Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 375 (2021) 113620

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Sparse identification of nonlinear dynamical systems via reweighted ℓ_1-regularized least squares

Alexandre Cortiella, Kwang-Chun Park, Alireza Doostan*

Smead Aerospace Engineering Sciences Department, University of Colorado, Boulder, CO 80309, USA

Available online xxxx

Abstract

This work proposes an iterative sparse-regularized regression method to recover governing equations of nonlinear dynamical systems from noisy state measurements. The method is inspired by the Sparse Identification of Nonlinear Dynamics (SINDy) approach of Brunton et al. (2016), which relies on two main assumptions: the state variables are known *a priori* and the governing equations lend themselves to sparse, linear expansions in a (nonlinear) basis of the state variables. The aim of this work is to improve the accuracy and robustness of SINDy in the presence of state measurement noise. To this end, a reweighted ℓ_1-regularized least squares solver is developed, wherein the regularization parameter is selected from the corner point of a Pareto curve. The idea behind using weighted ℓ_1-norm for regularization – instead of the standard ℓ_1-norm – is to better promote sparsity in the recovery of the governing equations and, in turn, mitigate the effect of noise in the state variables. We also present a method to recover single physical constraints from state measurements. Through several examples of well-known nonlinear dynamical systems, we demonstrate empirically the accuracy and robustness of the reweighted ℓ_1-regularized least squares strategy with respect to state measurement noise, thus illustrating its viability for a wide range of potential applications.

Keywords: Nonlinear system identification; Sparse regression; Basis pursuit denoising (BPDN); Reweighted ℓ_1-regularization; Pareto curve; SINDy

1. Introduction

Several branches in science and engineering represent systems that change over time as a set of differential equations that govern the underlying physical behavior. The structure of these differential equations is usually determined by observing the system and inferring relationships between variables, or derived from fundamental axioms and mathematical reasoning. Examples of the empirical method include Johannes Kepler and Isaac Newton's approaches in deriving laws of planetary motion. The accurate measurements of planet trajectories by Tycho Brahe enabled Kepler to empirically determine the laws that govern the motion of elliptic orbits. Newton, in turn, was able to derive the law of universal gravitation by inductive reasoning. Solving models derived from fundamental laws, either analytically or numerically, has proven to be a useful approach in engineering to produce reliable systems. However, the derived models often rely on simplifying assumptions that may not explain complex phenomena,

* Corresponding author.
E-mail addresses: alexandre.cortiella@colorado.edu (A. Cortiella), kcpark@colorado.edu (K.-C. Park), alireza.doostan@colorado.edu (A. Doostan).

https://doi.org/10.1016/j.cma.2020.113620

leading to a mismatch between predictions and observations. Moreover, parameters of these models may need to be estimated indirectly from system observables.

Recent advances in data acquisition systems along with modern data science techniques have fostered the development of accurate data-driven approaches, such as inverse modeling and system identification, in modeling physical or biological systems [1]. In particular, system identification, i.e., deducting accurate mathematical models from measured observations, has experienced significant advances in recent years [2,3]. System identification is a key to improve understanding of complex phenomena, dominant feature analysis, design of experiments, and system monitoring and control. Promising applications include space weather modeling [4,5], development of material constitutive laws [6], turbulence modeling [7], and model predictive control [8], to name a few.

In dynamical systems where the underlying physics is not fully understood or simplified for the interest of computational efficiency, system identification may be used to both determine a model structure, e.g., the governing equations, and estimate model parameters from data. When the model structure is unknown, one particular approach that has received increasing attention is to approximate the nonlinear dynamics in an overcomplete basis of the state variables and eliminate the expansion terms that do not contribute to the dynamics. Examples of this approach include polynomial NARMAX [9], symbolic polynomial regression [10,11], and sparse (polynomial) regression [12,13] dubbed SINDy in [13].

System identification via sparse (polynomial) regression employs techniques from compressed sensing – specifically regularization via sparsity promoting norms, such as ℓ_0- and ℓ_1-norms, to identify an *a priori* unknown subset of the basis describing the dynamics. The idea behind sparsity-promoting norms is based on Occam's razor principle, also known as the law of parsimony, which states: "Of two competing theories, the simpler explanation of an entity is preferred" [14]. The identified model may then be further analyzed to understand the physical behavior of the dynamical system, and can be integrated in time to predict future state variables of the system. Least Absolute Shrinkage and Selection Operator (LASSO) [15], Least Angle Regression (LARS) [16], Sequentially Thresholded Least Squares (STLS) [13], and Basis Pursuit Denoising (BPDN) [17] are some sparsity-promoting algorithms that may be used for model recovery. In particular, the work by Wang et al. [12] proposed a compressive sensing approach to polynomial basis expansion with ℓ_1-minimization of the coefficients in order to recover nonlinear dynamical systems. Similarly, Brunton et al. [13] proposed SINDy to recover the governing equations of dynamical systems. In their approach, an overdetermined system of equations is solved using a Sequentially Thresholded Least Squares scheme – instead of ℓ_1-minimization – to enforce sparsity in the expansion coefficients and mitigate the impact of state measurement noise.

A major challenge in model recovery via SINDy is the identification of accurate models when the input data, i.e., state measurements or their time derivatives, are corrupted with noise. Noisy measurements may lead to the identification of incorrect basis terms and inaccurate estimation of the parameters of the model. While algorithms based on standard ℓ_1-regularization or thresholded least squares lead to accurate recovery of the governing equations for small noise levels, they may produce inaccurate results or wrong sparsity patterns as the noise in the data increases [18,19]. An exception is a class of chaotic dynamical systems, e.g., Lorenz 63, satisfying certain ergodicity property [20]. In these cases, as shown in [20], the governing equation can be recovered exactly even when the state data is highly noisy and a large fraction of data is corrupted with outliers.

1.1. Contribution of this work

The focus of this article is to improve the accuracy and robustness of SINDy in the presence of state measurement noise. We advocate the utility of a weighted ℓ_1-norm to regularize SINDy's regression problem. In the fields of sparse regression and compressed sensing, weighted ℓ_1-minimization has been shown to empirically outperform ℓ_1-minimization in recovering sparse solutions to certain underdetermined linear systems [21,22]. Similar observations have been made in approximating multivariate functions in orthogonal polynomial bases [23–25]. The weights can be generated iteratively and inversely proportional to the values of the coefficients from the previous iteration [22,23], or based on an approximate value of the coefficients [24] or the ℓ_∞-norm of the basis functions [25]. In this work, we adopt a reweighted version of the Basis Pursuit Denoising algorithm (WBPDN) to solve an overdetermined system with the aim of mitigating the effects of noise, thereby recovering the governing equations of dynamical systems more accurately. In WBPDN, the sparsity of the recovered model and its accuracy in generating the training data, specifically, the time derivatives of the state variables are balanced via a regularization parameter. We propose the selection of the regularization parameter based on the corner point of a Pareto curve. Additionally, we present a method based on low-rank matrix factorization via interpolative decomposition [26] to identify single constraints of dynamical systems, such as conservation of total energy, from state measurements that are polluted with low noise levels. Beyond learning constraints, the identification and removal of redundant information introduced by the constraints improve the conditioning of SINDy's regression problem.

A related approach to improving the accuracy and robustness of SINDy is the utility of data associated with short trajectories of the state variables [27,28]. The trajectories correspond to multiple (random) initializations of the system. In the present study, however, we assume we have access to only single, but long, trajectories of the system state variables, an assumption that is more relevant when state measurements are obtained experimentally.

We begin, in the next section, by presenting a background on recovering dynamical system equations from state measurements using sparsity-promoting regression techniques. In Section 3, we present the WBPDN approach along with discussions on the selection of the regularization parameter and the calculation of time derivatives of state variables from their noisy measurements. In Section 4, we introduce our approach for the identification of single constraints from state data. In Section 5, the performance of the WBPDN method is assessed through several numerical examples of well-known dynamical systems with distinct features. Finally, in Section 6, we draw conclusions and discuss relevant aspects of the proposed method, and provide directions for future improvement.

2. Problem statement and background

Throughout this work, we assume that a dynamical system has the form

$$\dot{\mathbf{x}}(t) = \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{2.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ are the known and measurable state variables of the system at time $t \in [0, T]$ and $\mathbf{f}(\mathbf{x}(t))\colon \mathbb{R}^n \to \mathbb{R}^n$ is a state-dependent unknown vector that describes the motion of the system. An important observation is that in many systems $\mathbf{f}(\mathbf{x}) := \mathbf{f}(\mathbf{x}(t))$ is a *simple* function of the state variables $\mathbf{x} := \mathbf{x}(t)$ in that only a small set of state-dependent quantities, e.g., damping or inertial forces, contribute to the dynamics. Given that $\mathbf{f}(\mathbf{x})$ is unknown and following [12,13], we assume that each state dynamics $\dot{x}_j := \dot{x}_j(t)$ or, equivalently, $f_j(\mathbf{x})$, $j = 1, \dots, n$, is spanned by a set of p candidate nonlinear (in the state variables) basis functions $\phi_i(\mathbf{x})$ weighted by unknown coefficients ξ_{ji},

$$\dot{x}_j = \sum_{i=1}^{p} \xi_{ji}\phi_i(\mathbf{x}), \quad j = 1, \dots, n. \tag{2.2}$$

As the true dynamics may be described by only a subset of the considered basis $\{\phi_i(\mathbf{x})\}$, the unknown coefficients ξ_{ij} are sparse. Exploiting this sparsity in identifying $\mathbf{f}(\mathbf{x})$ is the key idea behind SINDy algorithms. The selection of the basis is crucial as the true $\dot{\mathbf{x}}_j$, while unknown, are assumed to be either exactly or approximately in the span of the basis. For arbitrary dynamical systems, such a selection is non-trivial and physical insight or a trial and error approach must be exercised. One approach to ease the difficulty of basis selection is to build an overcomplete basis, perhaps by concatenating different types of basis, e.g., polynomials and harmonics. However, this may negatively affect the accuracy of the computed ξ_{ij}, especially in the presence of large levels of state measurement noise.

To determine the governing equations $\mathbf{f}(\mathbf{x})$ via (2.2), we assume that the state variables $\mathbf{x}$ are known and can be measured at discrete times t_k, $k = 1, \dots, m$, where m is the number of measurements. Hence, (2.2) may be written in matrix form as

$$\dot{\mathbf{x}}_j = \mathbf{\Phi}(\mathbf{x})\boldsymbol{\xi}_j, \quad j = 1, \dots, n, \tag{2.3}$$

where,

$$\dot{\mathbf{x}}_j = \begin{bmatrix} \dot{x}_j(t_1), & \dot{x}_j(t_2), & \dots, & \dot{x}_j(t_m) \end{bmatrix}^T \in \mathbb{R}^m;$$

$$\mathbf{\Phi}(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}(t_1)) & \phi_2(\mathbf{x}(t_1)) & \dots & \phi_p(\mathbf{x}(t_1)) \\ \phi_1(\mathbf{x}(t_2)) & \phi_2(\mathbf{x}(t_2)) & \dots & \phi_p(\mathbf{x}(t_2)) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}(t_m)) & \phi_2(\mathbf{x}(t_m)) & \dots & \phi_p(\mathbf{x}(t_m)) \end{bmatrix} \in \mathbb{R}^{m \times p}; \quad \text{and}$$

$$\boldsymbol{\xi}_j = \begin{bmatrix} \xi_{1j}, & \xi_{2j}, & \dots, & \xi_{pj} \end{bmatrix}^T \in \mathbb{R}^p.$$

Hereafter, we refer to $\mathbf{\Phi}(\mathbf{x})$ as the *measurement* matrix. As we shall describe in Section 3.3, we estimate the dynamics $\dot{\mathbf{x}}_j$ in (2.3) via time derivatives of the state variables x_j, which may require access to a large number of state measurements. Therefore, the present work focuses on overdetermined systems (2.3), where the number of measurements is larger than the number of candidate functions, i.e., $m > p$. This assumption may be relaxed when $\dot{\mathbf{x}}_j$ is directly measured. While outside the scope of the present work, *identifiability* analysis tools may be used to determine the suitability of a given time span over which state measurements are collected. For a discussion on identifiability of nonlinear ODE systems, we refer the interested reader to the work in [29].

Notation. *For the interest of a simpler notation, we henceforth drop the subscript* j *from* $\dot{\mathbf{x}}_j$ *and* $\boldsymbol{\xi}_j$ *in* (2.3). *Unless otherwise stated,* $\dot{\mathbf{x}}$ *refers to the measurements of* $\dot{x}_j$ *and not the dynamics* $\dot{\mathbf{x}}$ *in* (2.1).

The coefficients $\boldsymbol{\xi}$ are computed from (2.3), for each j, subject to a sparsity constraint on $\boldsymbol{\xi}$. Wang et al. [12] achieve this via ℓ_1-minimization or basis pursuit denoising (BPDN), a widely used compressed sensing technique,

$$\min_{\boldsymbol{\xi}} \|\boldsymbol{\xi}\|_1 \quad \text{subject to} \quad \|\mathbf{\Phi}(\mathbf{x})\boldsymbol{\xi} - \dot{\mathbf{x}}\|_2 \leq \delta, \tag{2.4}$$

where $\|\boldsymbol{\xi}\|_1 = \sum_{i=1}^{p} |\xi_i|$ is the ℓ_1-norm of $\boldsymbol{\xi}$ and $\delta \geq 0$ is some tolerance parameter to avoid overfitting. The unconstrained formulation of (2.4) is given by the second-order cone program,

$$\text{(BPDN)} \qquad \min_{\boldsymbol{\xi}} \|\mathbf{\Phi}(\mathbf{x})\boldsymbol{\xi} - \dot{\mathbf{x}}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_1, \tag{2.5}$$

which coincides with the unconstrained LASSO [15], and is also referred to as ℓ_1-regularized least squares [30]. In (2.5), the regularization parameter $\lambda > 0$ creates a trade-off between the accuracy of satisfying (2.3) and the sparsity of the solution.

The SINDy algorithm of [13] proposes a Sequentially Thresholded Least Squares (STLS) algorithm, which iteratively solves a least squares regression problem and hard-thresholds the coefficients to promote sparsity and thereby regularize the regression problem. The procedure is repeated on the non-zero entries of $\boldsymbol{\xi}$ until the solution converges or the algorithm reaches a maximum number of iterations. In more details, let $\mathcal{S}(\boldsymbol{\xi}) := \{i : \xi_i \neq 0\}$ denote the support of an instance of $\boldsymbol{\xi}$. At the $(k+1)$th iteration of STLS, $\boldsymbol{\xi}^{(k+1)}$ is computed from a least squares problem over $\mathcal{S}(\boldsymbol{\xi}^{(k)})$ and its components smaller than some threshold parameter $\gamma > 0$ are set to zero,

$$\begin{aligned} \text{(STLS)} \qquad \boldsymbol{\xi}^{(k+1)} &\longleftarrow \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \left\{ \|\mathbf{\Phi}(\mathbf{x})\boldsymbol{\xi} - \dot{\mathbf{x}}\|_2^2 \quad \text{subject to} \quad \mathcal{S}(\boldsymbol{\xi}) = \mathcal{S}(\boldsymbol{\xi}^{(k)}) \right\} \\ \boldsymbol{\xi}^{(k+1)} &\longleftarrow \mathcal{T}(\boldsymbol{\xi}^{(k+1)}; \gamma), \end{aligned} \tag{2.6}$$

where the thresholding operator $\mathcal{T}(\cdot; \gamma)$ is defined as

$$\mathcal{T}_i(\boldsymbol{\xi}; \gamma) = \begin{cases} \xi_i & \text{if} \quad |\xi_i| > \gamma \\ 0 & \text{otherwise,} \end{cases} \qquad i = 1, \ldots, p. \tag{2.7}$$

The choice of the threshold parameter γ remains a challenge since the magnitude of each entry of $\boldsymbol{\xi}$ is unknown and depends on the selected candidate basis. Mangan et al. [31] suggest the Akaike Information Criteria (AIC) for selecting γ; however, their approach is computationally expensive since many validation sets are needed to produce accurate results, and they use exact state derivatives instead of computing them from noisy state variables. Later, Rudy et al. [32] and Quade et al. [33] proposed a Sequential Thresholded Ridge Regression (STRidge) to solve for $\boldsymbol{\xi}$. STRidge solves a Ridge regression problem – a Tikhonov regularized and an improved variant of least squares regression – and thresholds the coefficients with magnitude smaller than γ. The aim of STRidge it thus to improve the condition number of the linear system arising from the least squares problem (2.6) in STLS. The threshold parameter is chosen – from several candidates – based on cross-validation or a Pareto curve [13].

In practice, BPDN, STLS, and STRidge lead to accurate recovery of $\boldsymbol{\xi}$ for small state measurement noises. However, as we shall illustrate in the examples of Section 5, the accuracy of the recovered coefficients for general dynamical systems may deteriorate considerably when the state variables and their time derivatives are polluted with relatively large noise levels. One reason for this lack of robustness to noise is that the measurement matrix $\mathbf{\Phi}(\mathbf{x})$ does not in general satisfy the incoherence [34–36] or restricted isometry property [37–39] conditions in the undersampled case, $m < p$, or the incoherence property [40,41] in the oversampled case, $m \geq p$. The reason for this is two fold. Firstly, $\mathbf{f}(\mathbf{x})$ may be sparse in a basis $\{\phi_i(\mathbf{x})\}$ that is not orthonormal, e.g., monomials. Secondly, unlike in standard compressed sensing or least squares regression settings, $\phi_i(\mathbf{x})$ are sampled at the state variables

$\mathbf{x}$, which follow the dynamics of the system, as opposed to random (or experimentally designed) samples that would lead to well-conditioned measurement matrices; see, e.g. [42].

Remark. In practice, only noisy measurements of state variables $\mathbf{x}$ are available and not their true values. For the interest of a simpler notation, we refer to both the true and measured values of state variables as $\mathbf{x}$, unless specified otherwise. Similarly, $\dot{\mathbf{x}}$ denotes both exact and approximate values of time derivatives of states. We, however, note that, in the presence of noise, only a *perturbed* variant of (2.3) is accessible.

3. Approach: Reweighted ℓ_1-regularized least squares

To improve the robustness of SINDy with respect to the state and state derivative noise, we propose regularizing the regression problem involving (2.3) via weighted ℓ_1-norm of $\boldsymbol{\xi}$,

$$\|\mathbf{W}\boldsymbol{\xi}\|_1 = \sum_{i=1}^{p} w_i|\xi_i|. \tag{3.1}$$

Here, $\mathbf{W} \in \mathbb{R}^{p\times p}$ is a diagonal matrix with diagonal entries $w_i > 0$, $i = 1, \dots, p$. Our approach is inspired by the work in [21–25] from the statistics, compressed sensing, and function approximation literature, where the weighted ℓ_1-norm has been shown to outperform the standard ℓ_1-norm in promoting sparsity, especially in the case of noisy measurements or when the solution of interest is not truly sparse, i.e., many entries of $\boldsymbol{\xi}$ are near zero [22–25]. Depending on the choice of $\mathbf{W}$, $\|\mathbf{W}\boldsymbol{\xi}\|_1$ gives a closer approximation to the ℓ_0-norm of $\boldsymbol{\xi}$, $\|\boldsymbol{\xi}\|_0 := \#\{i : \xi_i \neq 0\}$, than $\|\boldsymbol{\xi}\|_1$, and thus better enforces sparsity in $\boldsymbol{\xi}$.

More specifically, we solve the weighted variant of the BPDN problem (2.5),

$$\text{(WBPDN)} \qquad \min_{\boldsymbol{\xi}} \|\boldsymbol{\Phi}(\mathbf{x})\boldsymbol{\xi} - \dot{\mathbf{x}}\|_2^2 + \lambda\|\mathbf{W}\boldsymbol{\xi}\|_1, \tag{3.2}$$

which coincides with the adaptive LASSO approach of [21]. The problem in (3.2) may be solved via BPDN solvers for standard ℓ_1-minimization with the simple transformations $\tilde{\boldsymbol{\xi}} := \mathbf{W}\boldsymbol{\xi}$ and $\tilde{\boldsymbol{\Phi}}(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{x})\mathbf{W}^{-1}$, i.e.,

$$\min_{\tilde{\boldsymbol{\xi}}} \|\tilde{\boldsymbol{\Phi}}(\mathbf{x})\boldsymbol{\xi} - \dot{\mathbf{x}}\|_2^2 + \lambda\|\tilde{\boldsymbol{\xi}}\|_1. \tag{3.3}$$

Given the solution $\tilde{\boldsymbol{\xi}}$ to (3.3), $\boldsymbol{\xi}$ is then computed from $\boldsymbol{\xi} = \mathbf{W}^{-1}\tilde{\boldsymbol{\xi}}$. In what follows, we describe the selection of the weight matrix $\mathbf{W}$ and regularization parameter λ.

3.1. Setting weights W

The main goal of using a weighted ℓ_1-norm – instead of its standard counterpart – is to place a stronger penalty on the coefficients ξ_i that are anticipated to be small (or zero). The obvious choice is to set w_i inversely proportional to $|\xi_i|$, which is of course not possible as ξ_i is unknown. An alternative approach, proposed first in [21,22], is to use approximate values of $|\xi_i|$ to set w_i. In details, an iterative approach is devised where, at iteration $k+1$, the WBPDN solution $\boldsymbol{\xi}^{(k)}$ from iteration k is used to generate the weights according to

$$w_i^{(k+1)} = \frac{1}{|\xi_i^{(k)}|^q + \epsilon}, \tag{3.4}$$

where $q > 0$ represents the strength of the penalization and ϵ is a small value to prevent numerical issues when ξ_i is zero. In our numerical experiments, we set $q = 2$ and $\epsilon = 10^{-4}$ as they consistently produce better solution. However, *optimal* values of q and ϵ may be selected along with λ using the approach discussed in Section 3.2. A similar approach for selecting q is employed in adaptive LASSO [21]. The iterations are started by solving the standard BPDN problem (2.5) to compute $\boldsymbol{\xi}^{(0)}$. Algorithm 1, adopted from [22], outlines the steps involved in WBPDN.

Similar to BPDN, WBPDN possesses the properties listed below [30]:

- Nonlinearity: WBPDN yields a solution $\boldsymbol{\xi}$ that is nonlinear in $\dot{\mathbf{x}}$.
- Limiting behavior as $\lambda \to 0$: the WBPDN solution tends to the ordinary least squares solution as $\lambda \to 0$.

- Finite convergence to zero as $\lambda \to \infty$: the WBPDN solution is zero for a finite value of λ defined as $\lambda \geq \lambda_{\max} = \|2\boldsymbol{\Phi}(\mathbf{x})^T\dot{\mathbf{x}}\|_\infty$, where $\|\cdot\|_\infty := \max|\cdot|$ is the infinity norm.
- Regularization path: there are values $0 = \lambda_k < \cdots < \lambda_1 = \lambda_{\max}$ such that the solution is a piece-wise linear curve on $\mathbb{R}^p$.

Algorithm 1 Iteratively Reweighted Basis Pursuit Denoising adopted from [22]

```
procedure WBPDN($\boldsymbol{\Phi}(\mathbf{x})$, $\dot{\mathbf{x}}$, $\lambda$, $q$, $\epsilon$)
    Set the iteration counter to $k = 0$ and $w_i^{(0)} = 1, \quad i = 1, \dots, p$.
    while not converged or $k < k_{\max}$ do
        Solve the WBPDN problem (3.2) for a specific $\lambda$
            $\boldsymbol{\xi}^{(k)} = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \left\{ \|\boldsymbol{\Phi}(\mathbf{x})\boldsymbol{\xi} - \dot{\mathbf{x}}\|_2^2 + \lambda\|\mathbf{W}^{(k)}\boldsymbol{\xi}\|_1 \right\}.$
        Update the weights for $i = 1, \dots, p$
            $w_i^{(k+1)} = \frac{1}{|\xi_i^{(k)}|^q + \epsilon}.$
        $k = k + 1$.
    end while
end procedure
```

Remark. It is worthwhile highlighting that Algorithm 1 is computationally more expensive than STLS and STRidge, as it requires (multiple) solutions of the WBPDN problem (3.2). In situations where the dimension n of the states and the size p of the basis $\{\phi_i(\mathbf{x})\}$ are small, the overall cost of recovering the governing equations is dominated by the data collection step.

3.2. Selection of λ via Pareto curve

The Pareto curve is a graph that traces the trade-off between the residual and the regularization constraint by varying the parameter λ in (3.2). In Tikhonov regularization, where ℓ_2-norm is used as a regularizer, the Pareto curve is called L-curve when a log–log scale is employed [43]. In this work, the Pareto curve is defined as the graph generated by solving (3.2) for different λ values in the $(\|\boldsymbol{\Phi}(\mathbf{x})\boldsymbol{\xi} - \dot{\mathbf{x}}\|_2, \|\mathbf{W}^{(k)}\boldsymbol{\xi}\|_1)$ space. Recall that λ controls the sparsity of the solution $\boldsymbol{\xi}$; setting $\lambda = 0$ yields the ordinary least squares solution, which is not sparse. As λ increases, the non-sparse solutions are increasingly penalized. In contrast to the ℓ_2-norm regularization, where the solution tends to zero as λ tends to infinity, the ℓ_1-norm regularization yields an upper bound on the regularization parameter given by $\lambda_{\max} = \|2\boldsymbol{\Phi}(\mathbf{x})^T\dot{\mathbf{x}}\|_\infty$ (i.e. a $\lambda \geq \lambda_{\max}$ yields the zero solution). As proven in [44], the ℓ_1-norm Pareto curve is convex, continuously differentiable and non-increasing. Its slope at each point is given by $-1/\lambda$, as shown in Fig. 1. Therefore, the regularization parameter that yields an approximation close to the exact $\boldsymbol{\xi}$, within a noise-dependent distance, must live within $\lambda_{\min}$ and $\lambda_{\max}$. For ℓ_2-regularized least squares, Hansen [43] suggests the corner point criterion to select λ from the L-curve. The underlying idea is that the L-curve has an L-shaped corner located where the solution changes from being dominated by regularization errors, corresponding to the steepest part of the curve, to being dominated by noise errors, where the curve becomes flat. The corner point corresponds to an *optimal* balance between the sparsity of the solution and the quality of the fit. The present work adopts this corner point criterion to select the λ parameter from the log–log ℓ_1-norm Pareto curve.

For continuous Pareto curves, [43] suggests defining the corner point as the point with maximum curvature. For discrete curves, however, it becomes more complicated to define the location of the corner point. Hansen et al. [45] highlight the difficulties in computing the corner point from discrete L-curves built using a finite number of λ values. The discrete curves may contain small local corners other than the global one that may give sub-optimal regularization parameters. To alleviate this issue, they propose an adaptive pruning algorithm, where the best corner point is computed by using candidate corners from curves at different resolutions. Since the L-curves must be

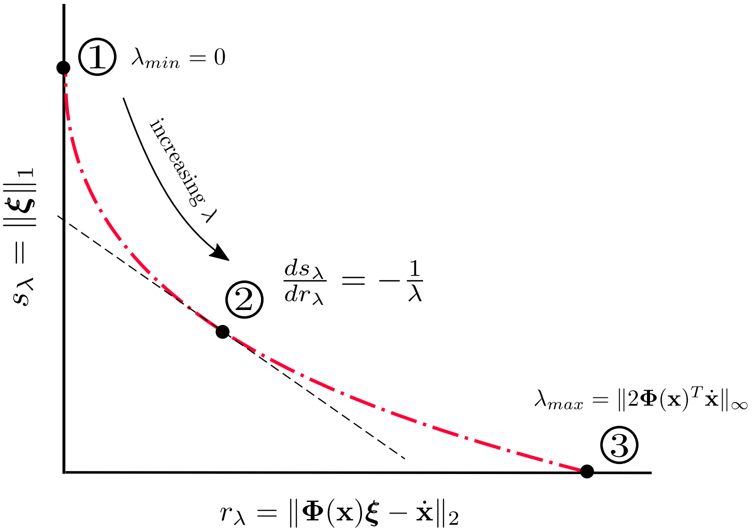
① $\lambda_{min} = 0$
increasing λ
② $\frac{ds_\lambda}{dr_\lambda} = -\frac{1}{\lambda}$
$\lambda_{max} = \|2\boldsymbol{\Phi}(\mathbf{x})^T\dot{\mathbf{x}}\|_\infty$
③
$s_\lambda = \|\boldsymbol{\xi}\|_1$
$r_\lambda = \|\boldsymbol{\Phi}(\mathbf{x})\boldsymbol{\xi} - \dot{\mathbf{x}}\|_2$

Fig. 1. Illustration of the ℓ_1-norm Pareto curve. Point 1 represents the unregularized least squares solution with infinite slope. Point 2 represents any point along the curve with slope $-1/\lambda$. Point 3 represents the upper bound on λ that gives the zero solution.

sampled at different scales, the pruning algorithm can be computationally expensive. We instead compute the corner points using a simpler method proposed in Cultrera et al. [46]. The algorithm iteratively employs the Menger curvature of a circumcircle and the golden section search method to efficiently locate the point on the curve with maximum curvature. The benefit of this algorithm is that it does not require computing the solution for a large set of λ values. Additionally, it consistently selects the corner point corresponding to smaller coefficient error across the numerical examples of Section 5.

Cross-validation (CV) is a popular approach for choosing the best regularization parameter λ in regression problems. However, when applied to (2.5), it is known that CV prefers smaller values of λ that are associated with smaller residual errors and less sparse coefficients. While CV was proposed to select the regularization parameter of adaptive LASSO, as we shall illustrate in the numerical examples of Section 5, the Pareto curve method produces more accurate results. Similarly, in [47], the Pareto method has been shown to outperform CV for sparse approximation using LASSO.

3.3. Numerical computation of state time derivatives $\dot{\mathbf{x}}$

In most practical applications, only the state trajectories are available, and their derivatives $\dot{\mathbf{x}}$ must be estimated numerically. To generate the data $\dot{\mathbf{x}}$ in the WBPDN problem (3.2), we seek to estimate the discrete time derivatives $\dot{x}_j(t_i)$, $i = 1, \dots, m$, at each time instances t_i from a set of discrete, noisy state trajectories $x_j(t_i)$.

The finite differences method is a common approach to compute derivatives from discrete data. However, if the data is corrupted with noise, the error in the derivative approximation by finite differences will generally be amplified. Several alternatives have been proposed to mitigate the effect of noise when computing derivatives: locally weighted regression [48], Tikhonov regularization [49], total variation regularization [50], to name a few. Following the work by Knowles et al. [49,51], we employ the second-order Tikhonov regularization approach, wherein the derivative is formulated as the solution to an integral equation and solved for using a Tikhonov-regularized least squares problem. The detail of our implementation is described in Appendix.

4. Identifying single constraints from data

Apart from satisfying governing equations of the form (2.1), some dynamical systems evolve under additional constraints on the state variables arising from certain conservation laws. For example, fluids are often assumed to satisfy the incompressibility condition, or the invariance of the total energy for conservative systems. In the case of a single constraint, the dynamical systems can be represented as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad \text{subject to} \quad g(\mathbf{x}) = 0, \tag{4.1}$$

where $g\colon \mathbb{R}^n \to \mathbb{R}$ is the constraint function. In this article, we focus on learning from data single constraints that are functions of the state variables with no explicit dependence on time. In the noise-free case, assuming that the constraint can be expressed in the same basis $\{\phi_k(\mathbf{x})\}$ for recovering the governing equations in (2.2), $g(\mathbf{x})$ is given by

$$g(\mathbf{x}) = \sum_{k=1}^{p} \eta_k \phi_k(\mathbf{x}) = 0, \tag{4.2}$$

for some unknown coefficients η_k. As an example, consider an undamped single degree of freedom spring–mass system. The conservation of energy can be put in implicit form as $g(\mathbf{x}) = -2\mathcal{E} + kx_1^2 + mx_2^2$, where $\mathcal{E}$ is the total energy of the system, k the spring stiffness, m the mass, and x_1 and x_2 the displacement and velocity of the system, respectively. In this case, the active bi-variate monomial basis functions in (4.2) are $\phi_1(\mathbf{x}) = 1$, $\phi_4(\mathbf{x}) = x_1^2$, $\phi_6(\mathbf{x}) = x_2^2$ and the corresponding coefficients are respectively $\eta_1 = -2\mathcal{E}$, $\eta_4 = k$ and $\eta_6 = m$.

Evaluating (4.2) at the state samples and letting $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_p]^T \in \mathbb{R}^p$, we arrive at

$$\boldsymbol{\Phi}(\mathbf{x})\boldsymbol{\eta} = \mathbf{0}, \tag{4.3}$$

which implies that $\boldsymbol{\Phi}(\mathbf{x})$ has a non-empty null space, i.e., is rank-deficient, and the solution $\boldsymbol{\xi}$ in (2.3) is non-unique. In practice, state measurements contain noise and the constraints are not satisfied exactly. As a result, the columns of $\boldsymbol{\Phi}(\mathbf{x})$ are *nearly* linearly dependent and $\boldsymbol{\Phi}(\mathbf{x})$ may have a large condition number. This, in turn, results in high sensitivity of the solution $\boldsymbol{\xi}$ to the noise in $\mathbf{x}$. In addition to learning the constraint $g(\mathbf{x})$, the identification of dependent columns of $\boldsymbol{\Phi}(\mathbf{x})$ provides a way to improve its condition number.

To this end, we perform a low-rank factorization of $\boldsymbol{\Phi}(\mathbf{x})$ via interpolative decomposition (ID) [26] in order to identify a subset of the columns of $\boldsymbol{\Phi}(\mathbf{x})$ that form a basis for its range space. In detail, using a rank-revealing QR factorization (RRQR) [52],

$$\boldsymbol{\Phi}(\mathbf{x})\mathbf{P} \approx \mathbf{QR}, \tag{4.4}$$

where $\mathbf{P} \in \mathbb{R}^{p\times p}$ is a permutation matrix, $\mathbf{Q} \in \mathbb{R}^{m\times r}$ has r orthogonal columns, and $\mathbf{R} \in \mathbb{R}^{r\times p}$ is an upper triangular matrix. Partitioning $\mathbf{R}$ into $\mathbf{R}_1 \in \mathbb{R}^{r\times r}$ and $\mathbf{R}_2 \in \mathbb{R}^{r\times(p-r)}$ and assuming the relation $\mathbf{R}_2 = \mathbf{R}_1\mathbf{Z}$ for $\mathbf{Z} = \mathbf{R}_1^{-1}\mathbf{R}_2$ yields

$$\boldsymbol{\Phi}(\mathbf{x})\mathbf{P} \approx \mathbf{QR}_1[\mathbf{I} \mid \mathbf{Z}], \tag{4.5}$$

where $\mathbf{I} \in \mathbb{R}^{r\times r}$ is the identity matrix. The rank-r factorization (4.5) can be rewritten as

$$\boldsymbol{\Phi}(\mathbf{x}) \approx \mathbf{S}[\mathbf{I} \mid \mathbf{Z}]\mathbf{P}^T = \mathbf{SC}, \tag{4.6}$$

where $\mathbf{S} = \mathbf{QR}_1$ contains r columns of $\boldsymbol{\Phi}(\mathbf{x})$ – specifically, the first r columns of $\boldsymbol{\Phi}(\mathbf{x})\mathbf{P}$ – and $\mathbf{C} = [\mathbf{I} \mid \mathbf{Z}]\mathbf{P}^T$ is the coefficient matrix specifying the linear combinations of those columns that approximate $\boldsymbol{\Phi}(\mathbf{x})$. Let $\mathcal{I} = \{1, \ldots, p\}$ and $\mathcal{J} = \{j_1, \ldots, j_r\} \subset \mathcal{I}$ denote, respectively, the set of indices of columns of $\boldsymbol{\Phi}(\mathbf{x})$ and the subset of those forming $\mathbf{S}$. Given the ID approximation (4.6), any column $\boldsymbol{\phi}_l(\mathbf{x})$ of $\boldsymbol{\Phi}(\mathbf{x})$ with index $l \in \mathcal{I}\backslash\mathcal{J}$ therefore satisfies the approximate relation

$$\sum_{m=1}^{r} C_{m,l}\boldsymbol{\phi}_{j_m}(\mathbf{x}) - \boldsymbol{\phi}_l(\mathbf{x}) \approx \mathbf{0}, \quad l \in \mathcal{I}\backslash\mathcal{J}, \tag{4.7}$$

in which $C_{m,l}$ is the entry (m, l) of $\mathbf{C}$. Rewriting (4.7) in the form of (4.3), the coefficients η_k in (4.3) are approximated by

$$\eta_k = \begin{cases} C_{m,l} & \text{if} \quad k = j_m, \quad m = 1, \ldots, r, \\ -1 & \text{if} \quad k = l, \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

Notice that the $\mathbf{C}$ matrix may contain non-zero but small elements due to the noise in the data. We therefore propose a thresholding step by setting to zero the elements of $\mathbf{C}$ whose magnitude is smaller than a threshold parameter τ.

To improve the conditioning of $\boldsymbol{\Phi}(\mathbf{x})$ in order to calculate $\boldsymbol{\xi}$ several paths may be taken. We may remove columns with indices in $\mathcal{I}\backslash\mathcal{J}$, i.e., replace $\boldsymbol{\Phi}(\mathbf{x})$ by $\mathbf{S}$, or remove columns with indices in $\mathcal{I}$ to which the columns in $\mathcal{I}\backslash\mathcal{J}$ depend on. In doing so, for the interest of arriving at a *simpler* model, we may remove the monomial basis functions of highest degree. If we happen to remove a basis function $\phi_i(\mathbf{x})$ that appears in the original dynamical system, the corresponding components of $\boldsymbol{\xi}$ will not be recovered. However, still a correct $\mathbf{f}(\mathbf{x})$ is computed. The procedure to learn single constraints $g(\mathbf{x})$ using ID of $\boldsymbol{\Phi}(\mathbf{x})$ and remove dependent columns of $\boldsymbol{\Phi}(\mathbf{x})$ is outlined in Algorithm 2.

Algorithm 2 Identification of constraints via ID

```
1: Compute the singular values of $\boldsymbol{\Phi}(\mathbf{x})$.
2: Identify a gap on the singular values and find the numerical rank $r$ of $\boldsymbol{\Phi}(\mathbf{x})$.
3: Perform rank-$r$ ID of $\boldsymbol{\Phi}(\mathbf{x}) \approx \mathbf{SC}$ given in (4.6) and obtain $\mathcal{J} = \{j_1, \dots, j_r\}$.
4: Threshold $\mathbf{C}$ by setting entries of $\mathbf{C}$ smaller than $\tau$ to zero.
5: Given $\mathbf{C}$ and $\mathcal{J} = \{j_1, \dots, j_r\}$, select any column $l \in \mathcal{I}\backslash\mathcal{J}$ and set the $\eta_k$ coefficients in (4.2) via (4.8).
6: Remove dependent columns of $\boldsymbol{\Phi}(\mathbf{x})$, e.g., set $\boldsymbol{\Phi}(\mathbf{x}) \longleftarrow \mathbf{S}$, for calculating $\boldsymbol{\xi}$.
```

Remark. Notice that the ID in (4.6) requires the knowledge of the numerical rank of $\boldsymbol{\Phi}(\mathbf{x})$, r, which may be detected by identifying a gap in the magnitude of the singular values of $\boldsymbol{\Phi}(\mathbf{x})$. However, depending on the level of noise in state measurements, such a gap may not exist or cannot be identified, thus rendering the proposed constraint learning approach inaccurate. In Section 5, we provide empirical results clarifying this remark.

Remark. In case the state variables are constrained by several implicit functions, there is no guarantee we are able to recover each individual one using Algorithm 2. This is because the state trajectory is confined to the intersection of the constraints, which leads to rank deficiency or ill-conditioning of $\boldsymbol{\Phi}(\mathbf{x})$. In the case of exact state measurements, removing the linearly dependent columns of $\boldsymbol{\Phi}(\mathbf{x})$ addresses the rank deficiency issue and reveals the constraint intersection, as illustrated in the numerical example of Section 5.5.

5. Numerical examples

In this section, we present numerical examples to assess the performance of the WBPDN method to recover the ordinary differential equations (ODEs) of five nonlinear dynamical systems. In all cases, we assume no prior knowledge of the governing equations that generated the data, except that they can be sparsely expressed in a multivariate polynomial basis in the state variables with known dimension. We only have access to the noisy state measurements at discrete times $t_k = k\Delta t$ sampled every Δt units of time. The exact state variables are computed by integrating the nonlinear ODEs using the fourth-order explicit Runge–Kutta (RK4) integrator implemented in MATLAB 2018b with a tolerance of 10^{-10}. We then corrupt the exact state variables by adding different levels of noise. In this work, we assume that the state variables are contaminated with independent zero-mean additive Gaussian noise with variance σ^2. The noise model is given by

$$x_j(t_k) = x_j^*(t_k) + \eta_j(t_k), \tag{5.1}$$

where $x_j^*(t_k)$, $j = 1, \dots, n$, denotes the exact state variable at time t_k, $k = 1, \dots, m$, and $\eta_j(t_k) \sim \mathcal{N}(0, \sigma)$. The noise levels are varied from $\sigma_{\min}$ to $\sigma_{\max}$ depending on the magnitude of the signal for each case. To measure the signal magnitude relative to the noise level, we provide the signal-to-noise ratio (SNR) for each state j. The SNR, expressed in decibels, is defined as

$$(\mathrm{SNR}_j)_{\mathrm{dB}} = 10\,\log_{10}\left(\frac{\sum_{k=1}^{m} x_j(t_k)^2}{\sigma^2}\right). \tag{5.2}$$

While here we work with a Gaussian white noise model, we note that correlated noise can be easily incorporated through the generalized least squares formulation of the WBPDN problem (3.2). For a thorough discussion on generalized least squares, we refer the interested reader to [53].

The sampling time is fixed to $\Delta t = 0.01$, and the number of samples depends on the simulation time used to capture the essential behavior of each dynamical system. As discussed in Section 3.3 and Appendix, we use Tikhonov regularization differentiation with a second-order differential operator as a smoothing constraint to compute the time derivative of state variables. Following the remark in Appendix, the state time derivatives are computed over an interval that is 5% (from each side) larger than the intended training time span, but only the data over the original training time is retained to compute $\boldsymbol{\xi}$. To show the quality of the resulting derivatives, we report the relative error

$$e_{\dot{x}} = \frac{\|\dot{\mathbf{x}} - \dot{\mathbf{x}}^*\|_2}{\|\dot{\mathbf{x}}^*\|_2}, \tag{5.3}$$

where $\dot{\mathbf{x}}$ and $\dot{\mathbf{x}}^*$ is the computed and true value of state derivatives, respectively.

We approximate the governing equations of each example by a multivariate monomial basis of total degree d in n state variables. That is, $\phi_i(\mathbf{x})$ in (2.2) (and (4.2)) are given by

$$\phi_i(\mathbf{x}) = \prod_{j=1}^{n} x_j^{i_j} \qquad \text{such that} \qquad \sum_{j=1}^{n} i_j \le d, \quad i_j \in \mathbb{N} \cup \{0\}, \tag{5.4}$$

where the non-negative integer i_j denotes the degree of the monomial in state variable x_j. The size of the approximation basis is then $p = \binom{n+d}{n}$. For all test cases, we set the total degree of basis d to one more than the highest total degree monomial present in the governing equations. Our primary goal is to demonstrate the consistency and robustness to noise of the WBPDN algorithm to identify an accurate approximation of the governing equations, as well as to enable the accurate prediction of future states of the system. We run the WBPDN optimization procedure using solveBP routine from the open-source SparseLab 2.1 package [54,55] and report the relative solution error defined as

$$e_{\xi} = \frac{\|\boldsymbol{\xi} - \boldsymbol{\xi}^*\|_2}{\|\boldsymbol{\xi}^*\|_2}, \tag{5.5}$$

where $\boldsymbol{\xi}$ and $\boldsymbol{\xi}^*$ are the approximate and exact solution vectors for each state variables, respectively.

The considered nonlinear dynamical systems are as follows: the Lorenz 63 system as a base model for identifying chaotic dynamics, the Duffing and Van der Pol oscillators as nonlinear stiffness and damping models, a harmonic single degree of freedom spring–mass system, and Euler Rigid Dynamics equations as conservative models satisfying physical constraints.

5.1. Lorenz 63 system

The Lorenz 63 system is a canonical model for nonlinear chaotic dynamics that was developed by Lorenz as a simplified model for atmospheric convection [56]. This system of nonlinear ODEs has been fundamental for the study of chaotic systems, wherein the future states of the system are highly sensitive to initial conditions. The state trajectories are three-dimensional, chaotic, deterministic, non-periodic and confined within a butterfly-shaped attractor, making them hard to predict. The Lorenz 63 model is given by the following system of first-order equations

$$\dot{x} = \gamma(y - x), \quad x(0) = x_0, \tag{5.6a}$$

$$\dot{y} = x(\rho - z) - y, \quad y(0) = y_0, \tag{5.6b}$$

$$\dot{z} = xy - \beta z, \quad z(0) = z_0, \tag{5.6c}$$

where the parameter values are set to $\gamma = 10$, $\rho = 28$ and $\beta = 8/3$, and the initial condition is $(x_0, y_0, z_0) = (-8, 7, 27)$. Note that the first state derivative is linear in the state variables, and the second and third ones contain quadratic nonlinearities. Assuming a degree $d = 3$ expansion for the right-hand-side of (5.6), is described exactly by seven of the $p = 20$ monomials.

We simulated the Lorenz 63 system from $t = 0$ to $t = 2.2$ time units to obtain the state trajectories. We then sampled the exact state variables at $\Delta t = 0.01$ resulting in $m = 221$ samples, and perturbed them with noise at different levels σ. The first step to recover the governing equations is to numerically compute state derivatives from noisy measurements. We performed Tikhonov-regularized numerical differentiation and truncated the discrete

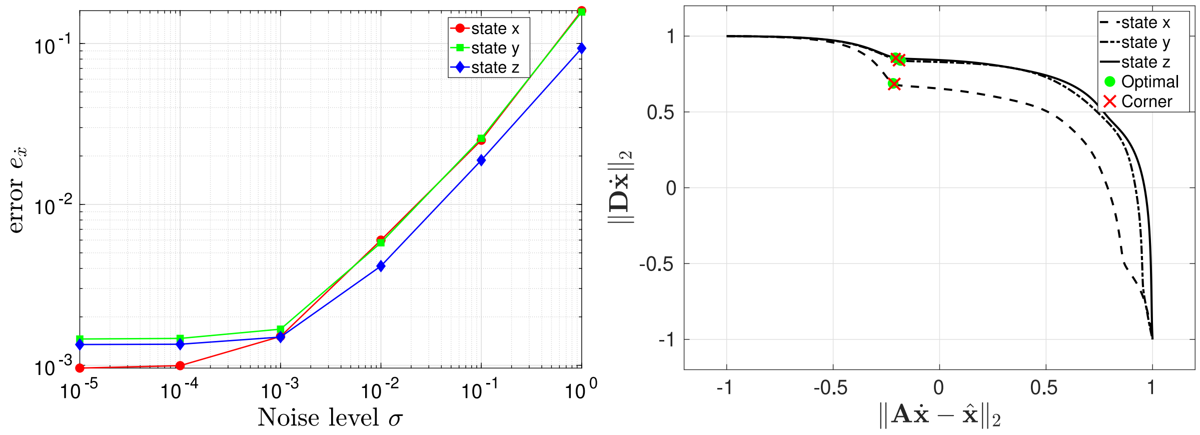

Fig. 2. Left: Relative error of Lorenz 63 state derivatives with respect to different noise levels. Right: Pareto curves for each state at noise level $\sigma = 10^{-2}$. The signal-to-noise ratios are: $\mathrm{SNR}_x = 58.11$ dB, $\mathrm{SNR}_y = 59.46$ dB, and $\mathrm{SNR}_z = 67.80$ dB.

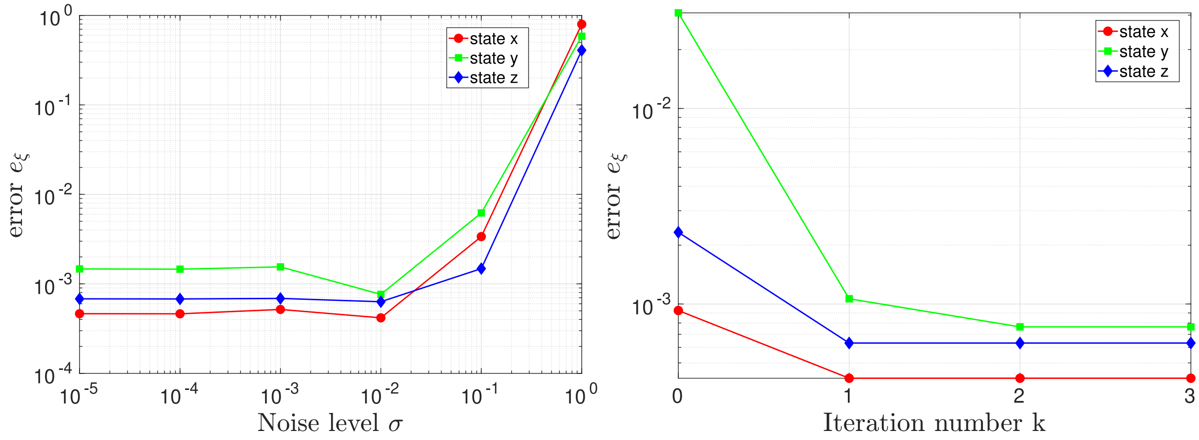

Fig. 3. Left: Relative solution error of Lorenz 63 with respect to different noise levels σ. Right: Relative solution error of Lorenz 63 with respect to the iteration count k at noise level $\sigma = 10^{-2}$.

state trajectory and resulting state derivatives yielding 201 samples from $t = 0.1$ to $t = 2.1$ time units. Fig. 2 (left) shows the relative error in the state derivatives with respect to different noise levels. The flat region on Fig. 2 (left) is dominated by time discretization errors $\sim \mathcal{O}(\Delta t^2)$ (independent of noise) when computing derivatives via numerical differentiation, whereas the steep region is dominated by noise. The Pareto curves are computed for each state at a noise level $\sigma = 10^{-2}$ in Fig. 2 (right). For clarity, the curves were normalized between $[-1, 1]$. The locations of the computed corner points match well with the optimal points corresponding to which the solution error is minimal.

Fig. 3 (left) illustrates the relative solution error as a function of the noise level σ. For small noise levels, the curve is almost flat and the solution error starts to rapidly increase around $\sigma \approx 10^{-2}$. Fig. 3 (right), shows the effectiveness of iteratively reweighting the coefficients; few iterations are required (two in this example) to converge and considerably reduce the error in the computed coefficients. Notice that iteration 0 of WBPDN is basically the standard BPDN solution.

The Pareto curves for each state at iteration 0 and noise level $\sigma = 10^{-2}$ are shown in the left panel of Fig. 4. We observe that the corner point criterion to select λ yields near optimal regularization. Here, we also compare the corner point criterion with K-fold CV with $K = 5$ to select the regularization parameter λ. As seen, λ obtained with CV matches the optimal and corners points for the x and y state variables, but is quite suboptimal for the z state. The reason for this discrepancy is that CV hinges upon selecting the regularization parameter that minimizes the

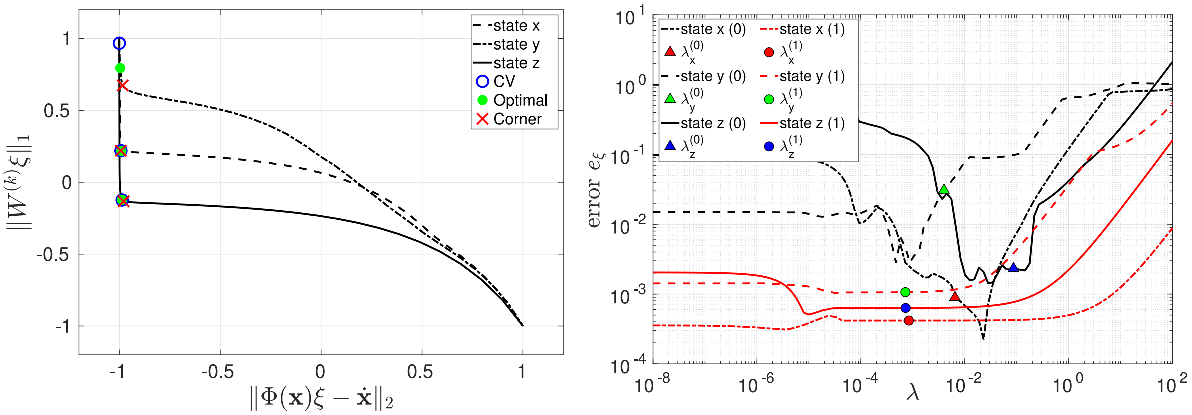

Fig. 4. Left: Pareto curves for each state of Lorenz 63 at noise level $\sigma = 10^{-2}$ and iteration 0. Right: Relative solution error as a function of the regularization parameter λ for each state at iterations 0 (in black) and 1 (in red). The colored markers indicate the λ corresponding to the converged corner points for each state at iterations 0 and 1 (the iteration number is enclosed in parenthesis). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

mean residual error over the folds, and does not take into account the ℓ_1-norm of the solution. In our experience, the CV function (mean residual error vs. λ) is usually flat for a wide range of λ, meaning that there exist many λ values that produce similar mean residual errors. The right panel of Fig. 4 shows the relative solution error as a function of λ for each state at iterations 0 and 1. Unlike in iteration 0 (i.e., BPDN), the solution error in iteration 1 (i.e. WBPDN) varies smoothly with respect to λ, and is insensitive to relatively large variations in λ around the selected value. Similar observations were made for the later iterations of WBPDN and other examples of this study. Additionally, the corner point of the Pareto curves corresponding to WBPDN iterations were better defined.

We now assess the performance of WBPDN by examining the prediction accuracy. Fig. 5 compares the exact trajectory of the Lorenz 63 system with the predicted trajectory of the identified model for $\sigma = 10^{-2}$. The exact trajectory is computed by integrating (5.6), whereas the predicted one is computed using the identified model starting at the same initial conditions. The training set used to identify the system – red points in Fig. 6 – ranges from $t = 0.1$ to $t = 2.1$, and the total simulation time is $t = 100$. As shown in Fig. 5, the predicted trajectory agrees with the exact one for short prediction times. Due to the chaotic nature of the system, even small errors in $\boldsymbol{\xi}$ may lead to inaccurate long-time state predictions.

Lastly, we compare WBPDN with STLS and STRidge algorithms proposed in [32,33]. The implementation details and original STLS[1] and STRidge[2] codes used in this article are publicly available online. In the case of STLS, we used two model selection criteria: K-fold CV with $K = 5$ and the ℓ_0-based Pareto curve, i.e., residual error versus ℓ_0-norm of $\boldsymbol{\xi}$, and corner point criterion to select the threshold parameter, as suggested in the supporting information for [13]. For CV, we used 201 samples (80% training and 20% validation split) over 2 time units, and default tolerances and maximum number of iterations. The state derivatives were computed using the same Tikhonov regularization differentiation algorithm. The results are illustrated in Fig. 7. All three algorithms exhibit similar solution error trends with respect to the noise level; a flat region for low noise levels and an error increase for high noise levels. Except for $\sigma = 10^1$ where all methods fail to recover the equations, WBPDN outperforms STLS and STRidge for all states and noise levels. STLS with ℓ_0-based Pareto curve produced the closest solution errors to WBPDN; however, it yielded considerable error for the y state.

5.2. *Duffing oscillator*

The Duffing oscillator features a cubic nonlinearity and can exhibit chaotic behavior. Physically, it models a spring–damper–mass system with a spring whose restoring force is $F(\zeta) = -\kappa\zeta - \varepsilon\zeta^3$. When $\varepsilon > 0$, it represents

[1] STLS (MATLAB): http://faculty.washington.edu/sbrunton/sparsedynamics.zip.

[2] STRidge (Python): https://github.com/snagcliffs/PDE-FIND.

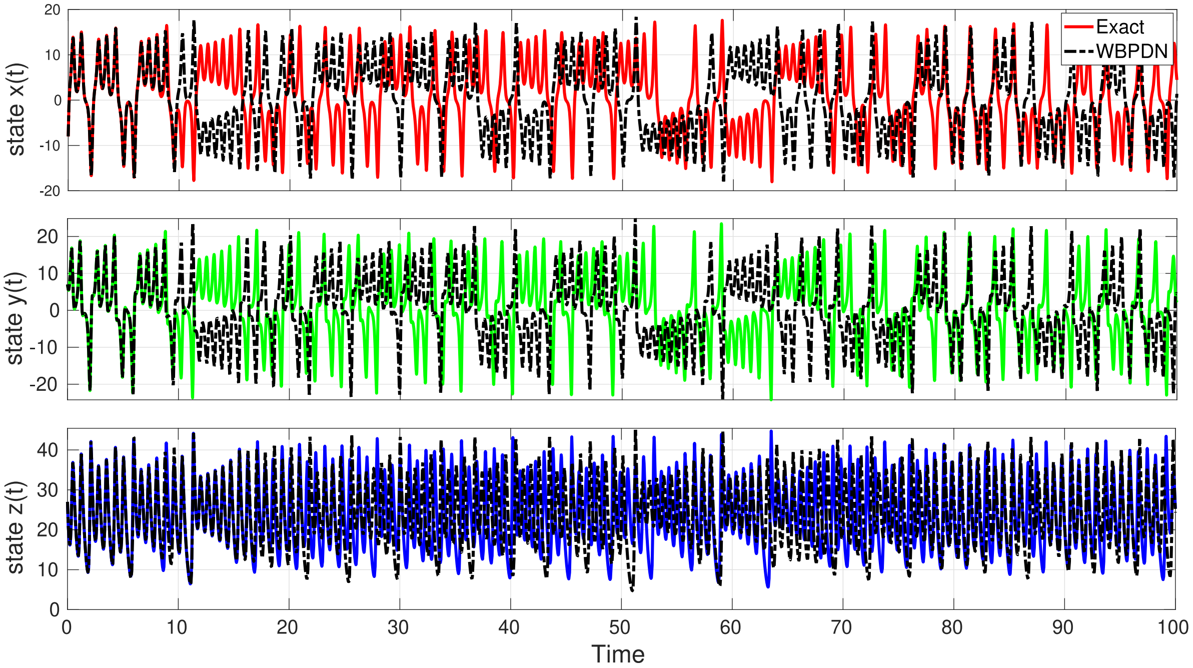

Fig. 5. Prediction of Lorenz 63 state trajectory by the identified model at noise level $\sigma = 10^{-2}$.

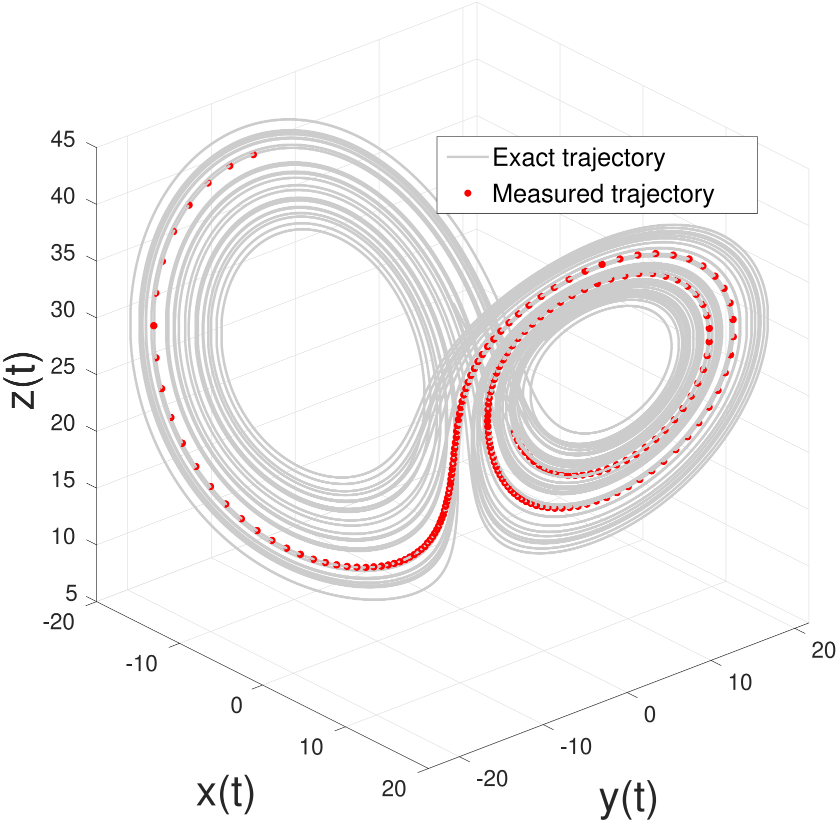

Fig. 6. Exact and measured state trajectory of the Lorenz 63 system.

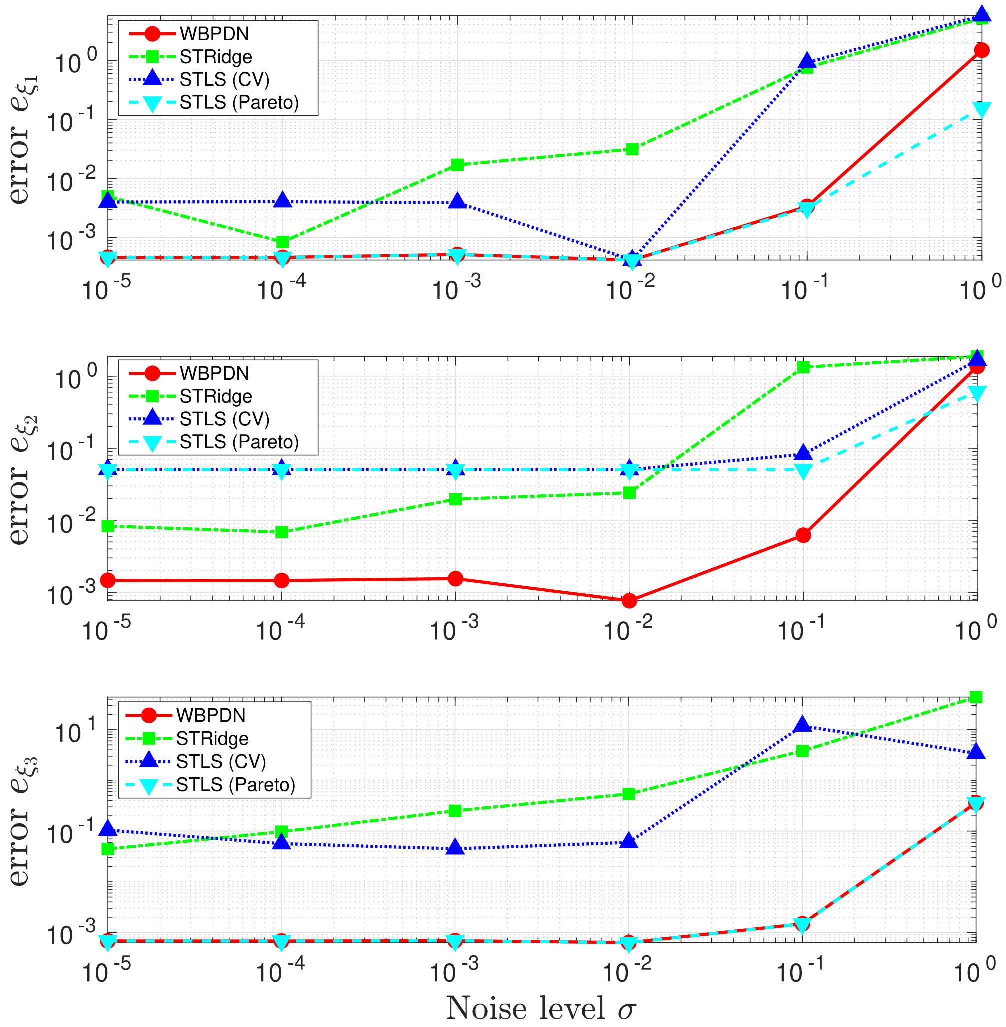

Fig. 7. Comparison between WBPDN, STRidge, STLS with CV and STLS with Pareto curve algorithms for the Lorenz 63 system for each state variable. Relative solution error with respect to different noise levels σ.

a *hard spring*. Conversely, for $\varepsilon < 0$ it represents a *soft spring*. The unforced Duffing equation is given by

$$\ddot{\zeta} + \gamma\dot{\zeta} + (\kappa + \varepsilon\zeta^2)\zeta = 0, \tag{5.7}$$

which can be transformed into a first-order system by setting $x = \zeta$ and $y = \dot{\zeta}$, giving

$$\dot{x} = y, \quad x(0) = x_0, \tag{5.8a}$$

$$\dot{y} = -\gamma y - \kappa x - \varepsilon x^3, \quad y(0) = y_0. \tag{5.8b}$$

The parameters of the system (5.7) are set to $\kappa = 1$, $\gamma = 0.1$ and $\varepsilon = 5$, and the initial conditions to $(x_0, y_0) = (1, 0)$. For these parameter values, the Duffing oscillator does not present chaotic behavior. The number of state variables is $n = 2$ and the degree of the polynomial basis is set to $d = 4$, yielding $p = 15$ monomial terms. Out of these, only four describe the dynamics. In this case, the displacement x and velocity y are measured and we used 201 samples over 2 time units, from $t = 0.1$ to $t = 2.1$, to recover the system. The errors of the state variable derivatives and the Pareto curves are shown in the left and right panels of Fig. 8, respectively.

As shown, the λ associated with the corner point agrees with the optimal one, yielding accurate state derivatives for each noise level. The effect of noise on the solution error, presented in Fig. 9, is consistent with previous results. Similar to the Lorenz problem, the WBPDN converges in only one iteration for $\sigma = 10^{-3}$.

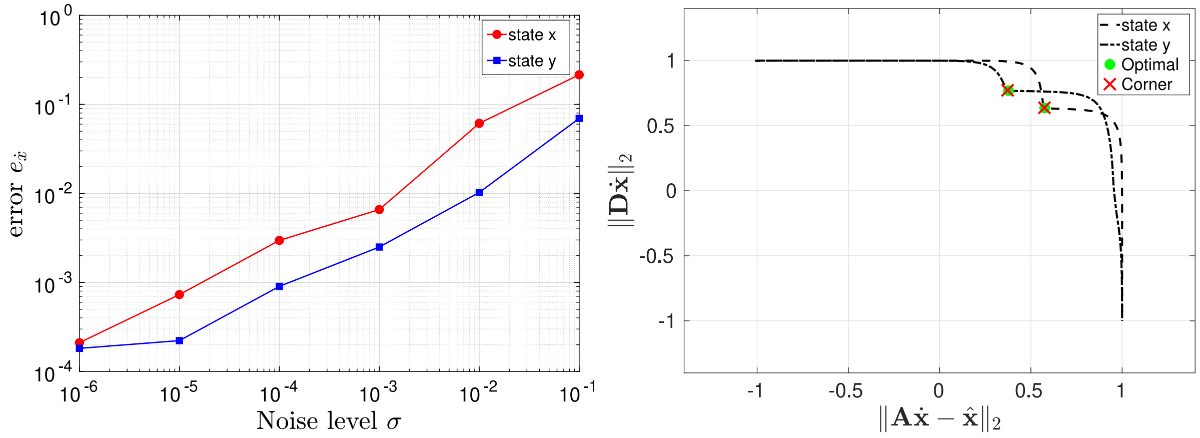

Fig. 8. Left: Relative error of the Duffing oscillator state derivatives with respect to different noise levels. Right: Pareto curves for each state of the Duffing oscillator at noise level $\sigma = 10^{-3}$. The signal-to-noise ratios are: $\text{SNR}_x = 52.58$ dB and $\text{SNR}_y = 57.44$ dB.

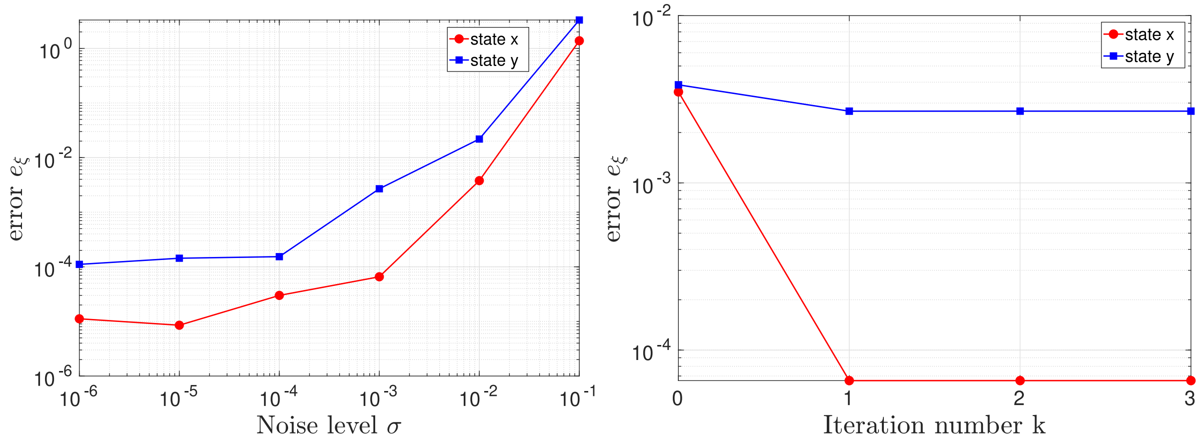

Fig. 9. Left: Relative solution error of the Duffing oscillator with respect to different noise levels σ. Right: Relative solution error of the Duffing oscillator with respect to iteration k at noise level $\sigma = 10^{-3}$.

Fig. 10 shows the effectiveness of the corner point criterion to find a regularization parameter close to the optimal, as opposed to the CV approach, which yields quite suboptimal λ estimates. The predicted trajectory matches the exact one, as illustrated in Fig. 11.

In this example, we also compared the performance of STLS, STRidge and WBPDN in recovering the coefficient vector $\boldsymbol{\xi}$. As in the Lorenz 63 example, we used an 80% training and 20% validation data split for both STLS with CV and STRidge. Fig. 12 displays the error in $\boldsymbol{\xi}$ at different noise levels. Overall, WBPDN outperforms both STLS and STRidge. In the case of STRidge, the solution happens to be unstable even for low noise levels. This may be because the default regularization parameters of the algorithm are not suitable for this problem and need careful tuning. In the STLS case, we also used CV and the Pareto criterion to set the threshold parameter γ. On the one hand, we noticed that the CV function, i.e., mean residual versus regularization parameter, is flat for the region where the regularization parameter with minimum solution error is located. This yields a minimum mean residual regularization parameter far from the optimal one, and therefore, inaccurate solution. On the other hand, we noticed that the corner point of the Pareto curve may not be well defined. Therefore, any algorithm trying to find the corner of the Pareto curve is deemed to fail. This fact may explain the sudden jumps in Fig. 12 for the STLS (Pareto) case.

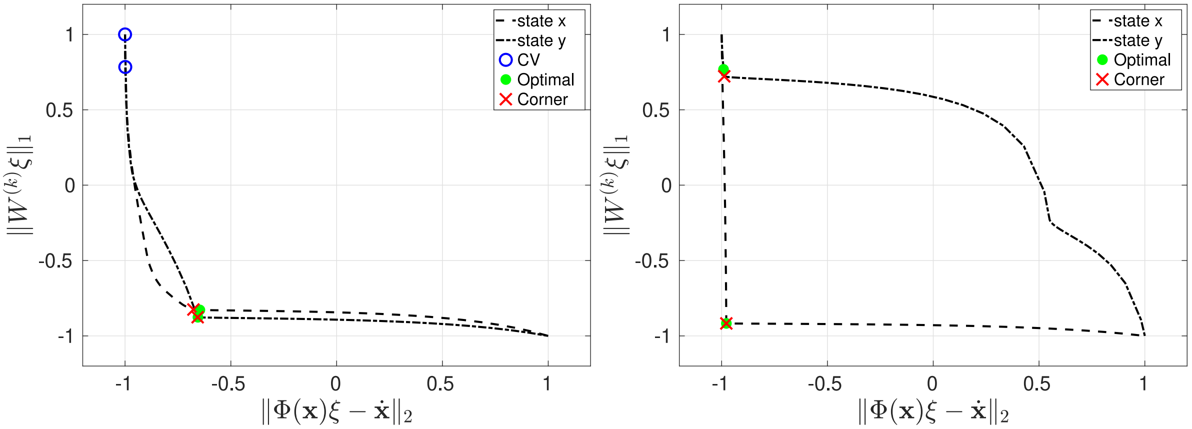

Fig. 10. Left: Pareto curves for each state of the Duffing oscillator at noise level $\sigma = 10^{-3}$ and iteration 0. Right: Pareto curves for each state of the Duffing oscillator at noise level $\sigma = 10^{-3}$ and iteration 1.

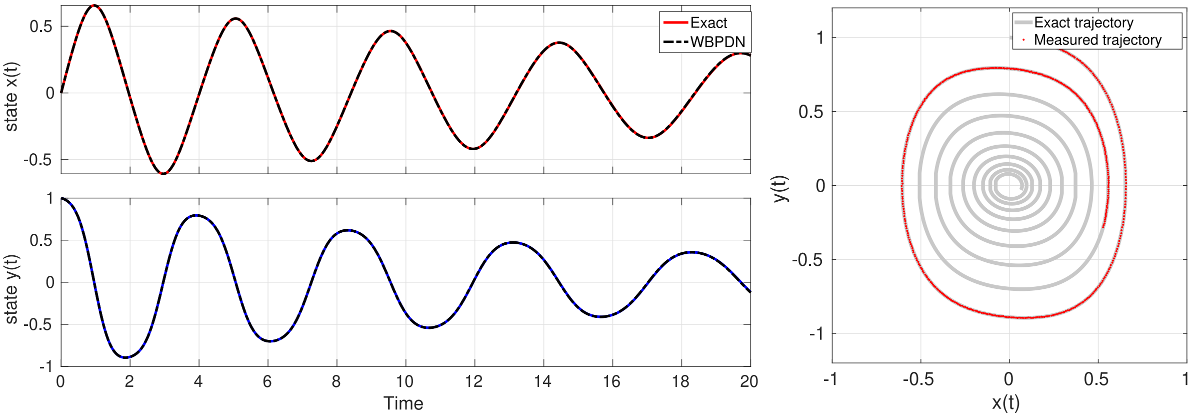

Fig. 11. Prediction of Duffing state trajectory by the identified model at noise level $\sigma = 10^{-3}$. Left: Prediction of Duffing state trajectory by the identified model at noise level $\sigma = 10^{-3}$. Right: Exact and measured state trajectory of the Duffing system.

5.3. Van der Pol oscillator

The Van der Pol model is a second-order oscillator with a nonlinear damping term. It was originally proposed by Van der Pol as a model to describe the oscillation of a triode in an electrical circuit [57]. The Van der Pol model exhibits a limit cycle behavior around the origin. The governing equation for this system is given by the following ODE

$$\ddot{\zeta} + (\gamma + \varepsilon\zeta^2)\dot{\zeta} + \kappa\zeta = 0, \tag{5.9}$$

which can be transformed into a first-order system as

$$\dot{x} = y, \quad x(0) = x_0, \tag{5.10a}$$

$$\dot{y} = -\kappa x - \gamma y - \varepsilon x^2 y, \quad y(0) = y_0. \tag{5.10b}$$

The parameters of the system (5.9) are set to $\kappa = 1$, $\gamma = 1$ and $\varepsilon = 2$, and the initial conditions to $(x_0, y_0) = (1, 0)$. As in the Duffing system, the sparsity of this system is four, the number of state variables is $n = 2$, and the degree of the polynomial basis is set to $d = 4$, giving $p = 15$ monomial terms. Again, the number of samples used in this example is 201 over 2 time units, ranging from $t = 0.1$ to $t = 2.1$. The performance of Tikhonov regularization differentiation with respect to different noise levels is shown in Fig. 13 (left), whereas Fig. 13 (right) shows the Pareto curves with corner points matching optimal λ values. This case illustrates a difficulty in locating

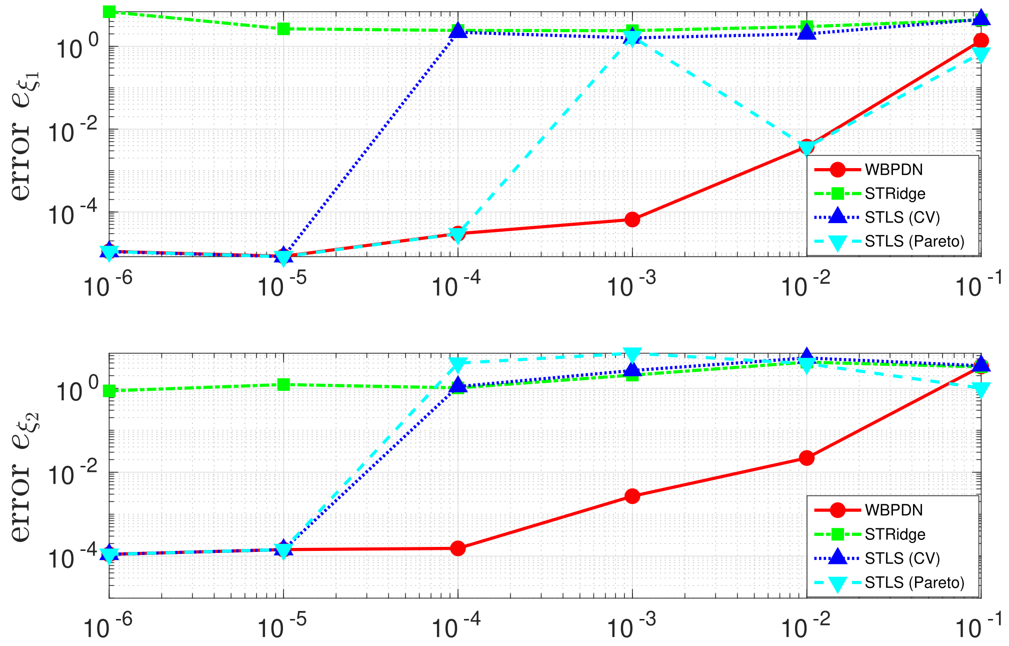

Fig. 12. Comparison between WBPDN, STRidge, STLS with CV and STLS with Pareto curve algorithms for the Duffing oscillator for each state variable. Relative solution error with respect to different noise levels σ.

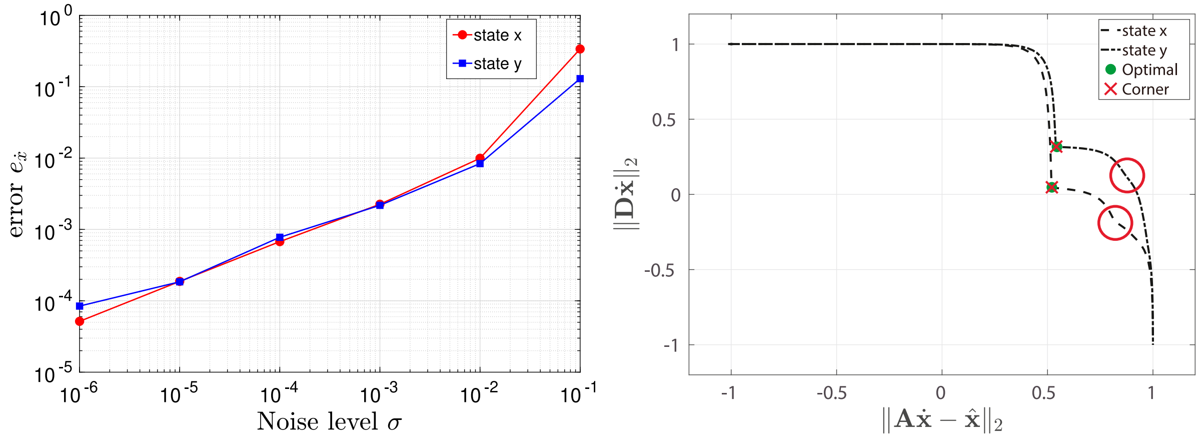

Fig. 13. Left: Relative error of the Van der Pol state derivatives with respect to different noise levels. Right: Pareto curves for each state of the Van der Pol oscillator at noise level $\sigma = 10^{-3}$. The red circles indicate additional corner points. The signal-to-noise ratios are: $\mathrm{SNR}_x = 62.41$ dB and $\mathrm{SNR}_y = 61.15$ dB.

global corner points instead of *local* ones. The Pareto curves for both state variables present two corner points for the range of λ used to generate them. The convergence to the wrong corner point can be avoided by selecting the one with higher curvature, by restricting the range of λ to some region close to the estimated residual given by the discrepancy principle [58], or by visually inspecting the Pareto curve plot and choosing the right corner point manually. Fortunately, the algorithm proposed in [46] is robust enough to select the right corner in this example as well as other examples of this work. Alternatively, the adaptive pruning algorithm in [45] may be used to detect multiple corner points and select the appropriate one.

The relative solution error agrees with the previous cases, as shown in Fig. 14. For low noise levels, the discretization errors on the state derivatives dominate the overall error of the solution. As we increase the noise

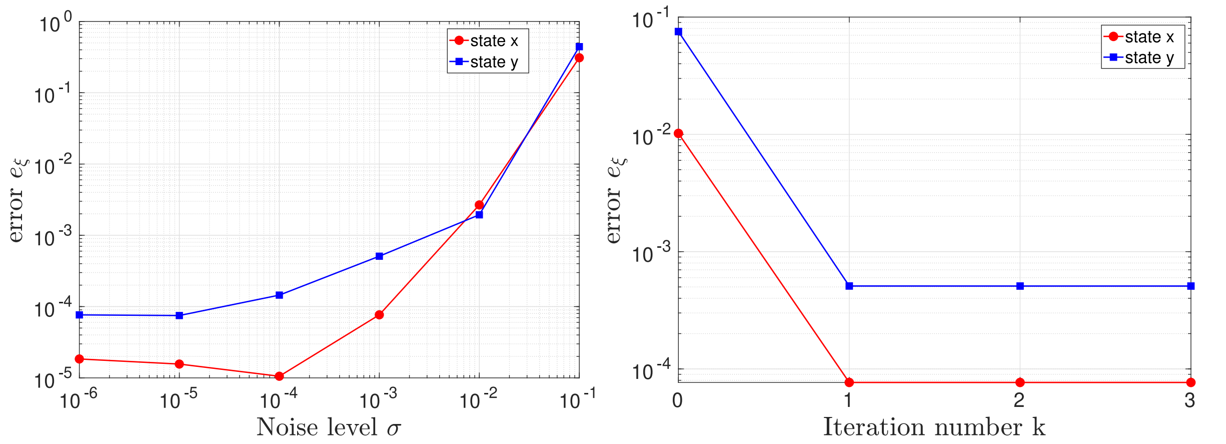

Fig. 14. Left: Relative solution error of the Van der Pol oscillator with respect to different noise levels σ. Right: Relative solution error of the Van der Pol oscillator with respect to iteration k at noise level $\sigma = 10^{-3}$.

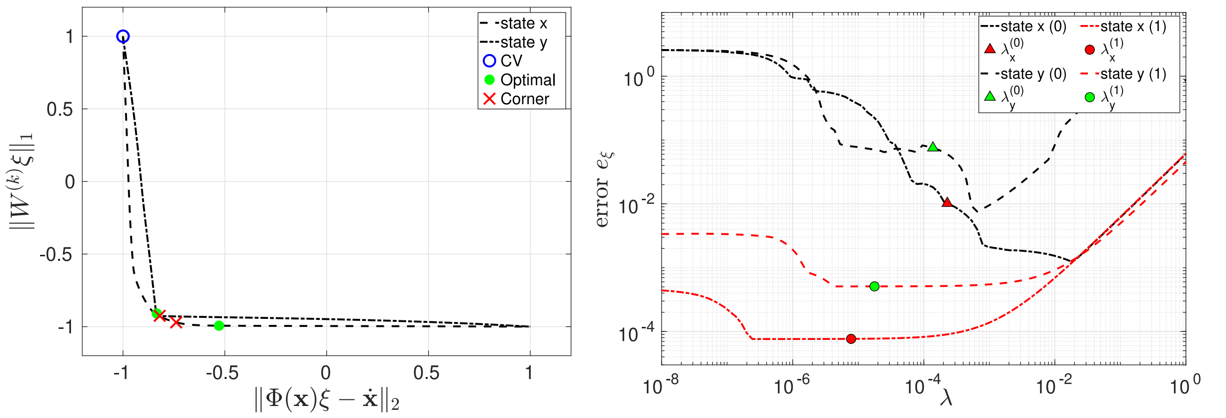

Fig. 15. Left: Pareto curves for each state of the Van der Pol oscillator at noise level $\sigma = 10^{-3}$ and iteration 0. Right: Relative solution error as a function of the regularization parameter λ for each state at iterations 0 (in black) and 1 (in red). The colored markers indicate the λ corresponding to the converged corner points for each state at iterations 0 and 1 (the iteration number is enclosed in parenthesis). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

level, the solution error is dominated by noise and starts to grow rapidly. Again, only one iteration is sufficient to converge and reduce the BPDN error by two orders of magnitude (for $\sigma = 10^{-3}$).

We observe from the left panel of Fig. 15 that, similar to previous cases, the Pareto curve criterion leads to better estimates of the optimal regularization parameter λ, as compared to the CV approach. The right panel of Fig. 15 illustrates the sensitivity of the solution error with respect to λ for iterations 0 and 1. As in the Lorenz 63 example, even though the minimum error is not attained, the solution at iteration 0 is accurate enough to make the iterative process improve the solution and eventually converge. Again, the solution $\boldsymbol{\xi}$ throughout the WBPDN iterations is less sensitive to variations in λ around the selected value by the Pareto curve approach. Similar to previous examples, the predicted and exact trajectories match well for $\sigma = 10^{-3}$; see Fig. 16.

Finally, the comparison among WBPDN, STRidge and STLS is displayed in Fig. 17. Similarly to previous examples, WBPDN yields the most accurate solution error. In this example, STRidge and STLS with CV produce inaccurate solutions for almost all noise levels. STLS with Pareto curve yields almost identical results as WBPDN for the x state variable. However, in general, it fails to recover the y state solution accurately.

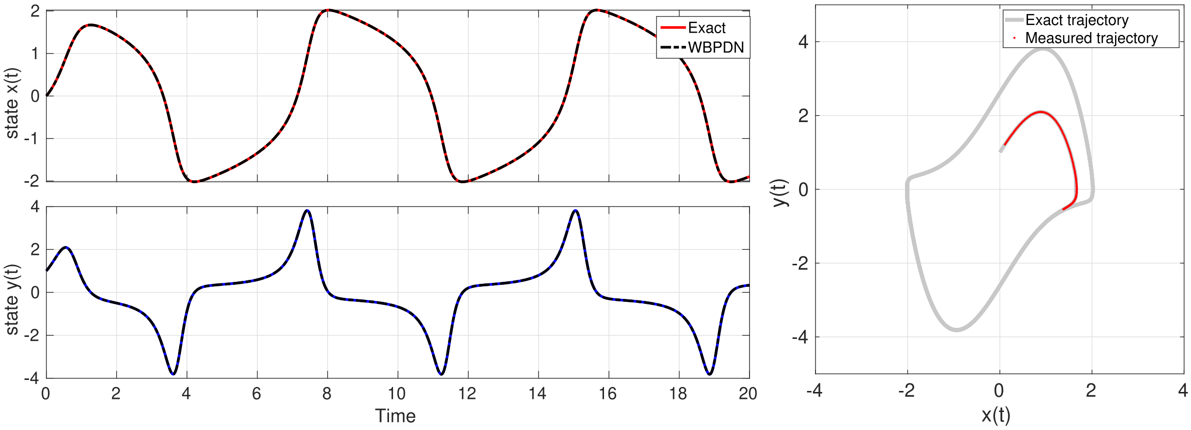

Fig. 16. Left: Prediction of Van der Pol state trajectory by the identified model at noise level $\sigma = 10^{-3}$. Right: Exact and measured trajectory for the Van der Pol oscillator at $\sigma = 10^{-3}$.

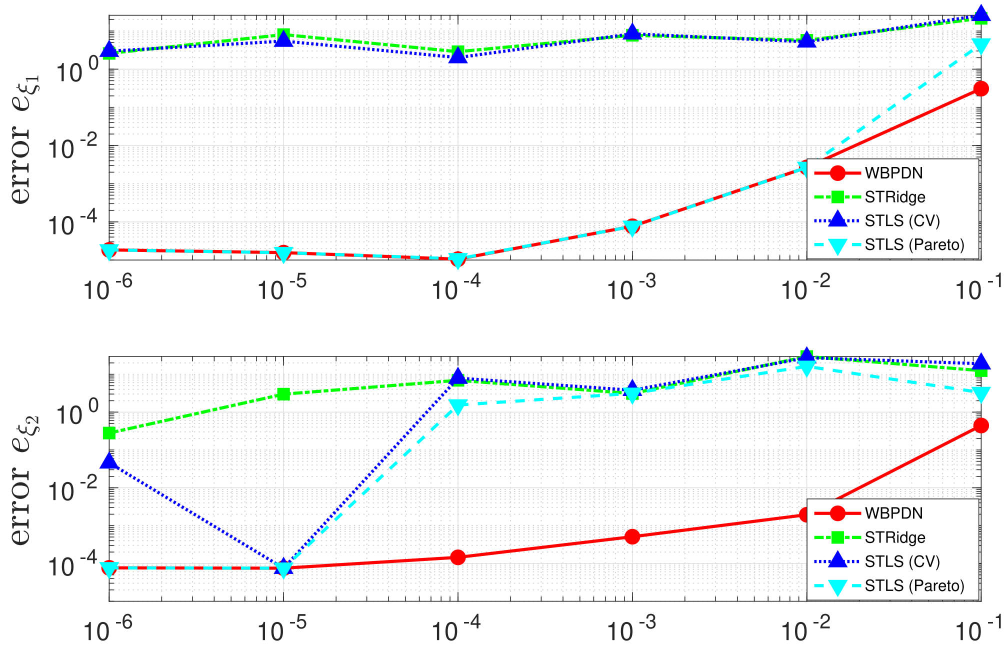

Fig. 17. Comparison between WBPDN, STRidge, STLS with CV and STLS with Pareto curve algorithms for the Van der Pol oscillator for each state variable. Relative solution error with respect to different noise levels σ.

5.4. Single degree of freedom spring–mass system

The single degree of freedom spring–mass system is a second-order harmonic oscillator that, when perturbed from its equilibrium, experiences a restoring force proportional to its displacement. In the absence of damping forces, the total energy of the system is conserved, resulting in a continuous transfer between potential and kinetic energy. The unforced harmonic oscillator is given by

$$m\ddot{\zeta} + k\zeta = 0, \tag{5.11}$$

where m, k and ζ are the mass, stiffness and displacement, respectively. The system can be transformed to a first order system as

$$\dot{x} = y, \quad x(0) = x_0, \tag{5.12a}$$

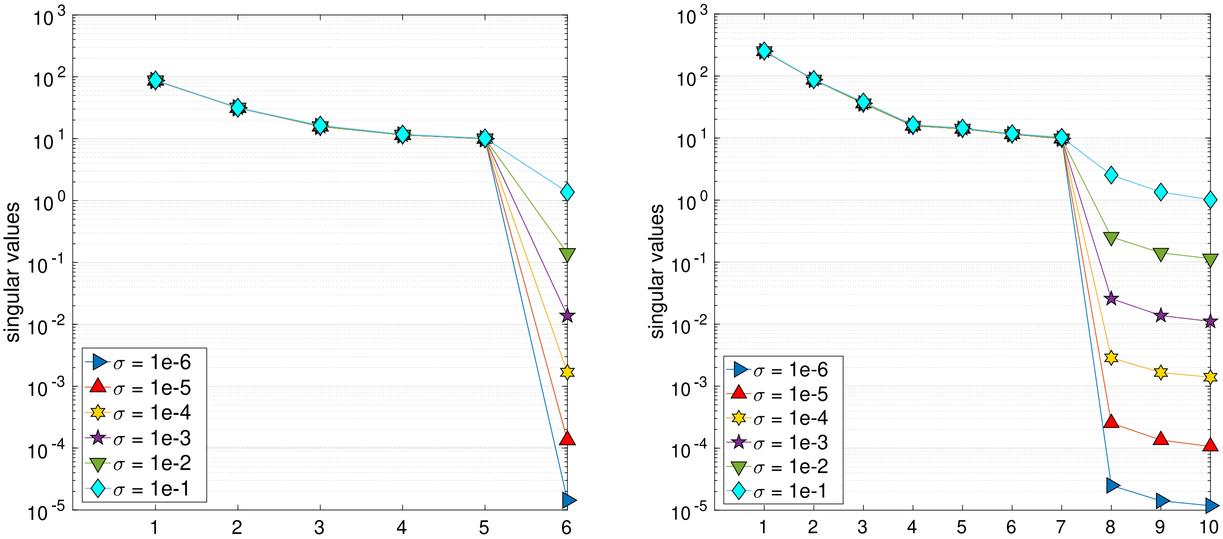

Fig. 18. Singular values of $\mathbf{\Phi}(\mathbf{x})$ for polynomial degree $d = 2$ (left) and $d = 3$ (right) at different noise levels σ. Note that the gap in the singular values becomes less evident at higher noise levels.

$$\dot{y} = -\omega^2 x, \quad y(0) = y_0, \tag{5.12b}$$

where $x = \zeta$, $y = \dot{\zeta}$, and $\omega = \sqrt{\frac{k}{m}}$ is the natural frequency of the system. In this example, the mass and stiffness are set to $m = 1$, $k = 10$, and the initial conditions to $(x_0, y_0) = (1, 0)$. The number of state variables is $n = 2$ and the sparsity of this system in the polynomial basis is two. The number of samples used in this example is 201 over 2 time units, from $t = 0.1$ to $t = 2.1$.

The total energy $\mathcal{E}$ is the sum of kinetic $\mathcal{T}$ and potential $\mathcal{V}$ energies given by

$$\mathcal{E} = \mathcal{V} + \mathcal{T} = \frac{1}{2}kx^2 + \frac{1}{2}my^2, \tag{5.13}$$

which represents the equation of an ellipse centered at the origin and semi-axis given by $a = \sqrt{2\mathcal{E}/k}$ and $b = \sqrt{2\mathcal{E}/m}$. It is straight-forward to verify that the corresponding constraint function $g(\mathbf{x})$ is given by

$$g(\mathbf{x}) = -1.0\phi_1(\mathbf{x}) + 1.0\phi_4(\mathbf{x}) + 0.1\phi_6(\mathbf{x}), \tag{5.14}$$

where $\phi_i(\mathbf{x}) \in \{1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3, \dots\}$, $i = 1, \dots, p$, i.e., $\phi_4(\mathbf{x}) = x^2$. Fig. 18 displays the singular values of $\mathbf{\Phi}(\mathbf{x})$ for polynomial degrees $d = 2$ (left) and $d = 3$ (right) to find the true ranks $r = 5$ and $r = 7$, respectively. As shown, the gap between the 5th and 6th largest singular values in the $d = 2$ case and the 7th and the 6th in the $d = 3$ case becomes smaller as we increase the noise level σ.

To identify the energy constraint, we then follow the procedure outlined in Algorithm 2 in Section 4 with the calculated numerical ranks. In the case of $d = 2$, the ID of $\mathbf{\Phi}(\mathbf{x})$ leads to $\mathcal{J} = \{2, \dots, 6\}$ as the indices of independent columns, i.e., first column of $\mathbf{\Phi}(\mathbf{x})$ depends linearly on the rest. This dependency is illustrated in Fig. 19 (left) in the form of a *dependency matrix*, where the horizontal axis is the index of each column of $\mathbf{\Phi}(\mathbf{x})$ and the vertical axis is the index of columns it depends on. White blocks show no dependency and the colored blocks show dependency with a magnitude obtained from the ID coefficient matrix $\mathbf{C}$ in (4.6). As can be observed, the first column of $\mathbf{\Phi}(\mathbf{x})$ depends on the 4th and 6th columns. Following (4.8), the recovered constraint function for the case of $d = 2$ and $\sigma = 10^{-3}$ is

$$g(\mathbf{x}) \approx -1.0000\phi_1(\mathbf{x}) + 0.9999\phi_4(\mathbf{x}) + 0.1000\phi_6(\mathbf{x}),$$

which closely matches (5.14), as also depicted in Fig. 20. Fig. 19 (right) shows the dependency matrix for the case of $d = 3$ at $\sigma = 10^{-3}$. Note that increasing the polynomial degree adds additional dependent columns in $\mathbf{\Phi}(\mathbf{x})$, which are trivial variants of (5.14). Specifically, one constraint is of the form $x = 0.9998x^3 + 0.1000xy^2$

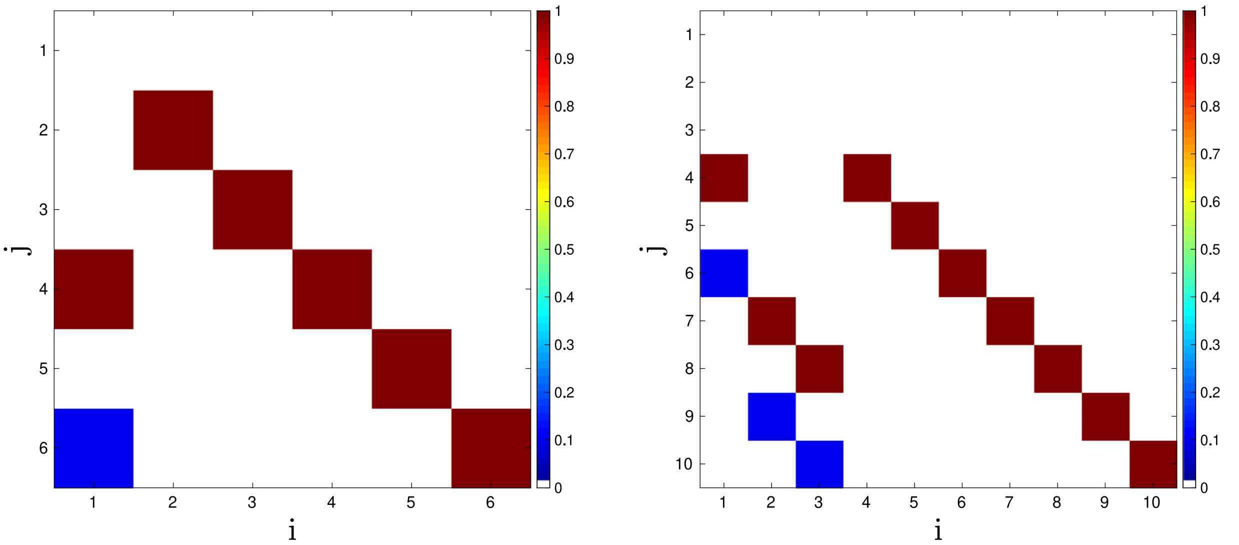

Fig. 19. Linear dependency matrix for $d = 2$ (Left) and $d = 3$ (Right) at noise level $\sigma = 10^{-3}$. The horizontal axis is the index i of each column of $\mathbf{\Phi}(\mathbf{x})$ and the vertical axis is the index j of columns it depends on. White blocks show no dependency and the colored blocks show dependency with a magnitude obtained from the ID coefficient matrix $\mathbf{C}$ in (4.6). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

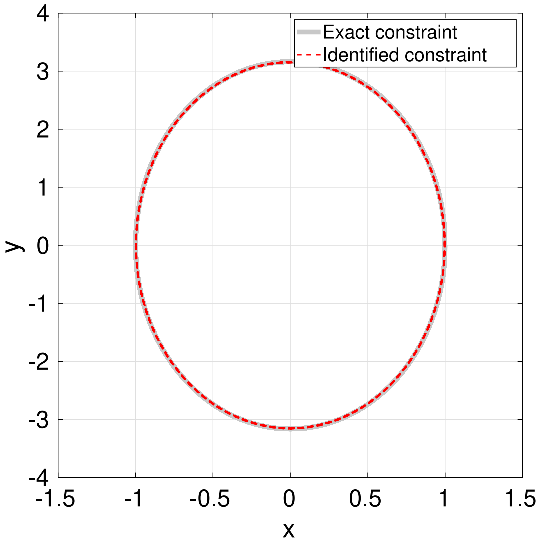

Fig. 20. Exact and identified energy constraint for the spring–mass problem with $d = 2$ at noise level $\sigma = 10^{-3}$.

or equivalently $x(-1 + 0.9998\phi_4 + 0.1000\phi_6) \approx xg(\mathbf{x}) \approx 0$ and the other one $y = 1.0001x^2y + 0.1000y^3$ or equivalently $y(-1 + 1.0001\phi_4 + 0.1000\phi_6) \approx yg(\mathbf{x}) \approx 0$.

We proceed with computing $\boldsymbol{\xi}$ using degree $d = 2$ basis by first removing one of the dependent columns of $\mathbf{\Phi}(\mathbf{x})$ to improve its conditioning. We may chose to eliminate either the 1st, 4th, or the 6th column of $\mathbf{\Phi}(\mathbf{x})$. With the aim of arriving at a *simpler*, i.e., lower order, model of the system, we remove the column associated with the highest polynomial degree, i.e., the 6th column in the case of $d = 2$. By doing so, the condition number of $\mathbf{\Phi}(\mathbf{x})$ is lowered from $1.46 \cdot 10^3$ to 3.16 for $\sigma = 10^{-3}$. The error of the state derivatives as well as the Pareto curves for the spring–mass system are illustrated in Fig. 21. The same trend as in the previous examples is shown in Fig. 22 for the relative solution error and the convergence of WBPDN for $\sigma = 10^{-3}$.

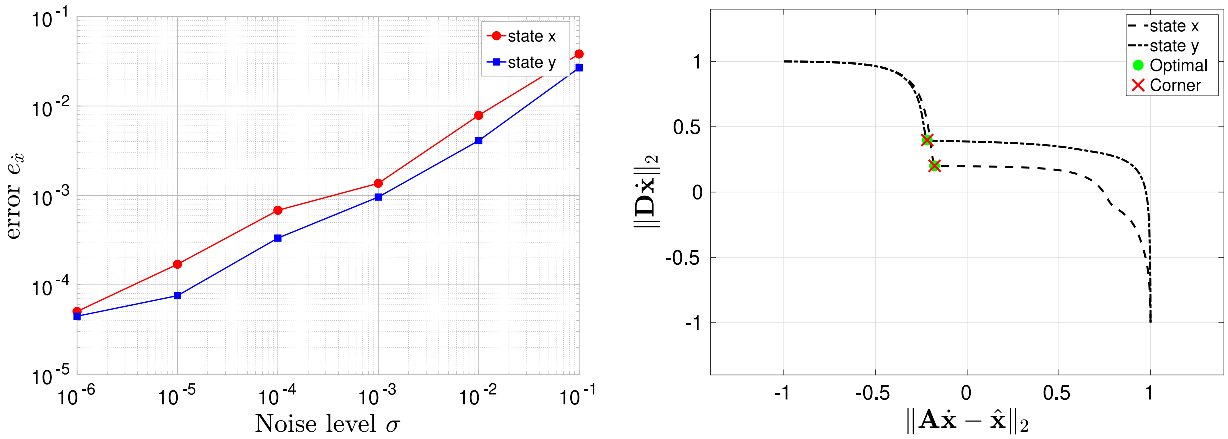

Fig. 21. Left: Relative error of the spring–mass state derivatives with respect to different noise levels. Right: Pareto curves for each state of the spring–mass system at noise level $\sigma = 10^{-3}$. The signal-to-noise ratios are: $\text{SNR}_x = 57.30$ dB and $\text{SNR}_y = 66.67$ dB.

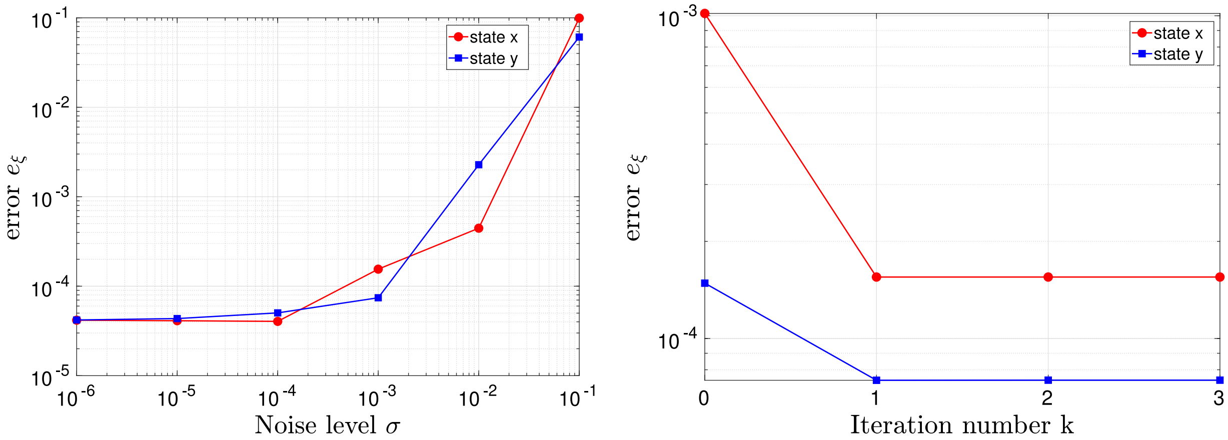

Fig. 22. Left: Relative solution error of the spring–mass system with respect to different noise levels. Right: Relative solution error of the spring–mass system with respect to iteration k at noise level $\sigma = 10^{-3}$.

The Pareto curves of Fig. 23 are smooth, and the location of the corner point is not as evident as in other examples. In fact, the corner points do not coincide with the optimal ones. This may be caused by the well-conditioning of the truncated $\mathbf{\Phi}(\mathbf{x})$. Because of this, the solution error does not depend on the regularization parameter as strongly. The exact and predicted state trajectories shown in Fig. 24 agree well.

5.5. Euler rigid body dynamics

In classical mechanics, Euler's rotation equations describe the rotation of a rigid body in a rotating reference frame with axes fixed to the body and parallel to the body's principal axes of inertia. These equations are widely known in the spacecraft dynamics community, where attitude performance is essential to meet pointing requirements. For instance, Earth observation satellites must achieve extreme accuracy to point antennas, optical instruments or other remote sensing devices toward specific targets. By improving spacecraft models and estimating physical parameters more accurately, one can enhance the overall performance of the mission. The equations in component form and derived with respect to the principal directions are given by

$$I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 = \tau_1, \quad \omega_1(0) = \omega_{1,0}, \tag{5.15a}$$

$$I_2\dot{\omega}_2 + (I_1 - I_3)\omega_3\omega_1 = \tau_2, \quad \omega_2(0) = \omega_{2,0}, \tag{5.15b}$$

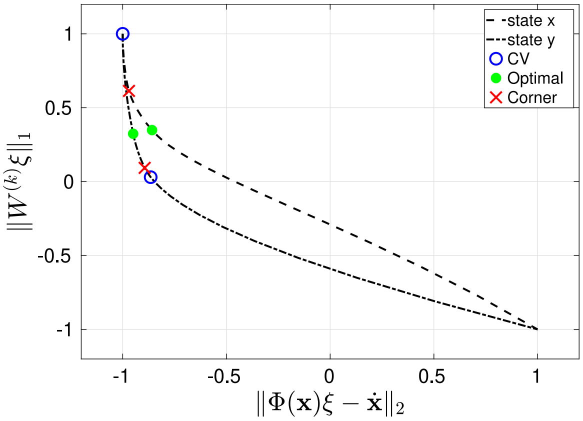

Fig. 23. Pareto curves for each state of the spring–mass system at noise level $\sigma = 10^{-3}$ and iteration 0.

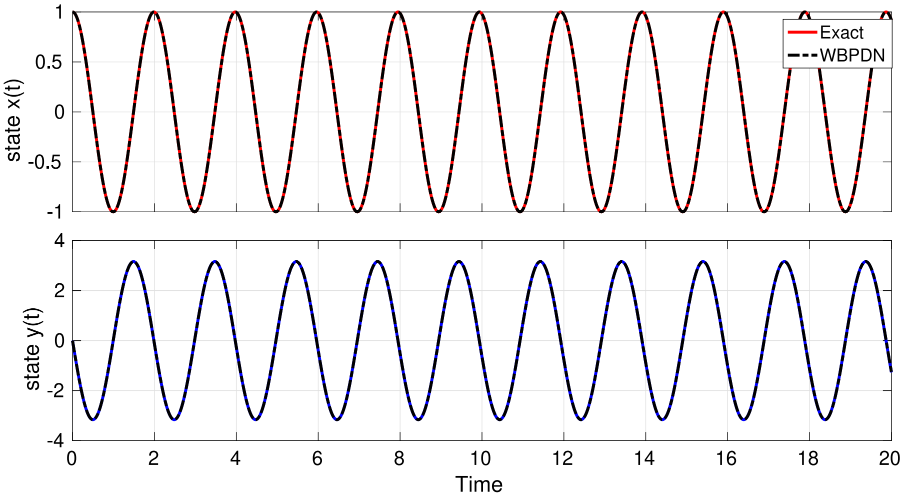

Fig. 24. Exact and predicted trajectories of the spring–mass system for $\sigma = 10^{-3}$.

$$I_3\dot{\omega}_3 + (I_2 - I_1)\omega_1\omega_2 = \tau_3, \quad \omega_3(0) = \omega_{3,0}. \tag{5.15c}$$

where I_k, ω_k, τ_k, $k = 1, 2, 3$, are the principal moments of inertia, the angular velocities, and the applied external torques, respectively. In this example, we assume no external torques excite the system, i.e., $\tau_1 = \tau_2 = \tau_3 = 0$. In the torque-free case, the Euler system (5.15) is conservative and satisfies two integrals of motion: conservation of kinetic energy (5.16a) and conservation of angular momentum (5.16b)

$$2\mathcal{T} = I_1\omega_1^2 + I_2\omega_2^2 + I_3\omega_3^2, \tag{5.16a}$$

$$\|\boldsymbol{L}\|_2^2 = I_1^2\omega_1^2 + I_2^2\omega_2^2 + I_3^2\omega_3^2, \tag{5.16b}$$

where $\mathcal{T}$ and $\boldsymbol{L}$ are the kinetic energy and angular momentum vector, respectively. The above constraints represent two ellipsoid-shaped manifolds in the $(\omega_1, \omega_2, \omega_3)$-space whose intersection curve is known as the polhode (Fig. 25). As the true state variables satisfy these two constraints, they lie on their intersection. The purpose of this example is to show that, while the proposed ID approach (Algorithm 2) is not able to learn these two constraints individually, it is able to learn their intersection for small noise levels.

In this example, the moments of inertia are set to $I_1 = 1$, $I_2 = 2$ and $I_3 = 3$, yielding a tri-inertial body, and the initial conditions to $(\omega_1, \omega_2, \omega_3) = (1, 1, 1)$. These specific values give $\mathcal{T} = 3$ and $\|\boldsymbol{L}\|_2^2 = 14$. The number of state variables is $n = 3$ and the degree of the polynomial basis is set to $d = 3$, giving $p = 20$ monomial terms. The sparsity of this system in a polynomial basis is three. We used 1000 samples over 10 time units, from

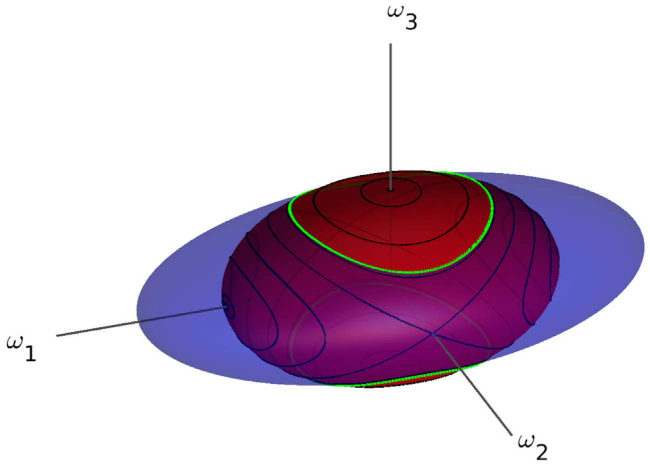

Fig. 25. The intersection of energy (red) and angular momentum (blue) constraint ellipsoids define the polhode curve (green), which corresponds to the state trajectory. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

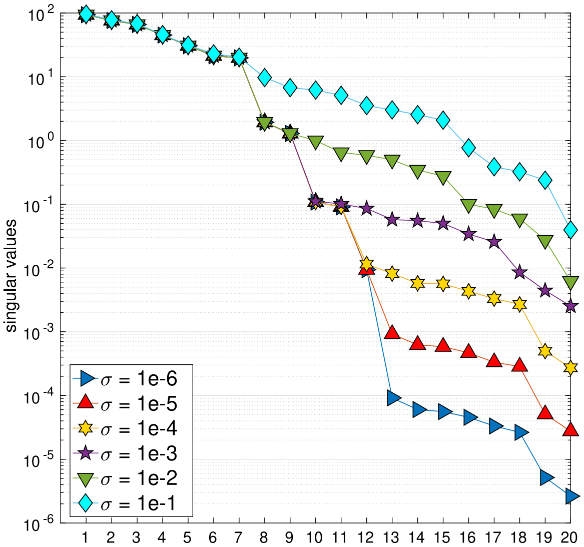

Fig. 26. Singular value decay of $\mathbf{\Phi}(\mathbf{x})$ at different noise levels for Euler Rigid Body Dynamics. In this example there is no clear gap on the singular values of $\mathbf{\Phi}(\mathbf{x})$. This may lead to the misconception that $\mathbf{\Phi}(\mathbf{x})$ is full-rank, where, in fact, there exist constraints making some of the columns approximately linearly dependent.

$t = 0.5$ to $t = 10.5$. The ellipsoid constraint functions associated with (5.16) are expressed as a linear combination of multivariate monomials as

$$g_1(\mathbf{x}) = 1\phi_1(\mathbf{x}) - \frac{1}{6}\phi_5(\mathbf{x}) - \frac{1}{3}\phi_7(\mathbf{x}) - \frac{1}{2}\phi_{10}(\mathbf{x}); \tag{5.17a}$$

$$g_2(\mathbf{x}) = 1\phi_1(\mathbf{x}) - \frac{1}{14}\phi_5(\mathbf{x}) - \frac{2}{7}\phi_7(\mathbf{x}) - \frac{9}{14}\phi_{10}(\mathbf{x}), \tag{5.17b}$$

where $\phi_i(\mathbf{x}) \in \{1, \omega_1, \omega_2, \omega_3, \omega_1^2, \omega_1\omega_2, \omega_2^2, \omega_1\omega_3, \omega_2\omega_3, \omega_3^2, \dots\}, i = 1, \dots, p$. Similar to the previous example, we compute the singular values of $\mathbf{\Phi}(\mathbf{x})$ for polynomial degree $d = 3$ at different noise levels σ to find the numerical rank. As shown in Fig. 26, except for the case of $\sigma = 10^{-6}$, there is no clear gap in the singular values. This presents an issue when learning the constraint via Algorithm 2 and subsequently computing $\boldsymbol{\xi}$.

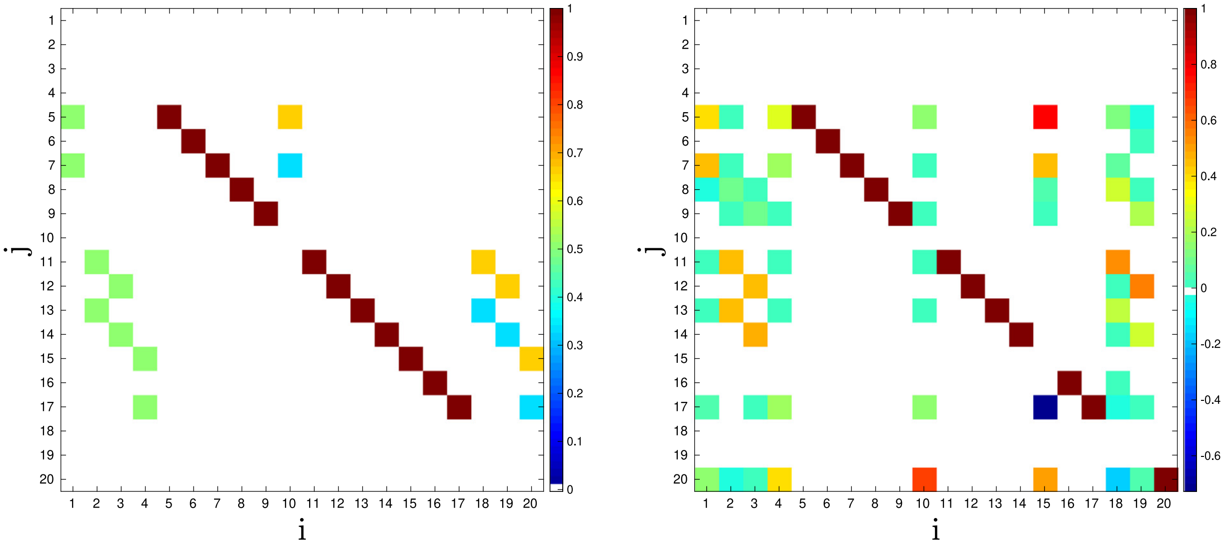

Fig. 27. Linear dependency matrix for Euler Rigid Body Dynamics at noise level $\sigma = 10^{-5}$ (Left) and $\sigma = 10^{-3}$ (Right). The horizontal axis is the index i of each column of $\mathbf{\Phi}(\mathbf{x})$ and the vertical axis is the index j of columns it depends on. White blocks show no dependency and the colored blocks show dependency with a magnitude obtained from the ID coefficient matrix $\mathbf{C}$ in (4.6). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We performed Algorithm 2 knowing the true rank *a priori*, which is 12 in this example. Fig. 27 illustrates the linear dependence matrix for $d = 3$ at two different noise levels $\sigma = 10^{-5}$ (left) and $\sigma = 10^{-3}$ (right). The ID leads to $\mathcal{J} = \{5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17\}$ as the indices of the independent columns of $\mathbf{\Phi}(\mathbf{x})$ for $\sigma = 10^{-5}$, which is the noise level we use for the results presented next. Larger noise levels, lead to several small η_k coefficients, which motivate future research in developing better denoising strategies for constraint identification. We select the 1st and 10th columns of $\mathbf{\Phi}(\mathbf{x})$ corresponding to the subset $\mathcal{I}\backslash\mathcal{J} = \{1, 2, 3, 4, 10, 18, 19, 20\}$ of linearly dependent columns to generate the following recovered constraint functions

$$\begin{aligned}\hat{g}_1(\mathbf{x}) &= -1.0000\phi_1(\mathbf{x}) + 0.5002\phi_5(\mathbf{x}) + 0.5001\phi_7(\mathbf{x});\\ \hat{g}_2(\mathbf{x}) &= 0.6625\phi_5(\mathbf{x}) + 0.3308\phi_7(\mathbf{x}) - 1.0000\phi_{10}(\mathbf{x}),\end{aligned}$$

also depicted in Fig. 28. In obtaining (5.18), we have thresholded the $C_{m,l}$ coefficients in (4.8) with a threshold parameter of size 10^{-3}. Notice that the recovered constraints $\hat{g}_1(\mathbf{x})$ and $\hat{g}_2(\mathbf{x})$ are different from the exact constraints $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$, respectively, hence the $\hat{}$ notation for the recovered constraints. This can be observed also by comparing Figs. 25 and 28. However, the intersection of $\hat{g}_1(\mathbf{x})$ and $\hat{g}_2(\mathbf{x})$ matches well with that of $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$, which is again what constrains the state variables. We note that selecting other columns with indices in $\mathcal{I}\backslash\mathcal{J}$ results in constraints that are trivial variants of $\hat{g}_1(\mathbf{x})$ and $\hat{g}_2(\mathbf{x})$.

This example shows the difficulty in uniquely identifying more than one constraints given by implicit functions of the state variables and using the ID approach. Given a basis $\mathbf{\Phi}(\mathbf{x})$, there exist a family of different hypersurfaces that contain the state trajectory, defined by their intersection. This is caused by the non-uniqueness of the solution to $\mathbf{\Phi}(\mathbf{x})\boldsymbol{\eta} = \mathbf{0}$; any linear combination of the null space vectors of $\mathbf{\Phi}(\mathbf{x})$ will also contain the state trajectory. Recovering constraints in the data poses new challenges in system identification. A thorough analysis of the conditions under which constraints can be uniquely identified is still being investigated and will be addressed in a future work.

In this case, Tikhonov regularization differentiation also performs well in computing derivatives from noisy measurements for all state variables. The Pareto curves are well defined, and the corners coincide with the optimal regularization parameters, as shown in Fig. 29.

Before performing WBPDN, we remove the linearly dependent columns of $\mathbf{\Phi}(\mathbf{x})$ in the subset $\mathcal{I}\backslash\mathcal{J}$ to lower the condition number of $\mathbf{\Phi}(\mathbf{x})$ from $3.40 \cdot 10^4$ to $1.96 \cdot 10^3$ for $\sigma = 10^{-3}$. The solution error and the convergence agree with the previous examples. We assume that the rank and the dimension of the null space of $\mathbf{\Phi}(\mathbf{x})$ are known. If we run the WBPDN on the numerically rank deficient matrix, we may obtain a different $\boldsymbol{\xi}$, yielding similar residuals

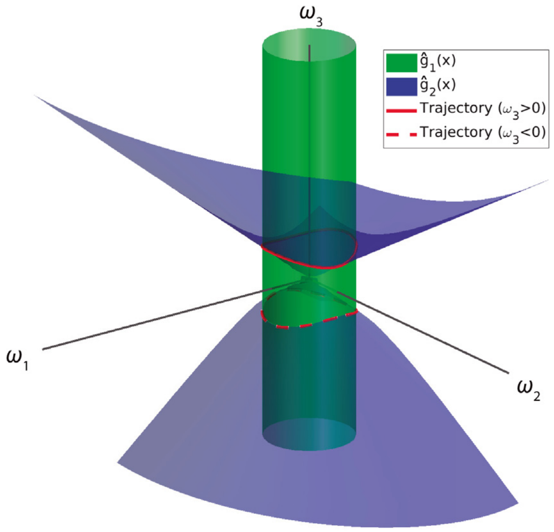

Fig. 28. Identified constraints $\hat{g}_1(\mathbf{x})$ and $\hat{g}_2(\mathbf{x})$ for the Euler Rigid Body Dynamics problem with $d = 3$ at noise level $\sigma = 10^{-5}$. The intersection of $\hat{g}_1(\mathbf{x})$ and $\hat{g}_2(\mathbf{x})$ matches well with the exact trajectory of the system shown in solid and dashed red curves. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

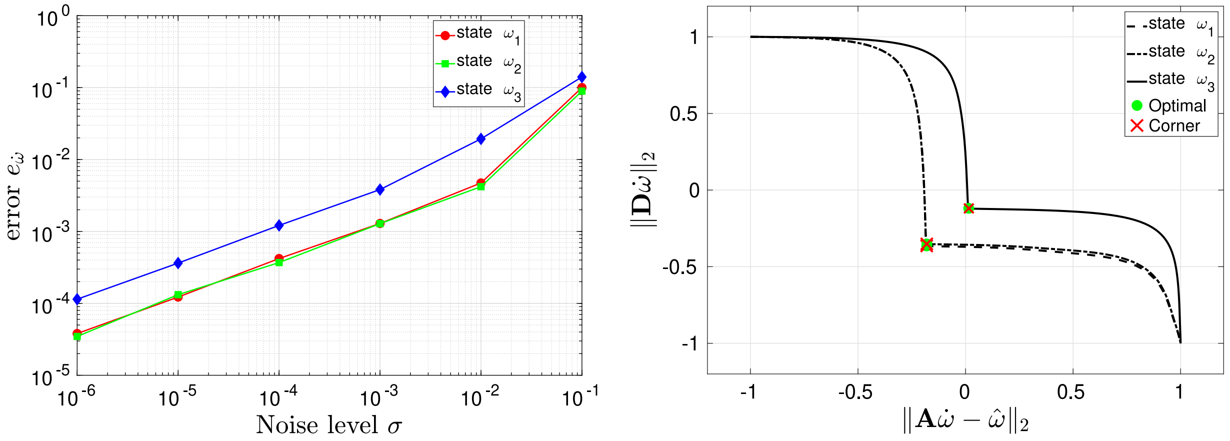

Fig. 29. Left: Relative error of the angular accelerations with respect to different noise levels. The signal-to-noise ratios are: $\text{SNR}_{\omega_1} = 58.18$ dB and $\text{SNR}_{\omega_2} = 60.68$ dB, $\text{SNR}_{\omega_3} = 59.75$ dB. Right: Pareto curves for each angular velocity at noise level $\sigma = 10^{-3}$.

and predicted trajectories caused by the non-uniqueness of the solution due to the ill-conditioning of $\mathbf{\Phi}(\mathbf{x})$. Fig. 31 displays results for the reduced $\mathbf{\Phi}(\mathbf{x})$ with $p = 12$ columns and the original one. For low noise levels both solutions happen to coincide, but for $\sigma = 10^{-2}$ and $\sigma = 10^{-1}$ the solution diverges and the residual remains similar (see Fig. 30).

From Fig. 32 no significant differences are observed in the Pareto curves with respect to the previous examples. The curves are well-defined and the corner points match the optimal ones. The regularization parameters given by CV only coincide with the optimal ones for the ω_1 state. Similarly, the exact and predicted trajectories agree until $t = 50$ (Fig. 33).

6. Discussion and conclusion

In summary, the motivation of this work was to improve the accuracy and stability of sparse regression techniques, a.k.a SINDy, for the identification of governing equations of nonlinear dynamical systems from noisy

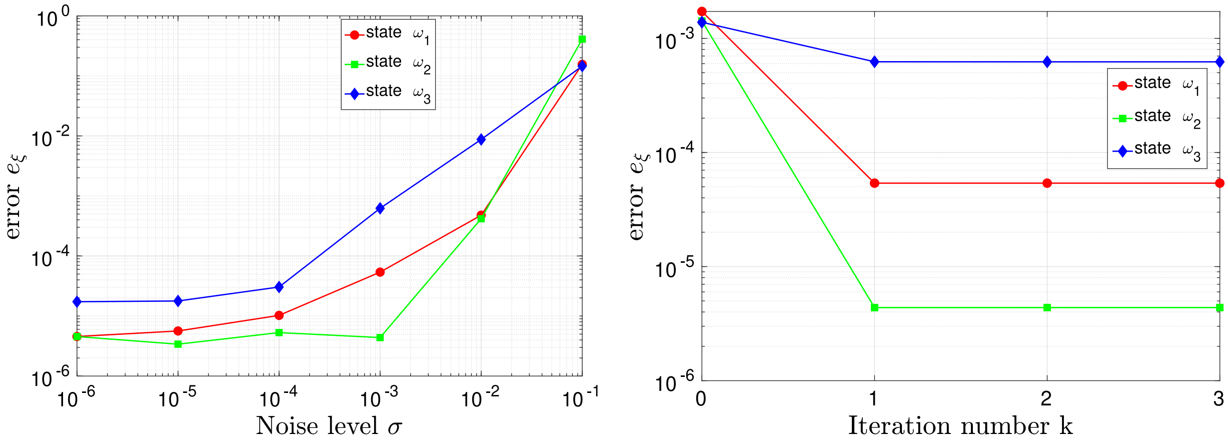

Fig. 30. Left: Relative solution error of the Euler rigid body equations with respect to different noise levels. Right: Relative solution error with respect to iteration k at noise level $\sigma = 10^{-3}$.

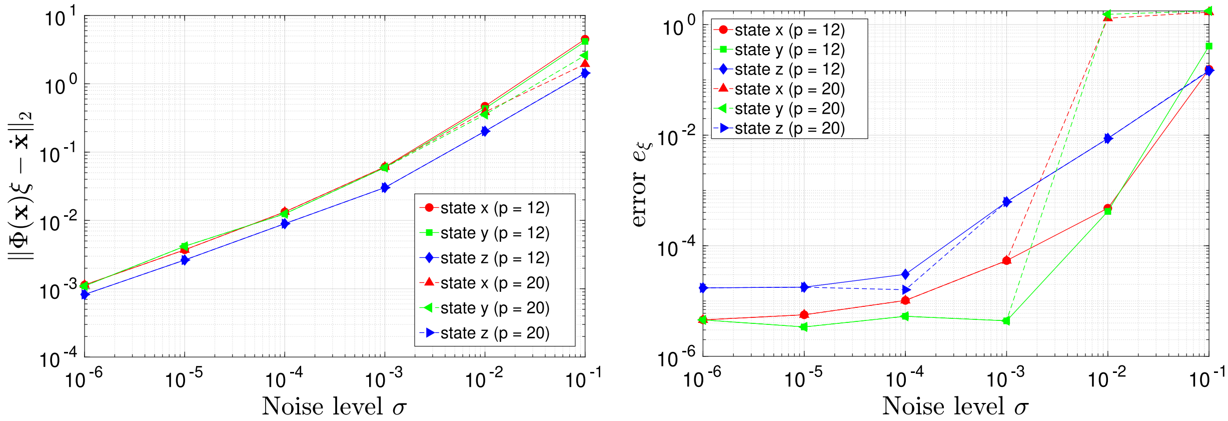

Fig. 31. Left: Residual error at different noise levels for $p = 12$ and $p = 20$. Right: Relative solution error at different noise levels for $p = 12$ and $p = 20$. $p = 12$ represents the number of columns in $\mathbf{\Phi}(\mathbf{x})$ after removing the linear dependent ones, whereas $p = 20$ corresponds to $\mathbf{\Phi}(\mathbf{x})$ keeping all the columns.

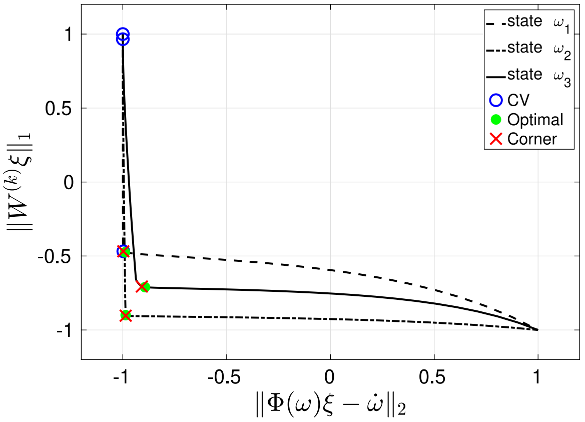

Fig. 32. Pareto curves for each state at noise level $\sigma = 10^{-3}$ and iteration 0.

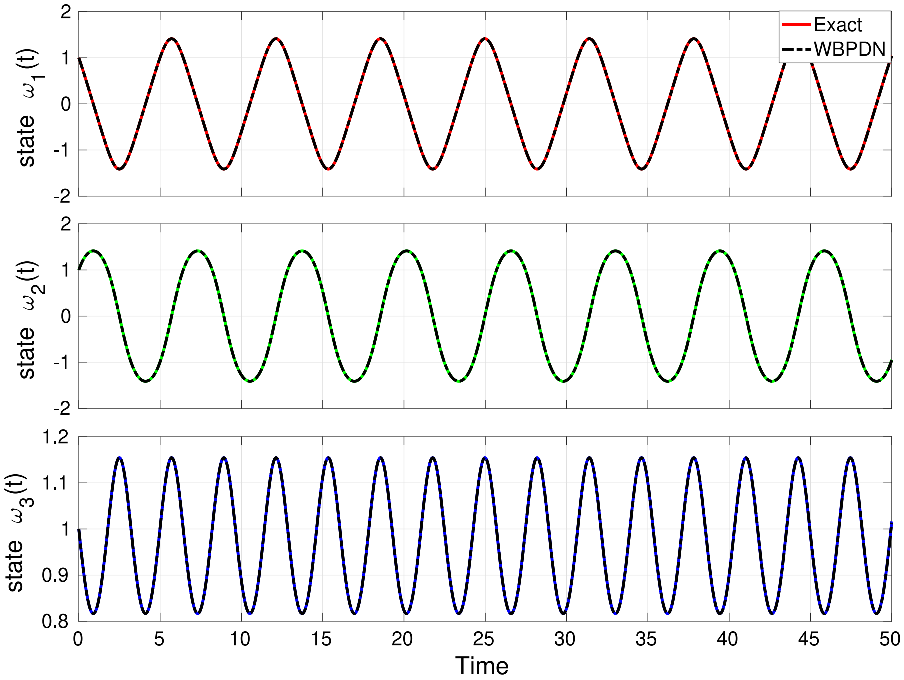

Fig. 33. Exact and predicted trajectories of the Euler rigid body dynamics for $\sigma = 10^{-3}$.

state measurements. As in SINDy, the governing equations are approximated in a multivariate basis of state variables, here of monomial type. To identify the expansion coefficients, we proposed a strategy based on the iteratively reweighted Basis Pursuit Denoising (WBPDN) or Adaptive LASSO techniques, previously used in the context of compressed sensing and sparse regression. Penalizing the weighted ℓ_1-norm of the coefficients via solution-dependent weights enhances sparsity and the stability of the regression problem. The selection of the regularization parameter balancing the sparsity of the recovered coefficients and the proximity of the model to data – the time derivatives of the state variables – was done using the Pareto curve criterion. We observed that the corner point of the Pareto curve yielded near optimal selection of regularization parameters, and that the Pareto curve criterion consistently outperforms the commonly used K-fold cross-validation. For the range of noise levels we used, the Pareto curves showed well-defined corner points that agreed with regularization parameters that provided minimum solution errors. We demonstrated empirically that the WBPDN approach outperformed Sequentially Thresholded Least Squares (STLS) and Sequential Thresholded Ridge Regression (STRidge) techniques previously utilized within the SINDy framework.

We also addressed the recovery of a single constraint on dynamical systems, e.g., conservation of total energy, given by an implicit function of the state variables. This was achieved by generating the interpolative decomposition (ID) of the sampled measurement matrix used to approximate the governing equations. We illustrated the utility of ID in recovering a single constraint for small levels of state measurement noise.

In light of the effectiveness of WBPDN for recovering governing equations of nonlinear dynamical systems, it is desirable to extend this work to high-dimensional Partial Differential Equations (PDE) along with novel model reduction techniques to learn the dynamics of multi-physics or multi-scale systems. Robust identification of physical constraints from data – specifically in the presence of large noise levels – or the imposition of physical constraints to inform the identification of the governing equations are other directions to further investigate in future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Science Foundation grant CMMI-1454601.

Appendix. Estimating derivatives via Tikhonov regularization

The discrete derivatives are computed at the midpoints of (t_i, t_{i+1}) using the midpoint integration rule

$$x_{i+1} = x_i + \dot{x}_{i+1/2}\,\Delta t, \quad i = 1, \ldots, m-1. \tag{A.1}$$

The discretization (A.1) yields the linear system

$$\mathbf{A}\dot{\mathbf{x}} = \hat{\mathbf{x}}, \tag{A.2}$$

where $\mathbf{A} \in \mathbb{R}^{(m-1)\times(m-1)}$ is the discrete integral operator, $\dot{\mathbf{x}} \in \mathbb{R}^{m-1}$ are the approximate derivatives at the midpoints, and $\hat{\mathbf{x}} = \{x_j - x_1\}_{j=2}^{m} \in \mathbb{R}^{m-1}$. Since noisy state variables may amplify the error in the derivatives, we regularize (A.2) by introducing a differential operator $\mathbf{D}$. The regularized problem then becomes

$$\min_{\dot{\mathbf{x}}} \|\mathbf{A}\dot{\mathbf{x}} - \hat{\mathbf{x}}\|_2^2 + \alpha\|\mathbf{D}\dot{\mathbf{x}}\|_2^2, \tag{A.3}$$

where α is a non-negative parameter that controls the smoothness of the solution $\dot{\mathbf{x}}$. In (A.3), $\mathbf{D}$ is defined as

$$\mathbf{D} = \begin{bmatrix}\mathbf{I}, & \mathbf{D}_1, & \mathbf{D}_2\end{bmatrix}^T \in \mathbb{R}^{(3m-6)\times(m-1)}, \tag{A.4}$$

where $\mathbf{I}$ denotes the identity matrix, and $\mathbf{D}_1$ and $\mathbf{D}_2$ are the discrete first and second difference operators. The minimization problem (A.3) is convex, differentiable, and admits a closed-form solution. The α parameter modulates the balance between the fidelity to the data and the smoothness of the derivative. For a given value of α, the optimal solution is given by

$$\dot{\mathbf{x}}_\alpha = (\mathbf{A}^T\mathbf{A} + \alpha\mathbf{D}^T\mathbf{D})^{-1}\mathbf{A}^T\hat{\mathbf{x}}. \tag{A.5}$$

Once approximations of the state time derivatives are generated, we recover the derivatives at the sample points t_i, $i = 1, \ldots, m$, using cubic interpolation. We found that Tikhonov regularization differentiation was robust, computationally efficient and easy to implement. We remark that the accuracy of the state derivatives depends on the sample size m, sampling rate Δt and level of noise σ in the data. Assuming the Nyquist–Shannon sampling theorem [59] is satisfied and the noise is zero-mean independent and identically distributed, the midpoint integration yields a discrete approximation error (bias) $\sim \mathcal{O}(\Delta t^2)$ and noise amplification (variance) that scales as $\sim \mathcal{O}(\frac{\sigma^2}{\Delta t^2})$ [60]. In our numerical experiments, we observed that the condition number of $\mathbf{A}$ scales linearly with the sample size; the larger the condition number, the larger the least squares error may be. Tikhonov regularization attempts to lower the condition number of $\mathbf{A}$ and reduce the effect of noise by appropriately selecting α. However, it introduces bias in the computed solutions [61].

In our experience, the L-curve criterion for choosing α yields the best results. In the case of Tikhonov regularization, Regińska [62] proves that the log–log L-curve is always strictly convex at its ends for $\alpha \le \sigma_n$ and $\alpha \ge \sigma_1$ (σ_1 and σ_n being the largest and smallest singular values of $\mathbf{A}$, respectively). Let $\mathbf{u}_i$, $i = 1, \ldots, m-1$, denote the right singular vectors of $\mathbf{A}$ associated with the ith largest singular value of $\mathbf{A}$. The L-curve can be concave if $|\mathbf{u}_i^T\hat{\mathbf{x}}|$ are monotonic with respect to i or are constant [63].

Remark. Numerical differentiation may produce inaccurate results around the boundaries of a function if boundary conditions are unknown. This is the case in the present application where the time derivatives of the state variables are not known at the ends of the training time period. As such, we here compute the time derivatives over a time interval that is 5% larger than the intended training time span (from each side), but use the state data and estimated time derivatives over the original training time span.

References

[1] S. Fassois, D. Rivera, Applications of system identification, IEEE Control Syst. Mag. 27 (2007) 24–26.

[2] L. Ljung, System identification, Wiley Encyclopedia of Electrical and Electronics Engineering.

[3] L. Ljung, Perspectives on system identification, Annu. Rev. Control 34 (1) (2010) 1–12.

[4] M.A. Balikhin, R.J. Boynton, S.N. Walker, J.E. Borovsky, S.A. Billings, H.-L. Wei, Using the narmax approach to model the evolution of energetic electrons fluxes at geostationary orbit, Geophysical Research Letters 38 (18).

[5] R. Boynton, M. Balikhin, S. Billings, G. Reeves, N. Ganushkina, M. Gedalin, O. Amariutei, J. Borovsky, S. Walker, The analysis of electron fluxes at geosynchronous orbit employing a narmax approach, J. Geophys. Res.: Space Phys. 118 (4) (2013) 1500–1513.

[6] R. Mahnken, Identification of material parameters for constitutive equations, in: Encyclopedia of Computational Mechanics, second ed., 2017, pp. 1–21.

[7] S.L. Brunton, B.R. Noack, Closed-loop turbulence control: progress and challenges, Appl. Mech. Rev. 67 (5).

[8] M. Morari, J.H. Lee, Model predictive control: past, present and future, Comput. Chem. Eng. 23 (4–5) (1999) 667–682.

[9] I. Leontaritis, S.A. Billings, Input–output parametric models for non-linear systems part i: deterministic non-linear systems, Internat. J. Control 41 (2) (1985) 303–328.

[10] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, Science 324 (5923) (2009) 81–85.

[11] J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 104 (24) (2007) 9943–9948.

[12] W.X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, C. Grebogi, Predicting catastrophes in nonlinear dynamical systems by compressive sensing, Phys. Rev. Lett. 106 (15) (2011) 154101.

[13] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 113 (15) (2016) 3932–3937.

[14] B. Duignan, Occam's razor, (2018).

[15] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1) (1996) 267–288.

[16] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al., Least angle regression, Ann. Statist. 32 (2) (2004) 407–499.

[17] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (1) (2001) 129–159.

[18] S. Zhang, G. Lin, Robust data-driven discovery of governing physical laws with error bars, Proc. R. Soc. A 474 (2217) (2018) 20180305.

[19] K. Kaheman, J.N. Kutz, S.L. Brunton, Sindy-pi: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics, arXiv preprint arXiv:2004.02322.

[20] G. Tran, R. Ward, Exact recovery of chaotic systems from highly corrupted data, Multiscale Model. Simul. 15 (3) (2017) 1108–1129.

[21] H. Zou, The adaptive lasso and its oracle properties, J. Amer. Statist. Assoc. 101 (476) (2006) 1418–1429.

[22] E.J. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, J. Fourier Anal. Appl. 14 (5) (2008) 877–905.

[23] X. Yang, G.E. Karniadakis, Reweighted ℓ_1 minimization method for stochastic elliptic differential equations, J. Comput. Phys. 248 (2013) 87–108.

[24] J. Peng, J. Hampton, A. Doostan, A weighted ℓ_1-minimization approach for sparse polynomial chaos expansions, J. Comput. Phys. 267 (2014) 92–111.

[25] B. Adcock, Infinite-dimensional ℓ^1 minimization and function approximation from pointwise data, Constr. Approx. 45 (3) (2017) 345–390.

[26] H. Cheng, Z. Gimbutas, P.G. Martinsson, V. Rokhlin, On the compression of low rank matrices, SIAM J. Sci. Comput. 26 (4) (2005) 1389–1404.

[27] H. Schaeffer, G. Tran, R. Ward, Extracting sparse high-dimensional dynamics from limited data, SIAM J. Appl. Math. 78 (6) (2018) 3279–3295.

[28] K. Wu, D. Xiu, Numerical aspects for approximating governing equations using data, J. Comput. Phys. 384 (2019) 200–221.

[29] H. Miao, X. Xia, A.S. Perelson, H. Wu, On identifiability of nonlinear ODE models and applications in viral dynamics, SIAM Rev. 53 (1) (2011) 3–39.

[30] S. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, An interior-point method for large-scale ℓ_1-regularized least squares, IEEE J. Sel. Top. Sign. Proces. 1 (4) (2007) 606–617.

[31] N.M. Mangan, J.N. Kutz, S.L. Brunton, J.L. Proctor, Model selection for dynamical systems via sparse regression and information criteria, Proc. R. Soc. A 473 (2204) (2017) 20170009.

[32] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial differential equations, Sci. Adv. 3 (4) (2017) e1602614.

[33] M. Quade, M. Abel, J. Nathan Kutz, S.L. Brunton, Sparse identification of nonlinear dynamics for rapid model recovery, Chaos 28 (6) (2018) 063116.

[34] A. Bruckstein, D. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, SIAM Rev. 51 (1) (2009) 34–81.

[35] A. Doostan, H. Owhadi, A non-adapted sparse approximation of PDEs with stochastic inputs, J. Comput. Phys. 230 (2011) 3015–3034.

[36] J. Hampton, A. Doostan, Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies, J. Comput. Phys. 280 (2015) 363–386.

[37] E.J. Candès, The restricted isometry property and its implications for compressed sensing, C. R. Math. 346 (9) (2008) 589–592.

[38] H. Rauhut, R. Ward, Sparse legendre expansions via ℓ_1-minimization, J. Approx. Theory 164 (5) (2012) 517–533.

[39] J. Peng, J. Hampton, A. Doostan, On polynomial chaos expansion via gradient-enhanced ℓ_1-minimization, J. Comput. Phys. 310 (2016) 440–458.

[40] A. Cohen, M.A. Davenport, D. Leviatan, On the stability and accuracy of least squares approximations, Found. Comput. Math. 13 (5) (2013) 819–834.

[41] J. Hampton, A. Doostan, Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression, Comput. Methods Appl. Mech. Eng. (0).

[42] M. Hadigol, A. Doostan, Least squares polynomial chaos expansion: A review of sampling strategies, Comput. Methods Appl. Mech. Engrg. 332 (2018) 382–407.

[43] P. Hansen, Analysis of discrete ill-posed problems by means of the l-curve, SIAM Rev. 34 (4) (1992) 561–580.

[44] E. Van den Berg, M. Friedlander, Probing the pareto frontier for basis pursuit solutions, SIAM J. Sci. Comput. 31 (2) (2009) 890–912.

[45] C. Hansen, T.K. Jensen, G. Rodriguez, An adaptive pruning algorithm for the discrete l-curve criterion, J. Comput. Appl. Math. 198 (2) (2007) 483–492.

[46] Alessandro Cultrera, Luca Callegaro, A simple algorithm to find the l-curve corner in the regularisation of ill-posed inverse problems, IOP SciNotes 1 (2) (2020) 025004.

[47] J.N. Tehrani, A. McEwan, C. Jin, A. Van Schaik, L1 regularization method in electrical impedance tomography by using the l1-curve (pareto frontier curve), Appl. Math. Model. 36 (3) (2012) 1095–1105.

[48] W.S. Cleveland, Robust locally weighted regression and smoothing scatterplots, J. Amer. Statist. Assoc. 74 (368) (1979) 829–836.

[49] J. Cullum, Numerical differentiation and regularization, SIAM J. Numer. Anal. 8 (2) (1971) 254–265.

[50] R. Chartrand, Numerical differentiation of noisy, nonsmooth data, ISRN Appl. Math. (2011).

[51] I. Knowles, R.J. Renka, Methods for numerical differentiation of noisy data, Electron. J. Differential Equations 21 (2014) 235–246.

[52] G. Golub, C. Van Loan, Matrix computations, Baltimore, MD..

[53] T. Kariya, H. Kurata, Generalized Least Squares, John Wiley & Sons, 2004.

[54] D. Donoho, I. Drori, V. Stodden, Y. Tsaig, Sparselab: Seeking sparse solutions to linear systems of equations, 2019, URL http://sparselab.stanford.edu.

[55] D.L. Donoho, V.C. Stodden, Y. Tsaig, About sparselab.

[56] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (2) (1963) 130–141.

[57] B. Van der Pol, A theory of the amplitude of free and forced triode vibrations, Radio Rev. 1 (1920) 701–710, 754–762.

[58] V.A. Morozov, On the solution of functional equations by the method of regularization, in: Doklady Akademii Nauk, Vol. 167, Russian Academy of Sciences, 1966, pp. 510–512.

[59] C.E. Shannon, Communication in the presence of noise, Proc. IRE 37 (1) (1949) 10–21.

[60] J.C. Spall, Introduction To Stochastic Search and Optimization: Estimation, Simulation, and Control, 65, John Wiley & Sons, 2005.

[61] P. Hansen, Discrete Inverse Problems, Society for Industrial and Applied Mathematics, 2010.

[62] T. Regińska, A regularization parameter in discrete ill-posed problems, SIAM J. Sci. Comput. 17 (3) (1996) 740–749.

[63] P.C. Hansen, The l-curve and its use in the numerical treatment of inverse problems.