodological innovations have been introduced into the SINDy discovery framework to make it robust, including the aforementioned *weak form* optimization by Messenger and Bortz [28], [29]. This approach solves the sparse regression problem after integrating the data over random control volumes, providing a dramatic improvement to the noise robustness of the algorithm. Weak form optimization may be thought of as a generalization of the integral SINDy [30] to PDE-FIND. Further improvements to noise robustness and limited data may be obtained through ensembling techniques [25], which use robust statistical bagging to learn inclusion probabilities for the sparse terms ξ, similar to Bayesian inference [23], [24], [31]. Many methodological innovations are integrated in the open-source PySINDy software library [32], reducing the barrier to entry when applying these methods to new problems. Additional techniques for learning dynamics from data include PDE-NET [33], kernel methods [34] and the Bayesian PDE discovery from data [35]. Symbolic neural net learning has also been developed, including symbolic learning on graphs [36], [37], [38].

B. KALMAN SMOOTHING

Kalman filtering and smoothing refers to a group of optimal estimation techniques to assimilate measurement noise to a random process. Filtering refers to incorporating new measurements in real-time, while smoothing refers to estimating the underlying state or signal using a complete trajectory of (batch) measurements. Kalman smoothing and filtering are widely used in engineering design for real-world control and prediction, such as tracking and navigation, in radar systems, econometric variables, and weather prediction [39]. The Kalman smoother can be considered as a best-fit Euler update, as the maximum likelihood estimator of integrated Brownian motion, or as the best linear fit of an unknown system. While the processes this paper is concerned with are not random, in the first step of SINDy they are unknown, and so probabilistic language is appropriate.While the processes this paper is concerned with are not random, in the first step of SINDy they are unknown, and so probabilistic language is appropriate. The best fit/maximum likelihood view extends the classic Kalman updates to a rich family of efficient generalized Kalman smooth algorithms for signals corrupted by outliers, nonlinear models, constraints through side information, and a myriad of other applications, see [40], [41], and [42]. In the simplest invocation, the Kalman estimator is determined given only the ratio of measurement noise to the process's underlying stochastic noise. Fixing both of these parameters allows Kalman methods to also identify the variance of the associated estimator. Furthermore, a line of research aims to identify parameters purely from data, including [43], [44], and [45]. Many such methods include their own hyperparameters and are not guaranteed to find a solution, but are an improvement on the indeterminate nature of direct maximum likelihood or a prior choice for the variance hyperparameter.

In adding Kalman smoothing to SINDy, we introduce a distinction between the measurement variables and the state variables of the dynamical system in equation 1. As such, the inputs to the problem become m time points of measurements of k variables ($Z \in \mathbb{R}^{k\times m}$) and a linear transform from the state to the measurement variables $H \in \mathbb{R}^{k\times n}$ describing how the process is measured.

Measurement error is assumed to be normally distributed with $HX - Z \sim \sigma_z \mathcal{N}(0, R)$ where the covariance matrix

$R \in \mathbb{R}^{k\times k}$. Measurement regimes where noise is autocorrelated or varies over time can be accomodated by flattening $HX - Z$ and describing $R \in \mathbb{R}^{nk\times nk}$.

As a simplifying assumption for brevity and applying to the experiments in this paper, we use $R = I$. Two parameters are required: σ_z, the measurement noise standard deviation, and σ_x, the process velocity standard deviation per unit time. As mentioned, if only point estimates of the state are required, and posterior variance is not, it suffices to use the ratio $\rho = (\sigma_z/\sigma_x)^2$.

Each process is assumed to have an independent, Brownian velocity. This leads to Kalman smoothing estimator:

$$\arg\min_{X,\dot{X}} \|HX - Z\|_{R^{-1}}{}^2 + \rho\|G[\dot{X}, X]\|_{Q^{-1}}{}^2. \tag{4}$$

Here, G is a linear transform to separate $[\dot{X}, X]$ into independent, mean-zero increments, and Q is the covariance of those increments. This is the full form of equation 2. A graphic displaying Kalman filtering is shown in Fig. 2. To illustrate the ideas, the figure presents step-by-step filtering updates; however, batch smoothing is used for the model discovery applications presented in the experiments.

We use the generalized cross validation (GCV) of [43] to choose ρ. This strategy chooses ρ in order to minimize the loss on a witheld set of data. While the algorithm described in that paper is not guaranteed to find a minimum, heuristic experience has shown that the longer the trajectory, the more likely their algorithm will succeed. The experiments in the next section show that this strategy works reasonably well.

The GCV approach witholds some measurement points in order to find the values of H, R, G, and Q that produce estimates $\hat{X}$ that fit withheld data most accurately. This powerful approach can apply to many systems, and from a basic view, merely applies prox-gradient descent on the withheld loss with respect to the system matrices. In our work, we presume to know the measurement parameters and most of the process parameters - after all, ‘‘position is the integral of velocity’’, implies structural constraints on G and Q. Since we specify H, G, R, and Q, up to a scaling parameter ρ (the ratio of measurement variance to process variance over unit time) in equation 4, we call the GCV method with prox function equal to a projection onto the known structure of these matrices.

Alternate methods to assimilate noisy data, as mentioned earlier, include Total Variation regularization, Savitzky-Golay smoothing([20]), and Weak SINDy ([29]. Total variation regularization, emerging from denoising overhead imagery in [47]. Slight variations in the optimization method exist, but the implementation used in this paper's experiments seeks to minimize:

$$\lambda\|\dot{x}\|_1 + \|\int \dot{x} - z\|_2^2$$

for some measurements z and smoothing hyperparameter λ. Similar to Kalman smoothing, there is a regularity condition on the derivative and a hyperparameter that balances it with the measurement error term. However, in this TV implementation, the state is always the direct integral of $\dot{x}$, whereas Kalman allows for some covariance between state and its derivatives. On the other hand, Savitzky-Golay smoothing seeks to find a local polynomial regression of some window. As a local method, the effects of outliers are ignored beyond the window width, and it can handle nonstationary problems better. However, unlike Total Variation and Kalman Smoothing, Savitzky-Golay's smoothing hyperparameter (window width) affects runtime. Finally, Weak SINDy eschews the need to estimate derivatives by building a regression from the weak form of the ODE:

$$\int \dot{X}\phi(t)dt = \int \Xi\Theta(X)\phi(t)dt$$
$$-\int X\phi_t(t)dt = \int \Xi\Theta(X)\phi(t)dt$$

for some suitable test function $\phi(t)$ that perishes at the integral boundary. By cutting up the data space into temporal bins (and in PDEs, spatiotemporal bins), the effect of measurement noise is averaged away. This is a compelling approach. It does not need two separate steps and may be better conditioned than Kalman smoothing, but it cannot take advantaged of a smoothed state in $\Theta(X)$. As the implementation in `pysindy` is incomplete, we do not include it in our experiments.

III. ENGINEERING

We sought to make the implementation of the work as much a part of the contribution as the mathematical and experimental work. The aim is to go beyond reproducibility and provide reusability not just of the method, but the experiments and their constituent functionality. This contribution is spread across four Python packages: `pysindy`, `derivative`, `mitosis`, and `pysindy-experiments`.

In `pysindy`, we enabled incorporating the smoothed coordinates into the second SINDy step, the ODE regression. It should be noted that previous experiments using `pysindy`'s derivative estimation would re-use the noisy coordinates in function library evaluation. This functionality is partially extended to smoothing methods that use the `derivative` package, but we only implemented it for Total Variation smoothing and Kalman. Finally, we added an entry point group so that others could write hyperparameter optimization plugins that `derivative` methods would pick up.

Additionally in `pysindy`, we redefined ensembling in a more flexible way to apply to a greater variety of underlying optimizers, such as the MIOSR one used in these experiments. The standardization of interfaces allows us to compose SINDy experiments from the following section in the `pysindy-experiments` package [48]. This package provides an API to specify data generation, model building, and evaluation. This package also registers a `derivative` plugin to conduct the Kalman GCV hyperparameter optimization of [43]. Given that their method optimizes a large

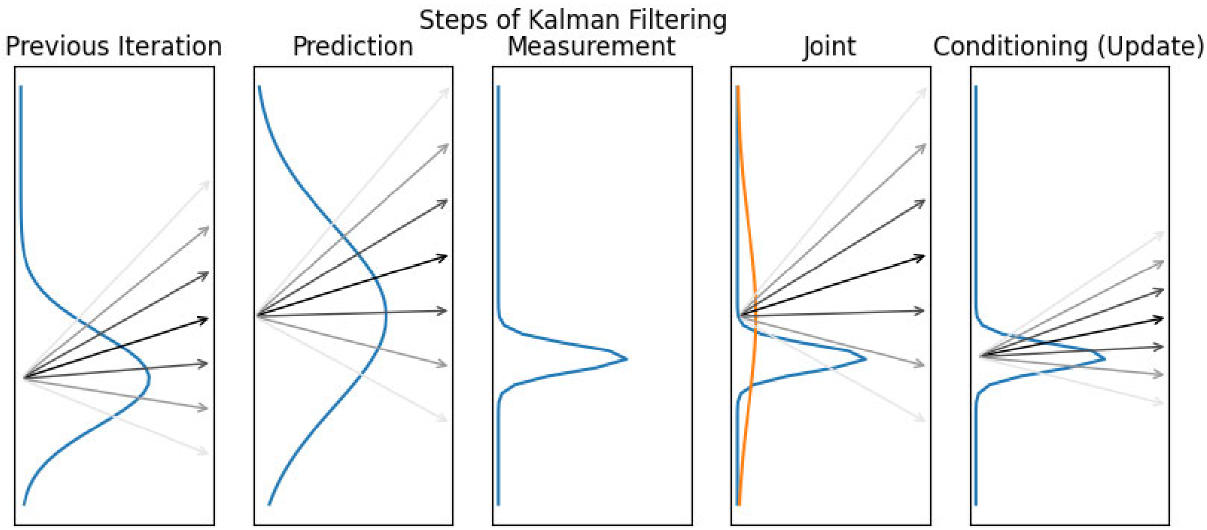


FIGURE 2. Explanatory depiction of Kalman filtering. A previous iteration gives a distribution $p(x_{i-1}, \dot{x}_{i-1})$. Multiplication by an update matrix produces the predictions $p(x_i, \dot{x}_i | x_{i-1}, \dot{x}_{i-1})$. Simultaneously, measurements z_i are taken that, with known measurement noise, give $p(z_i | x_i)$. Multiplication gives the joint distribution $p(x_i, \dot{x}_i, z_i | x_{i-1}, \dot{x}_{i-1})$, from which the conditional distribution $p(x_i, \dot{x}_i | z_i, x_{i-1}, \dot{x}_{i-1})$ can be calculated, shown in [46].

parameter space that we then restrict to a single parameter at each iteration, this setup is far from computationally optimal; a faster implementation is beyond the ambit of this paper.

Finally, in order to make it easier to collaborate and reproduce experiments, we built the `mitosis` package [49] for this and other projects. Rather than run experiments interactively, this package provides a way to enforce and log relevant runtime reproducibility information and specify experiment parameter variants in a declarative manner. The appendix contains instructions on how to use it to reproduce the experiments in this paper.

IV. EXPERIMENTS

A. SETUP

We seek to evaluate Kalman smoothing as a step in SINDy in comparison to other noise-mitigation innovations. We simulate eight dynamical systems[1] with noisy measurements across a variety of initial conditions, discovering ODEs from SINDy with different smoothing methods. These systems are of moderate complexity: two chaotic, one linear, three quadratic, four cubic, and all existing in `pysindy`'s repertoire. We aim to explore the behavior under varying noise regimes. An extension of the experiments to more systems in order to draw conclusions about the behavior across different dynamical characteristics represented in [50] would be a worthy extension, but beyond these first steps.The trials are run across a range of durations and relative noise levels, calculated as the noise-to-signal ratio of the measurement variance with the system's mean squared value. To compare the methods, we then integrate the discovered equations and observe how well they preserve the system's structure as well as directly comparing the coefficients.

We compare the results of SINDy with three different differentiation/smoothing methods, parametrized across a gridsearch: Kalman smoothing, Savitzky-Golay, and L-1 total variation minimization,[2] as well as with Kalman smoothing, with GCV hyperparameter optimization [43]. The gridsearch parameters sweep a range of reasonable values around the default provided to a user. We explore noise regimes ranging from minimal to severe, and data availability ranging from parsimonious to copious. It is worth noting that choosing the gridsearched optimum requires knowledge of the true system, in distinction to the hyperparameter optimization method used for Kalman smoothing. The GCV method itself requires hyperparameters only for the proportion of witheld data, stopping criteria, and any regularization.

Methods can be compared in several ways: by the coefficients of the equations they discover, by their accuracy in forecasting derivatives, and how well the discovered system recreates observed dynamics in simulation. There are many metrics for scoring dynamical system discovery, and the merit of a metric depends upon both the use case and whether the trajectory considered is one of importance. For instance, in controls engineering, the local derivative and very short-term forecasting is the primary imperative. On the other hand, for reduced-order PDE models, recreating larger-scale phenomena in simulation may be more important. Finally, in high-dimensional network dynamics, the accuracy of identifying connectivity, as measured by coefficient F1 score, is most important.

[1]Cubic Harmonic Oscillator, Duffing, Hopf, Lotka-Volterra, Rossler, Simple Harmonic Oscillator (SHO), Van Der Pol Oscillator, and Lorenz-63.

[2]TV requires a coefficient for the L-1 regularizer and Savitzky-Golay requires a smoothing window width.

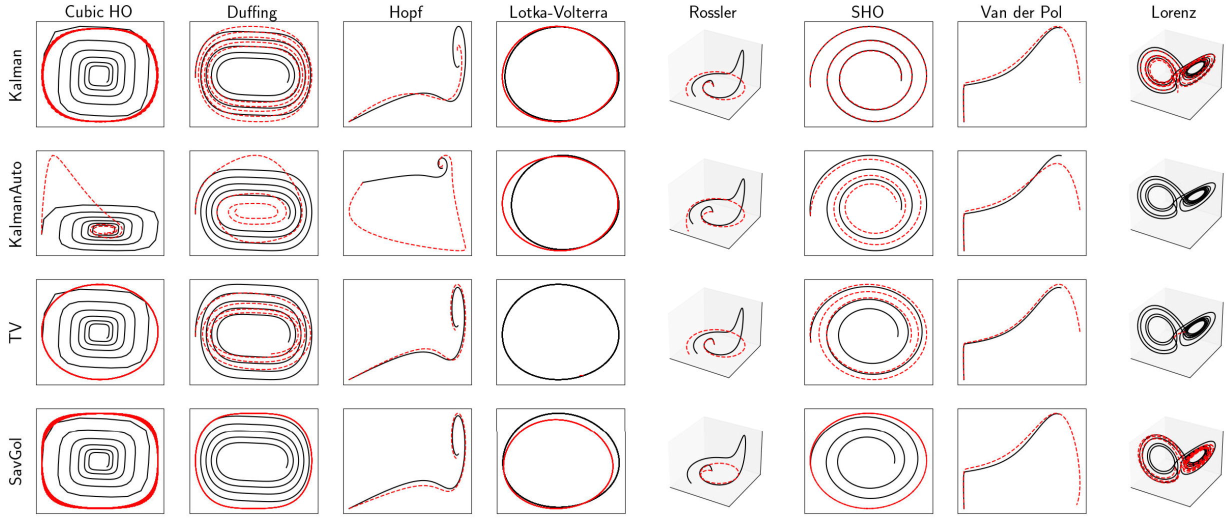


FIGURE 3. The simulation of discovered models compared to test data. Kalman appears better for half of eight ODEs. It represents the essential behavior of more ODEs than TV and Savitzky-Golay. Kalman with auto-hyperparameter selection performs similarly to total variation on a gridsearch. 10% relative noise, 8 seconds of data.

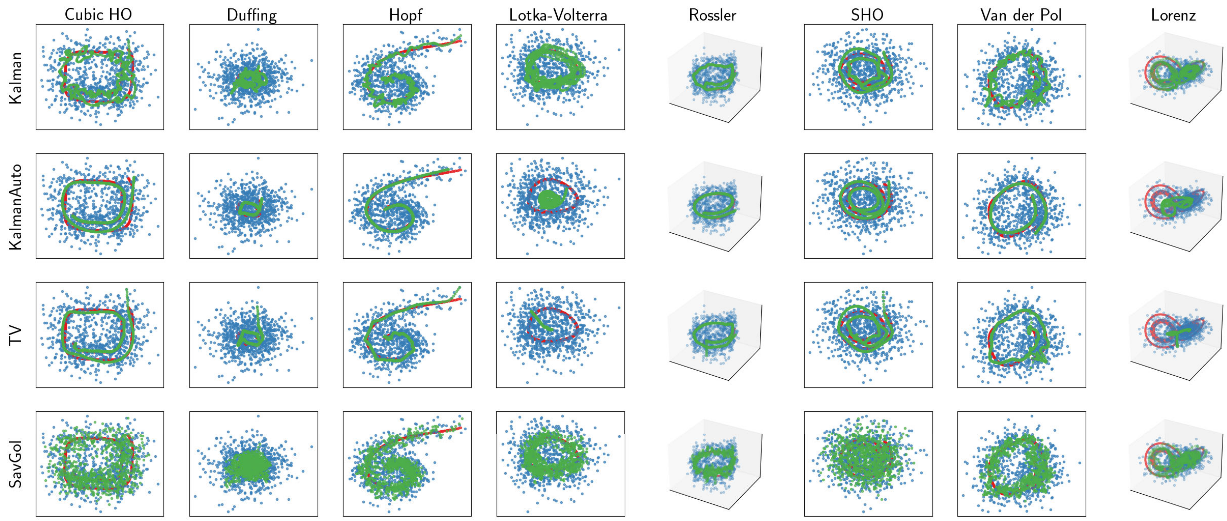


FIGURE 4. The smoothing of training data, performed by different differentiation methods prior to SINDy fit. It does not appear to be the case that a more visually accurate smoothing yields a model that behaves more correctly in simulation. Nevertheless, as Fig. 3 shows, Kalman-smoothed trajectories lead to better models in simulation. 10% relative noise, 8 seconds of data. Of interest, note the particular cases of SHO, cubic HO, and Hopf systems, in which the best smoother does not necessarily lead to the best simulation.

The integration metrics require additional parametrization and are best suited for low-noise systems. Coefficient metrics are the most straightforwards, and we compare methods by F1 score and MAE (mean absolute error) as the duration of training data increases, and separately, as the measurement noise increases. These metrics, based upon L1 and L0 norms, reflect the goal of sparsity better than metrics based upon the L2 norm. We also visually evaluate how well the discovered ODEs, simulated from random initial conditions in a test set, track the true data and display relevant behavior.

Beyond the differentiation step, the SINDy models also specify a function library and optimizer. The feature library used for all experiments was cubic polynomials, including mixed terms and a constant term. The optimizer was the mixed-integer SINDy optimizer of [22], configured with the correct number of nonzero terms a priori, and ensembled over 20 models each trained on 60% of the data. Presenting SINDy with the known number of nonzero coefficients is an attempt to present a best case, where we can ameliorate any interaction between the smoothing method and sparsification parameters. A full list of ODE, simulation, and experimental parameters are shown in the Appendix, tables 1 and 2.

B. RESULTS

We find that SINDy with Kalman smoothing recovers the problem structure in application as well or better than

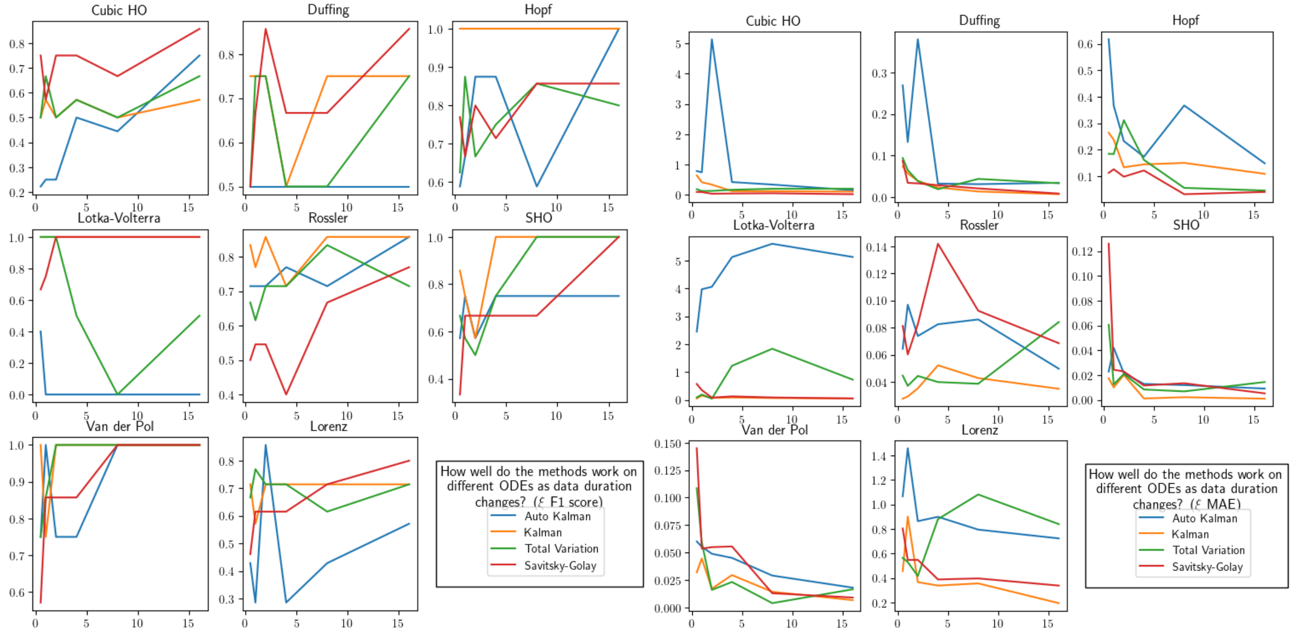


FIGURE 5. How well different smoothing methods in SINDy recover the ODE coefficients as data duration increases. 10% relative noise.

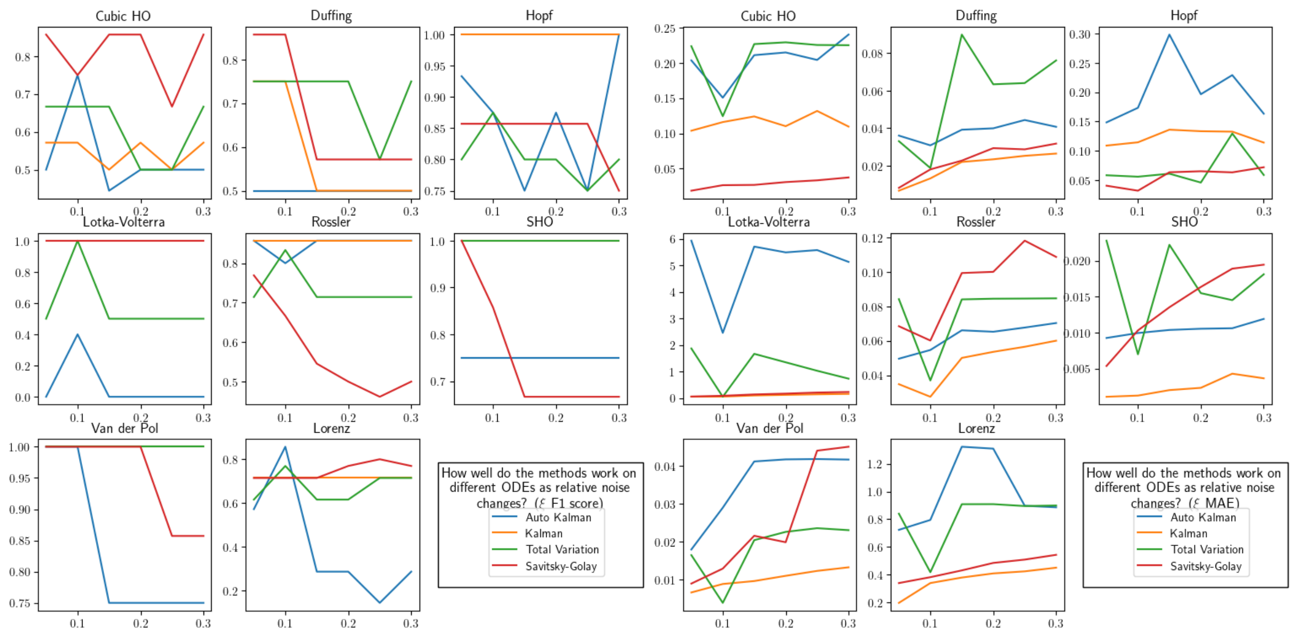


FIGURE 6. How well different smoothing methods in SINDy recover the ODE coefficients as noise increases. 8 seconds of data.

competing methods in seven of eight systems. Models discovered in this manner track the essential dynamics in most cases. GCV hyperparameter optimization is noticeably not as good as the ideal gridsearched optima, but still provides accurate smoothing in six of eight cases. Somewhat counterintuitively, however, the best smoothing parameter does not necessarily lead to the most accurate solution, either in coefficient accuracy or in simulation.

SINDy with Kalman hyperparameter optimization tends to perform worse than that with Savitzky-Golay, but on par with Total Variation gridsearched optima, and is itself outperformed only slightly by the Kalman gridsearched optima. While hyperparameter optimization imposes some runtime cost, it does not require access to the true data, making those results all the more inspiring for field use cases. Simulations of discovered

TABLE 1. The parametrization of ODEs used in these experiments. Mostly from defaults in the pysindy package.

System	ODE	Experiment parameters	x_0 mean
Linear Damped Oscillator	$\dot{x} = \begin{bmatrix} -\alpha & \beta \\ -\beta & -\alpha \end{bmatrix} x$	$\alpha = .1, \beta = 2$	(0,0)
Lorenz	$\dot{x} = \begin{bmatrix} \sigma(x_2 - x_1) \\ x_1(\rho - x_3) - x_2 \\ x_1 x_2 - \beta x_3 \end{bmatrix}$	$\sigma = 10, \rho = 28, \beta = 8/3$	(0, 0, 15)
Cubic Damped Oscillator	$\dot{x} = \begin{bmatrix} -\alpha & \beta \\ -\beta & -\alpha \end{bmatrix} \begin{bmatrix} x_1^3 \\ x_2^3 \end{bmatrix}$	$\alpha = .1, \beta = 2$	(0,0)
Duffing	$\dot{x} = \begin{bmatrix} x_2 \\ -\alpha x_2 - \beta x_1 - \gamma x_1^3 \end{bmatrix}$	$\alpha = .2, \beta = .05, \gamma = 1$	(0,0)
Hopf	$\dot{x} = \begin{bmatrix} -\alpha x_1 - \beta x_2 - \gamma x_1(x_1^2 + x_2^2) \\ \beta x_1 - \alpha x_2 - \gamma x_2(x_1^2 + x_2^2) \end{bmatrix}$	$\alpha = .05, \beta = 1, \gamma = 1$	(0,0)
Lotka-Volterra	$\dot{x} = \begin{bmatrix} \alpha x_1 - \beta x_1 x_2 \\ \beta x_1 x_2 - 2\alpha x_2 \end{bmatrix}$	$\alpha = 5, \beta = 1$	$(5, 5)$
Rossler	$\dot{x} = \begin{bmatrix} -x_2 - x_3 \\ x_1 + \alpha x_2 \\ \beta + (x_1 - \gamma) x_3 \end{bmatrix}$	$\alpha = .2, \beta = .2, \gamma = 5.7$	(0,0,0)
Van der Pol Oscillator	$\dot{x} = \begin{bmatrix} x_2 \\ \alpha(1 - x_1^2) x_2 - x_1 \end{bmatrix}$	$\alpha = .5$	(0,0)

models across all ODEs and methods are shown in Fig. 3

There are cases where the MAE scores of different methods do not indicate which method performs better in simulation, and where effective smoothing does not predict effective system recovery. As one case in point, Kalman GCV and Total Variation smoothing appears most accurate for the Hopf system in Fig. 4. However, the coefficient metrics show that Total Variation, Kalman gridsearch, and Savitzky-Golay recovered the system equations better that Kalman GCV, and simulation in Fig. 3 confirms Kalman GCV is the outlier in reconstructing the dynamics. Similarly, methods have a wide range of performance on MAE and F1 score on the Rossler system, despite all simulations missing the chaotic behavior. As a final case in point, the Kalman-smoothed training data itself does not seem as accurate as data smoothed by L-1 Total Variation in the the SHO trial. Yet a SINDy model based upon it performs better

Just as surprisingly, despite performing well in simulating the Lotka-Volterra system, SINDy with Kalman hyperparameter optimization performs poorly in the coefficient metrics. Coefficient metrics by data duration is shown in Fig. 5, and by noise level shown in Fig. 6. If anything appears consistent about Kalman with GCV, it is that, with a long enough duration, it becomes more consistent. Across the range of noise levels sampled, either Savitzky-Golay or Kalman (gridsearched) perform the best, depending upon system. As expected, Kalman (gridsearched) always outperforms Kalman GCV, but it is interesting to note that at some durations and noise levels Kalman GCV occasionally outperforms Savitzky-Golay (e.g. Rossler, SHO).

V. CONCLUSION

This paper has demonstrated that Kalman smoothing is a useful addition to SINDy. It makes the method more generally applicable across domains. The Kalman smoother behaves optimally for the simplest systems and provides a familiar process to the controls engineering community. It also appears to perform better at preserving global system structure in simulation. Incorporating the GCV hyperparameter optimization of [43] may not recover the best model, but it allows one to at least recover useful models without relying on an accurate parametrization a priori, particularly if substantial training data exists. Meanwhile, Savitzky-Golay, as a local method, struggles to discover the slow decay of some oscillators, whereas TV makes errors in systems that do not have sufficient corners for the L-1 regularization to be effective.

Plenty of ideas in extant literature could be leveraged to improve the approach in future work. In one direction, the GCV approach could extend to the second step of SINDy. Prior art in this direction, the Stepwise Sparse Regressor of [3], conducts cross validation on the sparsity of the solution, but has not fully been incorporated into `pysindy`. Completing the implementation, and combining it with Kalman GCV in the first step of SINDy, would provide the first hyperparameter-free approach to SINDy. In an other direction, it would be elucidating to contrast the hyper-parameter estimation techniques of [44] for Savitzky-Golay with those of [43] in the smoothing step of SINDy.

Since Kalman smoothing and SINDy regression loss terms both accommodate variable timesteps, a natural innovation

TABLE 2. Parametrization of data, SINDy models, and experiments conducted.
***Lotka-Volterra uses a gamma distribution, rather than normal, in order to enforce nonnegativity.**

Parameter	Value
Simulated Data	
Number of trajectories	10
Initial Condition (x_0) variance	9
Initial Condition (x_0) distribution	Normal*
Measurement error elative noise (default)	10%
Trajectory duration (default)	16
Measurement interval	0.01
Random seed	19
SINDy model	
Feature Library (Θ)	Polynomials to degree 3
Optimizer	Mixed Integer Optimizer
L2 regularization (coefficitents) (α)	0.01
Target sparsity	(true value from equation)
Unbiasing	Yes
Feature normalization	No
Ensembling	data bagging
Number of bags	20
Experiment	
Trajectory duration (grid)	0.5, 1, 2, 4, 8, 16
Relative noise (grid)	0.05, 0.1, 0.15, 0.2, 0.25, 0.3
Measurement:Process variance (Kalman grid)	1e-4, 1e-3, 1e-2, 1e-1, 1
L1 regularization (derivative) (TV grid)	1e-4, 1e-3, 1e-2, 1e-1, 1
Window length (Savitzky-Golay grid)	5, 8, 12, 15
GCV witheld rate	every fourth observation
GCV Tikhonov regularization	1e-1
All other GCV params default	

is to combine equation 3 and 2 into a single optimization problem. This would introduce the complexity of handling the nonlinear term, $\Xi\Theta(X)$. Bayesian SINDy [23], Langevin Regression [19], and Weak SINDy [28] each introduce alternate single-step methods, but do not evaluate their single step methods in comparison to the mathematically nearest two-step SINDy. As a result, it is difficult to evaluate that aspect of their innovations in isolation. A single-step SINDy that utilizes Kalman loss would allow a clearer picture of the trade-off between measurement noise and coefficient sparsity, as all of the same hyperparameters would be present in the two-step version. Moreover, the Gaussian process background of Kalman smoothing motivates a more detailed mathematical comparison with other stochastic ODE learning methods, including Kernel SINDy [34].

The biggest use case for the results in this work is building low-dimensional approximations of high-dimensional PDEs for faster simulation, e.g. for molecular bond angles [3]. Casual experimentation indicates that the nonconvexity of GCV is less of a problem with more data, and such a use case allows as much data as is required. Its reduction in hyperparameters also makes it amenable as a first effort in dealing with new, poorly understood systems. On the other hand, for uncovering connections between variables, such as in neural activity or chemical reaction networks [16], the performance on coefficient F1 metrics indicates that more accurate parameter tuning is essential. Finally, for engineering and control systems already built around Kalman filtering, using Kalman SINDy allows for recovery of meaningful parameters in that context.

APPENDIX

This paper is built from https://github.com/Jacob-Stevens-Haas/Kalman-SINDy-paper. To run the experiments, install the package located in images/gen_image and run the commands in images/gen_image/run_exps.sh. These are a series of mitosis commands to record reproducibility information and results for each trial and produce an html notebook of visualizations for that trial. Each experiment trial will generate a pseudorandom hex key for reproducibility. To build the final figures, edit images/gen_image/composite_plots.py with the keys to the experimental results and run it.

The experiments are made to be run across a range of compatible packages, defined in the pyproject.toml. To run against the exact package versions used in this paper, see the file images/published-requirements.txt. While the exact parametrization is in the experimental configuration and package defaults, it is recreated here in Tables 1 and 2.

REFERENCES

[1] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016.

[2] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2022.

[3] L. Boninsegna, F. Nüske, and C. Clementi, "Sparse learning of stochastic dynamical equations," *J. Chem. Phys.*, vol. 148, no. 24, p. 6, Jun. 2018.

[4] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Sci. Adv.*, vol. 3, no. 4, Apr. 2017, Art. no. e1602614. [Online]. Available: https://www.science.org

[5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[6] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng.*, vol. 83, no. 1, pp. 95–108, Mar. 1961.

[7] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, NC, USA, Tech. Rep. TR 95-041, 1995.

[8] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Philadelphia, PA, USA: SIAM, 2016.

[9] S. M. Ichinaga, F. Andreuzzi, N. Demo, M. Tezzele, K. Lapo, G. Rozza, S. L. Brunton, and J. N. Kutz, "PyDMD: A Python package for robust dynamic mode decomposition," 2024, *arXiv:2402.07463*.

[10] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.

[11] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," *J. Sci. Comput.*, vol. 92, no. 3, p. 88, Sep. 2022.

[12] J. S. North, C. K. Wikle, and E. M. Schliep, "A review of data-driven discovery for dynamic systems," *Int. Stat. Rev.*, vol. 91, no. 3, pp. 464–492, Dec. 2023.

[13] L. Rosafalco, P. Conti, A. Manzoni, S. Mariani, and A. Frangi, "EKF-SINDy: Empowering the extended Kalman filter with sparse identification of nonlinear dynamics," 2024, *arXiv:2404.07536*.

[14] J. Wang, J. Moreira, Y. Cao, and B. Gopaluni, "Time-variant digital twin modeling through the Kalman-generalized sparse identification of nonlinear dynamics," in *Proc. Amer. Control Conf. (ACC)*, Atlanta, GA, USA, Jun. 2022, pp. 5217–5222. [Online]. Available: https://ieeexplore.ieee.org/document/9867786/

[15] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz, ''A unified sparse optimization framework to learn parsimonious physics-informed models from data,'' *IEEE Access*, vol. 8, pp. 169259–169271, 2020.

[16] M. Hoffmann, C. Fröhner, and F. Noé, ''Reactive SINDy: Discovering governing reactions from concentration data,'' *J. Chem. Phys.*, vol. 150, no. 2, Jan. 2019, Art. no. 025101.

[17] Y. Guan, S. L. Brunton, and I. Novosselov, ''Sparse nonlinear models of chaotic electroconvection,'' *Roy. Soc. Open Sci.*, vol. 8, no. 8, Aug. 2021, Art. no. 202367.

[18] J. L. Callaham, J.-C. Loiseau, G. Rigas, and S. L. Brunton, ''Nonlinear stochastic modelling with Langevin regression,'' *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 477, no. 2250, Jun. 2021, Art. no. 20210092.

[19] S. H. Rudy, S. L. Brunton, and J. N. Kutz, ''Smoothing and parameter estimation by soft-adherence to governing equations,'' *J. Comput. Phys.*, vol. 398, Dec. 2019, Art. no. 108860. [Online]. Available: http://www.elsevier.com/locate/jcp

[20] R. Chartrand, ''Numerical differentiation of noisy, nonsmooth data,'' *ISRN Appl. Math.*, vol. 2011, pp. 1–11, May 2011.

[21] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, ''A unified framework for sparse relaxed regularized regression: SR3,'' *IEEE Access*, vol. 7, pp. 1404–1423, 2019.

[22] D. Bertsimas and W. Gurnee, ''Learning sparse nonlinear dynamics via mixed-integer optimization,'' *Nonlinear Dyn.*, vol. 111, no. 7, pp. 6585–6604, Apr. 2023.

[23] S. M. Hirsh, D. A. Barajas-Solano, and J. N. Kutz, ''Sparsifying priors for Bayesian uncertainty quantification in model discovery,'' *Roy. Soc. Open Sci.*, vol. 9, no. 2, Feb. 2022, Art. no. 211823, doi: 10.1098/rsos.211823.

[24] L. M. Gao and J. N. Kutz, ''Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants,'' 2022, *arXiv:2211.10575*.

[25] U. Fasel, J. N. Kutz, B. W. Brunton, and S. L. Brunton, ''Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control,'' *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 478, no. 2260, Apr. 2022, Art. no. 20210904.

[26] A. A. Kaptanoglu, J. L. Callaham, A. Aravkin, C. J. Hansen, and S. L. Brunton, ''Promoting global stability in data-driven models of quadratic nonlinear dynamics,'' *Phys. Rev. Fluids*, vol. 6, no. 9, p. 94401, Sep. 2021.

[27] M. Peng, A. Kaptanoglu, C. Hansen, J. Stevens-Haas, K. Manohar, and S. L. Brunton, ''Local stability guarantees for data-driven quadratically nonlinear models,'' 2024, *arXiv:2403.00324*.

[28] D. A. Messenger and D. M. Bortz, ''Weak SINDy for partial differential equations,'' *J. Comput. Phys.*, vol. 443, Oct. 2021, Art. no. 110525.

[29] D. A. Messenger and D. M. Bortz, ''Weak SINDy: Galerkin-based data-driven model selection,'' *Multiscale Model. Simul.*, vol. 19, no. 3, pp. 1474–1497, Jan. 2021, doi: 10.1137/20m1343166.

[30] H. Schaeffer and S. G. McCalla, ''Sparse model selection via integral terms,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 96, no. 2, p. 23302, Aug. 2017.

[31] L. M. Gao, U. Fasel, S. L. Brunton, and J. N. Kutz, ''Convergence of uncertainty estimates in ensemble and Bayesian sparse model discovery,'' 2023, *arXiv:2301.12649*.

[32] A. Kaptanoglu, B. de Silva, U. Fasel, K. Kaheman, A. Goldschmidt, J. Callaham, C. Delahunt, Z. Nicolaou, K. Champion, J.-C. Loiseau, J. Kutz, and S. Brunton, ''PySINDy: A comprehensive Python package for robust sparse system identification,'' *J. Open Source Softw.*, vol. 7, no. 69, p. 3994, Jan. 2022.

[33] Z. Long, Y. Lu, and B. Dong, ''PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network,'' *J. Comput. Phys.*, vol. 399, Dec. 2019, Art. no. 108925.

[34] D. Long, N. Mrvaljević, S. Zhe, and B. Hosseini, ''A kernel framework for learning differential equations and their solution operators,'' *Phys. D, Nonlinear Phenomena*, vol. 460, Apr. 2024, Art. no. 134095.

[35] S. Atkinson, ''Bayesian hidden physics models: Uncertainty quantification for discovery of nonlinear partial differential operators from data,'' 2020, *arXiv:2006.04228*.

[36] M. D. Cranmer, R. Xu, P. Battaglia, and S. Ho, ''Learning symbolic physics with graph networks,'' in *Proc. 33rd Annu. Conf. Neural Inf. Process. Syst.*, 2019, Paper 05862.

[37] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, ''Discovering symbolic models from deep learning with inductive biases,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, Paper 11287.

[38] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, ''Learning to simulate complex physics with graph networks,'' in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 8459–8468.

[39] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura, ''Industrial applications of the Kalman filter: A review,'' *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5458–5471, Dec. 2013.

[40] A. Aravkin, J. V. Burke, L. Ljung, A. Lozano, and G. Pillonetto, ''Generalized Kalman smoothing: Modeling and algorithms,'' *Automatica*, vol. 86, pp. 63–86, Dec. 2017.

[41] A. Aravkin, J. V. Burke, and G. Pillonetto, ''Robust and trend-following Kalman smoothers using student's t,'' *IFAC Proc. Volumes*, vol. 45, no. 16, pp. 1215–1220, Jul. 2012.

[42] J. Jonker, A. Aravkin, J. V. Burke, G. Pillonetto, and S. Webster, ''Fast robust methods for singular state-space models,'' *Automatica*, vol. 105, pp. 399–405, Jul. 2019.

[43] S. T. Barratt and S. P. Boyd, ''Fitting a Kalman smoother to data,'' in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 1526–1531.

[44] F. Van Breugel, J. N. Kutz, and B. W. Brunton, ''Numerical differentiation of noisy data: A unifying multi-objective optimization framework,'' *IEEE Access*, vol. 8, pp. 196865–196877, 2020.

[45] J. Jonker, P. Zheng, and A. Y. Aravkin, ''Efficient robust parameter identification in generalized Kalman smoothing models,'' *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4852–4857, Oct. 2021.

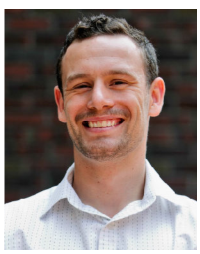
[46] M. L. Eaton, *Multivariate Statistics: A Vector Space Approach*, vol. 53. Institute of Mathematical Statistics, 2007.

[47] L. I. Rudin, S. Osher, and E. Fatemi, ''Nonlinear total variation based noise removal algorithms,'' *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, Nov. 1992.

[48] J. Stevens-Haas and Y. Bhangale. (Apr. 2024). *Pysindy-Experiments*. [Online]. Available: https://github.com/Jacob-Stevens-Haas/gen-experiments

[49] J. Stevens-Haas. (Apr. 2024). *Mitosis*. [Online]. Available: https://github.com/Jacob-Stevens-Haas/mitosis

[50] W. Gilpin. (2023). *Chaos as an Interpretable Benchmark for Forecasting and Data-Driven Modelling*. [Online]. Available: https://github.com/williamgilpin/dysts



JACOB M. STEVENS-HAAS received the B.S. degree in quantitative economics from United States Naval Academy, in 2011. He is currently pursuing the Ph.D. degree in applied mathematics with the University of Washington.

From 2011 to 2017, he was an Auxiliaries Officer on the USS Momsen (DDG-92) and a Weapons and Tactics Instructor for Destroyer Squadron NINE. He is the Lead Maintainer of `pysindy`, the python package for sparse identification of nonlinear dynamics, as well as various patches to various data science Python packages.

YASH BHANGALE was born in Jalgaon, Maharashtra, India, in 1999. He received the B.Tech. degree in mechanical engineering from MIT World Peace University, Pune, Maharashtra, in 2021. He is currently pursuing the M.S. degree in mechanical engineering with the University of Washington, Seattle, WA, USA. His research interests include dynamical systems, machine learning, system identification, and differential equations.

J. NATHAN KUTZ (Senior Member, IEEE) received the B.S. degree in physics and mathematics from the University of Washington, Seattle, WA, USA, in 1990, and the Ph.D. degree in applied mathematics from Northwestern University, Evanston, IL, USA, in 1994. He is currently the Director of the AI Institute in Dynamics Systems, a Professor of applied mathematics and electrical engineering, and a Senior Data Science Fellow with the eScience Institute, University of Washington.

ALEKSANDR ARAVKIN received the B.S. degree in mathematics and computer science, the M.S. degree in statistics, and the Ph.D. degree in mathematics from the University of Washington (UW), Seattle, WA, USA, in 2004 and 2010, respectively. In 2015, he rejoined UW, where he is currently an Associate Professor of applied mathematics; an Adjunct Associate Professor in statistics, computer science, mathematics, and health metrics sciences; a Data Science Fellow at the eScience Institute; and the Director of Mathematical Sciences at the Institute for Health Metrics and Evaluation.