Computers and Chemical Engineering 157 (2022) 107628

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Handling noisy data in sparse model identification using subsampling and co-teaching

Fahim Abdullah[a], Zhe Wu[b], Panagiotis D. Christofides[a,c,*]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA
[b] Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore
[c] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA

ARTICLE INFO

Article history:
Received 25 October 2021
Revised 20 November 2021
Accepted 3 December 2021
Available online 7 December 2021

Keywords:
Nonlinear processes
Sparse identification
Chemical processes
Subsampling
Co-teaching

ABSTRACT

In this paper, a novel algorithm based on sparse identification, subsampling and co-teaching is developed to mitigate the problems of highly noisy data from sensor measurements in modeling of nonlinear systems. Specifically, sparse identification is combined with subsampling, a method where a fraction of the data set is randomly sampled and used for model identification, as well as co-teaching, a method that mixes noise-free data from first-principles simulations with the noisy measurements to provide a mixed data set that is less corrupted with noise for model training. The proposed method is bench-marked against sparse identification without subsampling as well as subsampling but without co-teaching using two examples, a predator-prey system and a chemical process, both of which are modeled as nonlinear systems of ordinary differential equations. It was shown that the proposed method yields better models in terms of prediction accuracy in the presence of high noise levels.

1. Introduction

Historically, a key focus of research in the fields of science and engineering has been the discovery of physical laws in the form of governing equations. In recent years, however, the focus has shifted to data-driven discovery of these laws. These fundamental laws are often in the form of dynamical models *i.e.,* ordinary differential equations (ODE) or partial differential equations (PDE) in time. Examples include the Maxwell equations from electromagnetism, Boltzmann equation from thermodynamics, Navier-Stokes equations from momentum transfer, Black-Scholes equation from finance, and predator-prey equations from biology (Zhang and Lin, 2018). In other words, the laws are discovered as time-series predictive models, which are a necessary building block in many engineering applications, from predictive maintenance in industrial engineering to advanced control systems such as model predictive control (MPC) that are widely found in chemical process systems. MPC requires a dynamical model to predict the states and/or outputs of the process over a prediction horizon. As a result, considerable work on data-driven modeling can be found in the context of MPC (Abdullah et al., 2021a; 2021b; Aggelogiannaki and Sarimveis, 2008; Al Seyab and Cao, 2008; Aumi et al., 2013; Chaffart and Ricardez-Sandoval, 2018; Garg and Mhaskar, 2018; Xie et al., 2015; Zeng et al., 2010; Wu et al., 2021a; 2021b). Some of the data-driven system identification methods developed and investigated in the recent literature include singular value decomposition (Moore, 1986), Numerical algorithms for Subspace State Space System Identification (N4SID) (Van Overschee and De Moor, 1994), and auto-regressive models with exogenous inputs (ARX/ARARX) (Huusom et al., 2012; Menezes and Barreto, 2008; Siegelmann et al., 1997). Due to the exponential increase in computational power over the past decade, machine learning methods, a subset of data-driven modeling methods, have produced remarkable results when utilized to their fullest extents due to their ability to capture complex, interacting nonlinearities by tuning numerous hyper-parameters (Ali et al., 2015; Kosmatopoulos et al., 1995; Trischler and D'Eleuterio, 2016; Wong et al., 2018). Machine learning methods are a broad class of data-driven modeling methods including linear regression, support vector machines, deep neural networks, sparse identification, *etc.* (Brunton et al., 2016). The primary method investigated and improved in this article is sparse identification. While sparse identification has already been developed in many facets (Mangan et al., 2016a; Dam et al., 2017; Schaeffer et al., 2018; Tran and Ward, 2017; Kaiser et al., 2018; Mangan et al., 2019; Boninsegna et al., 2018; Schaeffer et al., 2020; Loiseau and Brunton, 2018; Zhang and Schaeffer, 2019), the aspect of noisy data remains largely a challenge for the method.

* Corresponding author. fax: +1 (310) 206 4107.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.compchemeng.2021.107628
0098-1354/

Since machine learning methods have mostly been developed in the domain of computer science, they often either assume the availability of high-fidelity data or use the term "noisy data" to refer to mislabels in classification problems rather than numerical inaccuracies in regression problems (Han et al., 2018). Hence, the reported accuracy and success of these methods stem from application of the algorithms to standard, clean data sets. However, in the field of engineering, particularly chemical engineering, the availability of noise-free data remains a challenge. Most engineering systems contain at least sensor and measurement noise, if not disturbances. Consequently, when machine learning methods are applied to noisy data from process engineering systems, the results may be unexpectedly inferior. To mitigate the subpar performance of traditional system identification methods when using noisy data, a significant amount of effort surfaces when the recent literature is reviewed. Examples include the extension of the ARX and ARARX methods to input/output data with additive white noise (Diversi et al., 2010), estimation of the noise term using methods involving principal component analysis (Wu et al., 2015), applying subspace identification methods to closed-loop operation data (Juricek et al., 1998), and the Kalman filter for linear dynamical systems with Gaussian white noise (Patwardhan et al., 2012; Yeo and Melnyk, 2019). For nonlinear systems, methods investigated in the literature include the extended Kalman filter and moving horizon estimation (Patwardhan et al., 2012). However, the methods described incorporate numerous assumptions regarding the structure of the system and noise distribution, limiting their applicability to industry (Patwardhan et al., 2012). As a result, the field of data-driven process modeling for dynamical systems using noisy sensor data requires further innovation and improvement to be implemented in practice.

Although the initial paper, Brunton et al. (2016) remarked that the total-variation regularized derivative is robust to noise, a deeper investigation into the levels of noise as well as the data generation and sampling details reveals that the claim cannot be broadened to every case, or even many practical cases. Under high levels of noise, the calculation of the model parameters in sparse identification suffers greatly. This is due to the dynamics possibly being sparse only in a non-orthonormal basis, such as monomials, or due to the sampling of the variables following the system dynamics instead of being random or experimentally designed, which can lead to ill-conditioning measurement data matrices (Hadigol and Doostan, 2018). Due to these two issues in the measurement data matrix, the matrix may generally violate the incoherence property (Bruckstein et al., 2009; Doostan and Owhadi, 2011; Hampton and Doostan, 2015b) or the restricted isometry property (Candès, 2008; Rauhut and Ward, 2012; Peng et al., 2016) in the underdetermined problem, or not satisfy the incoherence property (Cohen et al., 2013; Hampton and Doostan, 2015a) in the overdetermined problem. Therefore, a number of investigations have been carried out to explicitly detail the impact of noise on sparse identification, and several extensions have been proposed to the original sparse identification algorithm to account for noise and other practical concerns. In Nguyen et al. (2020), the issues of both noise and irregularly sampled data are addressed by using an assimilation step with an autoencoder or ensemble Kalman filter before the model-identification step to use recovered states in the identification step. Partially available data and noise have also been investigated in Didonna et al. (2019), although the noise levels considered are relatively low with a signal-to-noise ratio (SNR) of 22.33. In de Silva et al. (2020), the discrepancy between the true governing equations and the sparse-identified models due to noise is addressed by exploiting group sparsity, where partial knowledge of the underlying physics is used to group terms together and ensure all or none of the terms appear in the model. de Silva et al. (2020) used a smoothed finite-difference approach using the Savitzky-Golay filter, as will be used in this paper, and deeply studied the effect of most of the parameters and possible variables in the derivative estimation step. de Silva et al. (2020) provides the highest level of detail regarding the numerical differentiation procedure used in this paper. In summary, it was shown that smoothing the data generally improves performance, changing the type of finite-difference or smoothing weakly affect the results at sufficiently high noise levels, and that the default window used in the filter is appropriate for highly noisy data. These results further demonstrate the necessity to develop novel methods to extend sparse identification to the case of highly noisy data. Quade et al. (2018) proves that when sparse identification is used to update an existing model in real-time by using only new data as it becomes available, the algorithm is less susceptible to noise than re-identifying a new model. The focus of Quade et al. (2018) was on real-time model updates, which are highly applicable to many practical problems. Hence, the first and perhaps most important step of identifying a model from the bulk of the raw data collected did not include any noise. Hence, the derivative estimation was carried out using a simple first-order forward finite-difference routine, which causes no significant inaccuracies or even numerical instabilities. However, the effect of noise was studied in the sense that noise was added to the new data as it becomes available, which were used to update the model in real-time. However, as shown in their results, the amount of new data collected to update a model is much less than the amount of total data used to initially identify an accurate model. Furthermore, in this step, the noise was added directly to the derivative rather than using the forward finite-difference scheme used in the main data set. Due to these two factors, it is ambiguous what the effect of the new noisy data would be if a finite-difference method was used to compute the derivatives, or if the original data itself was corrupted with noise. Leylaz et al. (2021) uses an "algebraic operation", specifically Laplace transforms and inverse Laplace transforms, to reformulate the ODE model using integral terms, which mitigates noise in successive operations. In Lin et al. (2021), the computation of the second derivative in mass-spring systems is avoided by using the Duhamel's integral, while further de-noising is proposed by means of the RKHS (Reproducing Kernel Hilbert Space)-based non-parametric de-noise method. Sparse identification has also been assisted by the manifold boundary approximation method in Sarić et al. (2021) to extend it to power systems, which involve differential algebraic equations (DAE), stiff ODEs, and low measurement noise. Although the application was to PDEs rather than ODEs, Schaeffer (2017) studied a large number of example systems using sparse identification and employed spectral methods/filtering (Hesthaven et al., 2007) to estimate the derivatives.

To deal with implicit ODEs as well as noise, sparse identification was extended in Mangan et al. (2016b). However, the method suffered from an explosive increase in the sensitivity to noise. This was subsequently addressed and improved in Kaheman et al. (2020) via parallel processing and multiple optimization algorithms, making the new method, SINDy-PI, orders of magnitude more robust to noise than before. However, from the case studies in Kaheman et al. (2020), the improved method still failed at noise levels above variances of 10^{-4}, which is very low for practical purposes. Furthermore, as the method is implicit, the derivative terms and their interactions must all be included in the function library for this method, which can greatly expand the function space. Hence, even more caution is required in building the library when using SINDy-PI. While the increase in computational expense is offset by parallel processing, the additional infrastructure required for parallelization might not be available in many applications.

Two papers, Cortiella et al. (2021) and Zhang and Lin (2021), have pioneered the application of sparse identification to noisy data. In the first, Cortiella et al. (2021) modified the original L_1-regularized sequential least squares algorithm from Brunton et al. (2016) by instead using a reweighted L_1-regularized sequential least squares algorithm for the optimization and the second-order Tikhonov regularization for the derivative approximation. The regularization parameter was determined using a Pareto curve. The other paper, Zhang and Lin (2021), proposed a subsampling-based threshold sparse Bayesian regression or SubTSBR, where a fraction of the entire data set is randomly subsampled a number of times and the best model selected using a model-selection criterion.

In all of the papers mentioned above, with the exception of two, the derivatives were calculated either using the ODE in the data generation step with noise subsequently added to it, or the derivatives were estimated using numerical methods from the exact data generated and noise was added to both the data and computed derivatives afterwards. The only exceptions were Schaeffer (2017) and Cortiella et al. (2021). In Schaeffer (2017), except the final example, the remaining examples were also carried out with noise being added directly to the derivatives rather than the data sets themselves. However, in the final example, it was noted that the underlying assumption of such an estimation methodology is that the derivatives themselves are more corrupted by the noise than the original data, which restricted the scope of application of the method. Therefore, attempting to identify the governing PDEs with only noisy data and estimating derivatives from the data was attempted. It was demonstrated that, even with presmoothing and spectral filtering, the method worked until a noise level of 50% and failed at 100%, the latter of which is the low level of noise considered in this work. In Cortiella et al. (2021), despite using Tikhonov regularization and reweighted L_1-regularized sequential least squares, the authors showed that the errors in both the derivative approximation and the solution increase rapidly in orders of magnitude when the noise level increases beyond $\sigma = 10^{-2}$, which is also very small for all practical purposes. Moreover, their main results for all their studies are based on data with a SNR of approximately 60. Furthermore, all the papers considered only zero-mean Gaussian white noise. To the best of our knowledge, in the context of sparse identification, the direct impact of noisy industrial data on the derivative computation as well as non-Gaussian noise have not been studied in detail. The method used to approximate the derivatives, as well as the optimizer that is used with the derivative approximator are crucial hyperparameters that require in-depth analysis as will be seen in Section 4.

Beyond the domain of sparse identification, another recent method to prevent overfitting in the presence of high noise in measured data that has received significant attention, developed primarily in the context of neural networks, is co-teaching, where noise-free data from first-principles simulations is used to assist the training process (Wu et al., 2021a; 2021b). The method proposed in this work combines both subsampling (Zhang and Lin, 2021) and co-teaching (Wu et al., 2021a; 2021b) with sparse identification to deal with even higher levels of noise than was previously possible to handle.

The rest of this manuscript is organized as follows: in Section 2, the class of nonlinear systems considered and the basic methods combined for the implementation of the proposed novel method are reviewed in brief. In Section 3, the novel method combining sparse identification with subsampling and co-teaching is introduced and described in detail. In Section 4, the proposed method is applied to a predator-prey example and a chemical reactor example to demonstrate its effectiveness, and the conclusions are summarized in Section 5.

2. Preliminaries

2.1. Class of systems

The class of continuous-time nonlinear systems considered in this work can be written in the form,

$$\dot{x}(t) = f(x(t)), \qquad x(t_0) = x_0 \tag{1a}$$

$$y = x + w \tag{1b}$$

where $x \in \mathbb{R}^n$ is the state vector, $y \in \mathbb{R}^n$ is the vector of sampled measurements of the states, $w \in \mathbb{R}^n$ is the sensor noise, and the unknown vector function $f(\cdot)$ is the process model representing the inherent physical laws constraining the system. Without loss of generality, the initial time t_0 is taken to be 0 throughout the article.

2.2. Sparse identification

Sparse identification is a relatively new method for identifying nonlinear systems based on data. It has been effectively deployed on a wide range of engineering systems of relevance (Wang et al., 2011; Schaeffer et al., 2013; Ozoliņš et al., 2013; Mackey et al., 2014; Brunton et al., 2014; Proctor et al., 2014; Bai et al., 2015). Provided with only sensor measurements from a system of the form of Eq. (1), sparse identification aims to reconstruct the system as a first-order differential equation of the form,

$$\dot{\hat{x}} = \hat{f}(\hat{x}) \tag{2}$$

where $\hat{x} \in \mathbb{R}^n$ is the vector of state of the sparse-identified model $\hat{f}(\cdot)$.

The central idea of sparse identification is to consider many possible nonlinear terms for the right-hand side of Eq. (2), $\hat{f}$, and subsequently identify the few active terms in $\hat{f}$. This apparent simplification stems from the fact that physical systems in practice only contain a small number of nonzero terms when considering a large set of terms *i.e.,* candidate basis functions. Hence, the space of all nonlinear basis functions considered is sparse, and efficient algorithms may be used to compute the pre-multiplying coefficients. To carry out sparse identification we first sample and collect a set of measurements of the states $x_1, x_2, \ldots, x_n$ at times $t_1, t_2, \ldots, t_m$ from open-loop simulations and concatenate them into the data matrix X,

$$X = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix} \tag{3}$$

where $x_i(t_j)$ denotes the measurement of state i at the j-th sampling time with $i = 1, \ldots, n$ and $j = 1, \ldots, m$. The time-derivative of X is represented by $\dot{X}$ and is a required quantity in the sparse identification procedure. However, in the presence of measurement noise, the estimation of the derivative is a challenge. Although some methods to robustly approximate the derivative are detailed in Brunton et al. (2016), particularly the total-variation regularized derivative from Rudin et al. (1992); Chartrand (2011), the results in this paper disagree in terms of its robustness and show that a smoothed finite difference can often yield better results. After acquiring X and $\dot{X}$, we build a function library, $\Theta(X)$, containing p candidate nonlinear functions of X corresponding to the p basis functions that may be active or inactive in the right-hand side of the ODE, $\hat{f}$. The sparse identification technique leverages sparsity to identify the active terms in this library, Θ. The augmented library, $\Theta(X)$, is optimized like a hyperparameter in this paper. Specifically, it is constructed using either only monomials up to

third-order including interaction terms or a combination of monomials, their interactions, and the four common trigonometric functions: *sin, cos, tan*, and *tanh*. Hence, the latter library containing all possible terms is of the form,

$$\Theta(X) = \begin{bmatrix} | & | & | & | & | & | & | & | \\ \mathbf{1} & X & X^{P_2} & X^{P_3} & \sin X & \cos X & \tan X & \tanh X \\ | & | & | & | & | & | & | & | \end{bmatrix} \tag{4}$$

In Eq. (4), X^{P_2} represents all quadratic nonlinearities:

$$X^{P_2} = \begin{bmatrix} x_1^2 & x_1x_2 & \cdots & x_2^2 & x_2x_3 & \cdots & x_n^2 \end{bmatrix} \tag{5}$$

The aforementioned choice of candidate basis functions is rooted in the observation that polynomials and trigonometric functions can be found in the natural laws governing many known physical systems (Brunton et al., 2016).

The objective of the sparse identification algorithm is to find the p coefficients that pre-multiply the p nonlinear basis functions considered in $\Theta(X)$ for each state x_i. Each x_i is associated with a corresponding sparse vector of coefficients, $\xi_i \in \mathbb{R}^p$, that characterize the nonzero terms in the respective ODE, $\dot{x}_i = f_i(x)$. Hence, n such coefficient vectors must be computed. In matrix notation, the quantity to be found is

$$\Xi = \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_n \end{bmatrix} \tag{6}$$

Therefore, to determine the matrix Ξ, the following equation needs to be solved:

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

Equation (7) is solved using sequential least-squares by zeroing all coefficients in Ξ that are smaller than a threshold λ, known as the sparsification knob, and repeatedly solving the resulting equation with zeroed terms until the non-zero coefficients converge. Due to the sparse structure of Ξ, convergence of the iterative step is rapid. Once Ξ is calculated, the overall model can be written as the continuous-time differential equation,

$$\dot{x} = \Xi^\top(\Theta(x^\top))^\top$$

where $\Theta(x^\top)$ is a column vector of symbolic functions of x from the function library, and $x^\top$ denotes the transpose of x.

2.3. Subsampling

Subsampling is a statistical technique where a subset of the entire data set is randomly selected for analysis instead of the entire data set. While the method is typically employed to estimate statistical metrics (Efron and Stein, 1981) or speed up an algorithm (Rudy et al., 2017), the objective of subsampling in this paper is to increase the modeling accuracy in the presence of high levels of sensor noise. Specifically, in regression, when the number of data points, m, is more than the number of unknown weights, as is the case in most practical problems, the regression might be carried out with only a subset of the entire data set to estimate the regression parameters. This may be considered because a fraction of the data points are corrupted by high noise levels or are outliers. Traditional regression methods like least squares use the entire data set by making the smoothing assumption, which states that the larger proportion of low-noise or "good" values will sufficiently smooth out any large noise in the data set. However, in practice, this assumption may not hold in the presence of either very high noise or consistently high noise throughout the data set as the fraction of "good" data points is inadequate to compensate the noise in these cases. The idea behind subsampling is to mitigate the problem of data points with large noise by randomly subsampling to eliminate them before carrying out the regression or model identification step.

2.4. Co-teaching

Co-teaching is a technique that improves the accuracy of a machine learning method by utilizing noise-free data sets from first-principles models and simulations of a physical process. Co-teaching originated in the field of classification problems, particularly image classification, where neural networks are abundantly used to classify images into user-defined categories. However, a fraction of the images can be miscategorized, which drastically reduces the data quality and, consequently, neural network accuracy if used without accounting for the mislabeled data samples. Such mislabels are termed "noisy labels" in machine learning.

Although the co-teaching method has primarily been developed in the context of classification problems and deep neural networks, there is nascent research into extending this method to regression problems using long short-term memory (LSTM) networks (Wu et al., 2021a; 2021b). Specifically, in Wu et al. (2021b), co-teaching was compared to a Monte-Carlo dropout LSTM network, which is an advanced form of the standard dropout neural network where weights are randomly omitted during the training process to improve generalization and minimize overfitting. Dropout layers may be considered as a regularization term in the context of neural networks. Carrying out regression on a subsampled fraction of a data set repeatedly is analogous to a standard dropout neural network where the omission of the weights occurs only in the training and not the testing phase. In contrast to dropout neural networks and subsampling, the key proposition of co-teaching is that the loss function value is significantly lower for noise-free data as compared to noisy data. Therefore, mixing any proportion of noise-free data with measured noisy data can guide the model training process to be more robust to noise and overfitting. This assertion was proven in Wu et al. (2021b), where the co-teaching LSTM outperformed even the Monte-Carlo dropout LSTM in terms of lower open-loop mean-square error as well as computational time (without parallel processing in either case).

3. Subsampling-based sparse identification with co-teaching

In this paper, the original sparse identification method is improved by combining it with subsampling and co-teaching as described in Section 2. The proposed method, Sparse Identification with Subsampling and Co-teaching (SISC), aims to mitigate the problem of highly noisy measurement data. When the measured data is either too noisy or consistently noisy, subsampling alone may not be sufficient as it will merely randomly choose the least noisy data in the subsamples, which can still be heavily corrupted by noise in this case, or a very large number of subsamples may be required to successfully isolate a relatively low-noise subsample, which can lead to an excessive computational expense. Therefore, in this paper, we also add noise-free data into each subsample to further improve the identification procedure.

In SISC (Algorithm 1), a subset of the total data set is randomly selected and mixed with noise-free data from first-principles simulations. The mixed data subset is used to estimate the model parameters. The resulting model is then evaluated using a model-selection criterion. After repeating the above steps for each subsample, the best model is selected to be the model with the lowest value of the model-selection criterion. Specifically, the algorithm is initiated with three user-defined parameters: the subsampling fraction $p \in (0, 1)$, the noise-free subsampling fraction $q \in (0, 1)$, and the number of times $L \geq 1$ to subsample and identify a model. For each of the L subsamples, a fraction containing $p \times m$ randomly selected data points from the original data set is mixed with $q \times m$ randomly selected data points from the noise-free data set. The resulting data submatrix for subsample i, where $i = 1, 2, \ldots, L$, is of

Algorithm 1: Sparse Identification with Subsampling and Co-teaching (SISC).

Input: X, Θ, p, q, L
Output: Ξ
compute exact $\dot{X}$ of noise-free data using first-principles ODE model;
compute estimate of $\dot{X}$ of noisy data using SFD or TVRD;
for $i \leftarrow 1$ **to** L **do**
  randomly sample p fraction of industrial (noisy) data and its estimated derivative;
  randomly sample q fraction of first-principles simulation (noise-free) data and its exact derivative;
  mix and concatenate the data into X_i and $\dot{X}_i$;
  calculate function library $\Theta(X_i)$ using only polynomial functions or a combination of polynomial and trignonometric functions;
  solve $\dot{X}_i = \Theta(X_i)\Xi_i$ using STLSQ or SR3 with different values of λ to get Ξ_i;
  with calculated Ξ_i, integrate ODE and compute AIC;
end
Let final $R = \arg\min_i\{\text{AIC}_i\}$;
Let $\Xi = \Xi_R$.

the form

$$X_i = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_{p\times m+q\times m}) & x_2(t_{p\times m+q\times m}) & \cdots & x_n(t_{p\times m+q\times m}) \end{bmatrix} \tag{8}$$

The equation to be solved for each subsample is, instead of Eq. (7),

$$\dot{X}_i = \Theta(X_i)\Xi_i \tag{9}$$

where Ξ_i is the coefficient matrix that needs to be computed. Each computed Ξ_i yields an ODE model,

$$\dot{x} = \Xi_i^\top(\Theta(x^\top))^\top \tag{10}$$

which is simulated and evaluated using the model-selection criterion against a portion of the data that is reserved for validation. In this work, the Akaike Information Criterion (AIC) is used as the model-selection criterion since it is calculated using not only the mean-squared error (MSE) but also the number of terms in the model, promoting sparsity by penalizing models with a higher number of terms. The AIC is defined in Mangan et al. (2017) as

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(x(t_i) - \hat{x}(t_i)\right)^2 \tag{11}$$

$$\text{AIC} = m\log\text{MSE} + 2L_0 \tag{12}$$

where L_0 is the 0-th norm, which is equal to the number of non-zero terms in the sparse-identified model.

The hyperparameters to be tuned in the algorithm include the function library to be constructed, the method chosen to compute the derivative from noisy data, the optimizer, and the value of the sparsification knob λ. The function library contains either only polynomial functions or both polynomial and trignonometric functions. The derivatives are approximated using either the total variation regularized derivative (TVRD) or smoothed finite-difference (SFD) after presmoothing with the Savitzky-Golay filter. The optimization is carried out using sequential thresholded least squares (STLSQ) or SR3, which is an enhancement of the least absolute shrinkage and selection operator (LASSO) (Zheng et al., 2019). To find a suitable value of λ, values of 0.1, 0.2, and 0.3 were tested. A detailed justification for the choices of λ is given in Section 4.2.1. Once all the hyperparameters have been tuned, a model is selected based on the AIC value of the validation set. Finally, out of the L models obtained from L subsamples after tuning all hyperparameters, the sparse-identified model with the lowest value of the AIC is taken to be the optimal model. The flow of the data through the algorithm is depicted in Fig. 1.

Remark 1. Both of the derivative approximation algorithms utilize certain parameters that may be tuned as well. For the Savitzky-Golay filter in SFD, the default values of a window length of 11 and cubic polynomials for curve-fitting were retained as they were satisfactory. For the TVRD, the regularization term was set to 0.001 instead of the default 0.01 and the maximum number of iterations with increased to 1000 to ensure convergence. Further tuning was infeasible due to the large number of hyperparameters to be trained in the main algorithm.

Remark 2. It is noted that, due to the issues of noise and sparse identification, some works have attempted to find alternatives to explanatory dynamical modeling instead. Such methods include black-box modeling using Runge-Kutta time-steppers embedding neural networks to account for nonlinearities (Fablet et al., 2018; González-García et al., 1998; Raissi et al., 2018; Rudy et al., 2019). It is noted that these papers propose alternative methods to nonlinear dynamical modeling rather than developing sparse identification further. Consequently, their techniques of dealing with noisy data as well as the results obtained in this context cannot be directly inferred to sparse identification, where noisy data directly and heavily affects the derivative estimation. For example, in Raissi et al. (2018), neural network function approximators are used in combination with classical linear multi-step time-stepping schemes (such as Runge-Kutta) to identify the nonlinear dynamic constraints in the right-hand-side of an ODE model, similar to the problem to be solved in sparse identification. However, when the effect of specifically the noise on the results is analyzed, the pattern is initially unexpected. As the noise level in the input data is increased, the results become more accurate and then less accurate. This was explained by neural networks, in particular, benefiting from small noise levels in the input, which work as regularization. At even higher noise levels, however, the results deteriorate due to the model not being able to disambiguate the noisy dynamics from the ground truth. Clearly, this is not the case for sparse identification and most system identification methods, where overfitting is not a common issue to the extent that data needs to be intentionally corrupted with noise to increase a model's generalization capabilities. Rudy et al. (2019) presents a similar but slightly more advanced work, where the data is decomposed into separate true dynamics and noisy dynamics, each of which are separately modeled and predicted when forecasting. The modeling is similar to Raissi et al. (2018) in terms of combining neural networks with time-stepping schemes, although constraints are added, making the overall problem more complex. The effect of noise and general pitfalls of neural networks are discussed near the end of their work, where, for example, the point is made that a neural network is, at its core, an interpolation technique. Hence, their proposed methods work best with systems where the dynamics evolve on an attractor, especially if it can be sampled with a small sampling time. The generated models also predict the trajectories to remain on the attractor for long simulation durations. If their method is used to integrate from new initial conditions or further from the attractor, the models may be insufficient. This is particularly important in the context of control, where they suggest collecting dynamic, transient data further from the attractor as perturbations and actuators must be considered. Schaeffer and McCalla (2017) reformulates the ODE model using integral terms, which can be approximated using piecewise con-

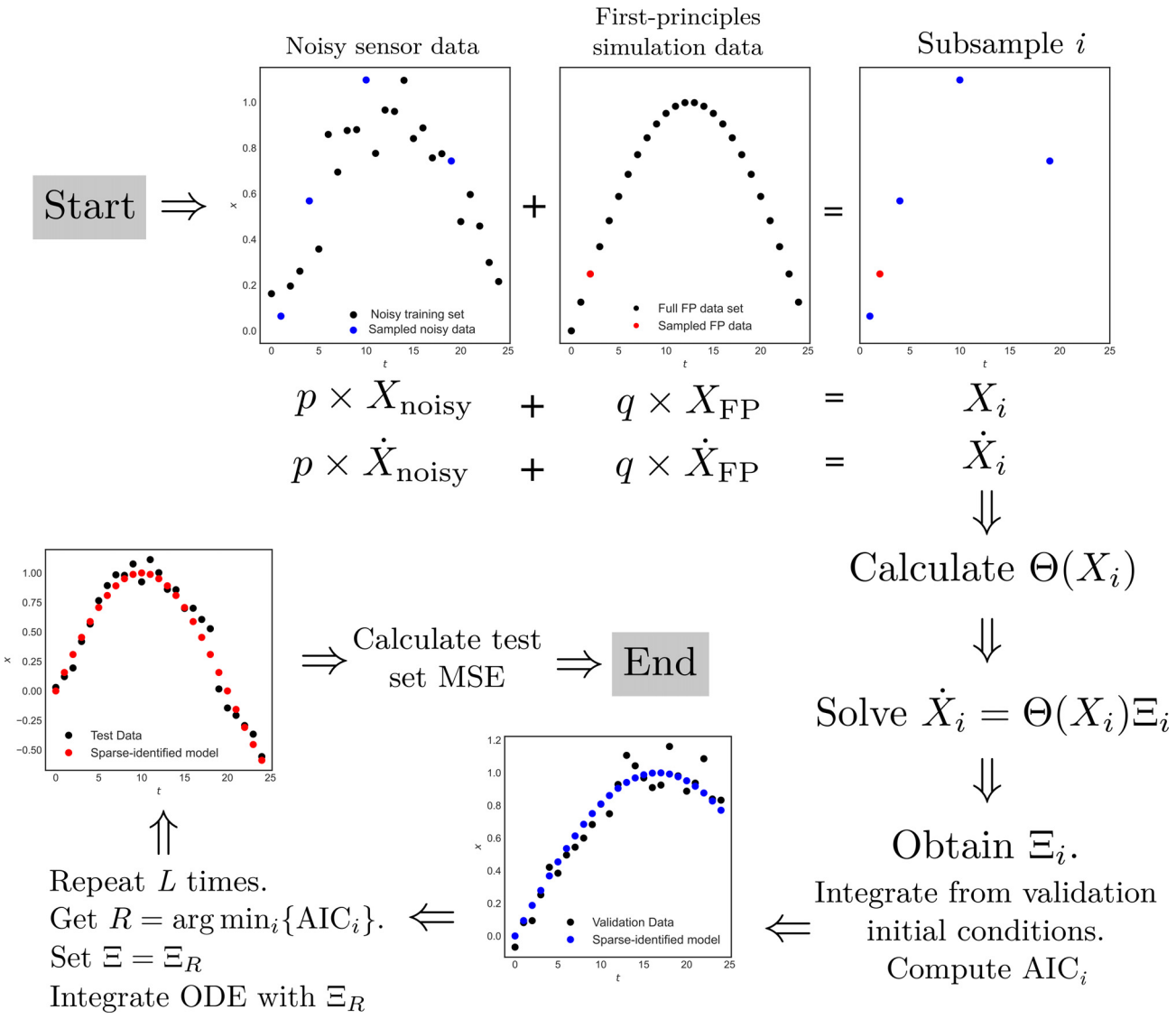

Fig. 1. Data flow diagram.

stant quadrature, and the resulting equations efficiently solved using the Douglas-Rachford algorithm. However, Schaeffer and McCalla (2017) mostly restricts its candidate functions for the model to polynomial terms and further notes that their method works better on fact and/or chaotic manifolds rather than slower, equilibrium behavior, which is more relevant in chemical process systems. A recent, novel method that has shown promise, especially when the data is corrupted by high noise or outliers, is entropic regression (AlMomani et al., 2020). We remark that these methods may be used instead of or in combination with our method.

Remark 3. While the focus of this work is the modeling of nonlinear dynamical systems, the ultimate objective is to incorporate the developed models into model predictive control (MPC) as the process model, which will be explored in a future work. Real-time optimization has been studied in several recent papers. Zhang et al. (2019) incorporated feedforward neural networks and first-principles process models into real-time optimization RTO and MPC. In the event of an increase in energy prices or in the presence of a large error in the feed concentration, it was shown that the performance of the controller with RTO outperforms the one without RTO. In Wu et al. (2020), a recurrent neural network (RNN) was used as the process model for a Lyapunov-based model predictive controller (LMPC) and a Lyapunov-based economic model predictive controller (LEMPC). The RNN model was updated in real-time based on event-triggers and/or error-triggers, which improved closed-loop performance in terms of stability, optimality, and smoothness of control actions. Hence, the extension of our proposed modeling approach to include both real-time optimization and model predictive control can be investigated in a future work after first extending it to solely model predictive control.

Remark 4. The proposed method is applicable to any dynamic system where data is obtained as time-series measurements. This includes most chemical processes and plant data. The method would not be directly applicable to data that are in other formats and structures such as images. An example is the sequence of moving digits from the Modified National Institute of Standards and Technology (MNIST) database, also known as the moving MNIST data set, where digits are moving. However, such data sets are mostly relevant in other disciplines of engineering and computer science rather than chemical engineering.

4. Applications

In this work, the effect of Gaussian white noise $w \sim \mathcal{N}(0, \sigma^2)$ in the sensor measurement y in Eq. (1) is analyzed. Specifically, the predictive ability of the model developed using SISC is benchmarked against the original sparse identification (SI) method as well as sparse identification with subsampling without co-teaching (SIS). First, the details of the demonstration procedure are outlined. Subsequently, the two examples, a predator-prey system and a chemical reactor example, are individually studied.

4.1. Data generation, sampling, and noise generation

The data generation procedure for both examples consists of first integrating the corresponding system of ODEs using 50 different initial conditions over a wide range of values to maximize coverage of the operating region of interest. The integration is carried out using the explicit Euler method with an integration time step of $h_c = 10^{-4}$, while the sampling time Δ varies between the examples. This yields the noise-free or clean data set.

For both examples, four levels of noise are considered in this study. Noise levels 1 (very low), 2 (low), 3 (medium), and 4 (high) refer to white Gaussian noise with standard deviations $\sigma_1 = 0.02$, $\sigma_2 = 0.1$, $\sigma_3 = 0.2$ and $\sigma_4 = 0.3$, respectively. For each level, white Gaussian noise is generated and amplified as necessary to account for any disparate scales of variables, and then added to the clean data set to generate the noisy data set.

Once the 50 different state trajectories are obtained from the open-loop simulations described above, the data set is split into 60% training, 20% validation, and 20% testing data, respectively. The train-validation-test split of 60-20-20 used is highly subjective, and many variations exist. Other common choices include 50-25-25, 70-15-15, and 80-10-10. There is no "right ratio" and trade-offs exist. If the training proportion is increased, the model is able to use more data and possibly generalize better. A larger validation set provides greater confidence in the model selection process, while a larger test set produces a better assessment of the ability of the model to generalize to new, unseen data. The 4 combinations suggested above attempt to balance these three objectives. However, as mentioned, these are not rigid rules and should be adjusted as required. For example, if data is scarce due to expensive experimental trials required to collect data, as is the case in several chemical processes, the smaller data set may warrant a larger training set to maximize the model performance. In this scenario, the validation data set can be reduced and a k-fold cross-validation technique may be used instead of information criteria. In fact, due to the lower number of data, the most expensive yet accurate validation technique, leave-one-out-cross-validation (LOOCV) can be used even though it is rarely used in practice for larger data sets due to the computational expense. For this scenario, an 80-10-10 scheme may be preferable. If the data set is extremely large, however, the preferred scheme may be 50-25-25 since there is sufficient data for training even with only 50% of the data being used. Moreover, as the training process is likely to be the most computationally expensive step, a smaller data set for training may also produce models faster. This strategy is common in training neural networks, where "minibatches" are used rather than the entire training data set.

For the base case with no subsampling, the entire training set is used for sparse identification to generate a single model following hyperparameters tuning. For both subsampling methods, the number of subsamples L is taken to be 5 in order to keep the computational burden reasonable. Furthermore, the total subsampling fraction, which is equal to p for the SIS case and $p+q$ for the SISC case, remains the same for both the SIS and SISC cases when compared. This is to ensure that both methods use the same total number of data points for training but only differ in the nature of the subsample used. The total subsampling fraction is assigned with the values of 0.2, 0.4, 0.6, and 0.8. The $p:q$ ratio is taken to be 4:1 throughout this paper as the goal is to demonstrate that only a small fraction of noise-free data is sufficient to improve the performance when the SI and SIS methods fail. For example, a total subsampling fraction of 0.2 indicates $p = .2$ for SIS and $p = .16, q = 0.04$ for SISC. Hence, for the SIS case, 20% of the training data set will be subsampled $L = 5$ times and a sparse-identified model will be trained for each subsample. For the SISC case, each of $L = 5$ times, 16% of the noisy training data set will be subsampled and then mixed with 4% of the noise-free training data set to yield a subsample that contains the same number of data points as the SIS method. Once the subsamples are extracted, a model is identified with sparse identification for each by tuning the hyperparameters and the best model based on the AIC is isolated.

Remark 5. The sampling time in the data generation step greatly affects the results of any model identification method because information is inevitably lost in the sampling step. A smaller sampling time will generally yield better results due to the more accurate derivative computations and a more comprehensive history of the trajectory of the state over the simulation duration. However, a very small sampling time such as 10^{-4} or 10^{-6} may be infeasible in practice. Hence, even if such a sampling time produces extremely accurate models, the variables, especially in a chemical process, can rarely be measured with such short sampling periods between each measurement. Hence, the values of the sampling period in this paper are of the order of 10^{-2}, which balances practicality with accuracy. This constitutes another limitation of past studies that primarily focus on high-fidelity data with impractically small sampling times to identify models with a very high accuracy that may not be immediately transferable to process engineering.

Remark 6. Apart from the sampling time and range of initial conditions chosen, the amount of data generated is also dependent on the simulation duration t_f. In case of a system with a steady-state solution, the simulation duration should be long enough for the system to reach the steady-state, ensuring that the same steady-state is reached for every initial condition. However, once the steady-state is reached, continuing the simulation for much longer is redundant as there are no new dynamics to be captured. Instead, focus can be shifted to regions of the trajectories with higher information density such as those with higher gradients and faster dynamics. For example, in the case of multiscale systems, Champion et al. (2019) suggests burst sampling—a sampling technique where the sampling time is as short as possible in the region of change of the fast subsystem before it converges to a slow manifold, and is much larger for the rest of the simulation duration since the dynamics of the slow subsystem can be captured with significantly fewer data points. Advanced sampling techniques such as burst sampling reduce the amount of data storage as well as computational burden and must be considered when generating data from simulations instead of following a simple, iterative procedure for data collection.

Remark 7. It is difficult to quantify the amount of data collection required to obtain an accurate model in a general manner that is applicable to all systems. However, when investigating the data structure required and attempting to train models for various patterns of generated data, it was observed from extensive simulations and trials that the dynamics covered over the duration of the simulation were key to identifying an accurate model as opposed to merely the amount of data itself or even the sampling time. Hence, a very short period of data collection with an extremely small, possibly physically infeasible, sampling time can

yield a very high number of data-points but will likely yield a poor model. Instead, a larger simulation period with a larger sampling time where a fewer total number of data points are collected will yield a superior model. Therefore, the emphasis should be not on the amount of data but rather the dynamics captured over the range of time of data collection. Once sufficient dynamic information is captured in the entire data set, the aforementioned rules of determining the train-validation-test split can be used.

Remark 8. The primary reason that some parameters such as the $p:q$ ratio were fixed as constants instead of tuning by treating as a hyperparameter was due to the large number of hyperparameters to be tuned in this system, and because it was sufficient to demonstrate the results for a low noise-free fraction as this implies that the results will generally improve when the noise-free fraction is increased.

4.2. Example 1: Predator-Prey model

The predator-prey system consists of two first-order differential equations describing the dynamics of two species: a predator and a prey. The equations take the form,

$$\dot{x}_1 = a_1 x_1 - a_2 x_1 x_2 \tag{13a}$$

$$\dot{x}_2 = a_3 x_1 x_2 - a_4 x_2 \tag{13b}$$

where x_1 and x_2 are the population (numbers) of the prey and the predator, respectively, with parameters $a_i \in \mathbb{R}, a_i > 0$ for $i = 1, 2, 3, 4$. In this paper, the specific system considered is

$$\dot{x}_1 = 0.5x_1 - 1.5x_1 x_2 \tag{14a}$$

$$\dot{x}_2 = x_1 x_2 - 0.5x_2 \tag{14b}$$

The objective is to reconstruct Eq. (14) from noisy data using the methods described in the previous section.

For data generation, using 40 out of 50 initial conditions between 0.01 and 2.0 for each variable, the system of Eq. (14) is integrated using a time step of $h_c = 10^{-4}$ from $t_0 = 0$ to $t_f = 20$ and sampled with $\Delta = 0.2$ for a total of 100 data points per trajectory. This yields a total of 4000 data points to be split into training and validation sets. For the testing set, the remaining 10 initial conditions are integrated identically as for the training/validation sets except t_f is increased to 30. The goal of extending the simulation period for the testing set beyond the training/validation sets is to gauge the extrapolating capacity of the model since one supposed advantage of reconstructing dynamical models as opposed to black-box models is the ability to extrapolate beyond the training data.

Fig. 2 shows the results of this system when the various model identification methods are employed. The various identified models are used to predict the states forward in time using the 10 initial conditions from the testing set and compared to the original testing data generated in the data generation step. The results were similar for both variables but more pronounced in x_1. To evaluate the results numerically, the mean-squared error (MSE) is also calculated between the original data and each predicted trajectory over the entire testing period. The MSE are summarized in Table 1. At the lowest noise level ($\sigma = 0.02$), the base case achieves a low test set MSE, and accurately models the system as seen in Fig. 2. However, further improvement is observed when subsampling with or without co-teaching, with the order of magnitude of the MSE being one order lower than the base case. Although it appears that the SISC method performed slightly inferior to SIS, both MSE are extremely small, and the difference between the two MSEs, 0.00018 and 0.00021, is negligible in this context, especially as also observed in Figs. 2 and 3. As the noise level is increased to low ($\sigma = 0.1$) and even medium ($\sigma = 0.2$) levels, the results do not change significantly, except all the errors are now higher. At the highest noise level ($\sigma = 0.3$), the base case deteriorates significantly, overpredicting during most of the simulation period. Subsampling without co-teaching is not sufficient, and yields even poorer results than the base case in this example. In contrast, mixing in only 20% of noise-free data to each subsample was sufficient to improve the model with a much lower MSE than the runs with no subsampling or co-teaching. Although specific time instances might have poor results in the case of SISC as well, this is not across most of the simulation period, and the results are generally much more accurate, especially as also proven by the lower MSE. It is noted that the test set MSE values are generally small, and hence, even the percent decreases in the test set MSE between the SIS and SISC cases are more significant than it appears, which confirms the large improvement in accuracy.

Table 1
Test set MSE for the predator-prey system.

σ	No subsampling	SIS	SISC
0.02	0.00596	0.00018	0.00021
0.1	0.15925	0.05550	0.03851
0.2	0.34151	0.08472	0.07614
0.3	0.95057	1.70773	0.35808

When the total subsampling fraction is increased from 0.2 to 0.4, 0.6, and 0.8, it is observed that the test set MSE usually decreases in this example. However, the difference between the MSE from the SIS and SISC methods also begins to decrease. This indicates that subsampling alone without co-teaching can be a large source of improvement in some (but not all) examples. However, there are two concerns to address if such a strategy is adopted. Firstly, the SISC method, when using only 40% of the data ($p + q = 0.4$), surpasses the accuracy of the SIS method under all values of p that are used, even $p = .8$ where double the amount of data is being used to identify a model. Secondly, as the proportion used for subsampling increases, the computational expense increases. Hence, the SISC method outperforms the SIS method in the sense that it uses smaller subsamples to identify a model with a lower (or even equal) MSE.

4.2.1. Justification of coarse search for λ

The most important hyperparameter in the sparse identification algorithm is the sparsification knob λ due to its impact on the model as well as the continuous nature of the parameter. It may be considered analogous to the learning rate in neural networks in these aspects. Therefore, it is typically tuned using either a fine search or a coarse-to-fine search. In this work, however, only 3 values were tested. To ensure the coarse search for the optimal λ did not invalidate the results, an investigation into the effect of λ on the results was carried out.

To study the effect of λ only, since the best optimizer, derivative approximator, and function library have already been determined, these parameters can be fixed at their optimal selections and only λ can be varied. This yields the best possible model that can be achieved by finely tuning λ while all other parameters have already been optimized. The results for $\sigma = 0.1$ are shown in Fig. 4. A number of observations can be made from Fig. 4.

1. Values of λ above 1.0 zero too many terms, possibly all terms, leading to very high errors, with the error remaining constant until $\lambda = 4.0$. Therefore, it is sufficient to only consider values of λ between 0 and 1.
2. For values of λ below 1.0, smaller values generally yield lower values for both the AIC and the MSE. However, large differences, particular in orders of magnitude, occur at intervals of approximately 0.1. Hence, a few small values of λ space 0.1 units apart

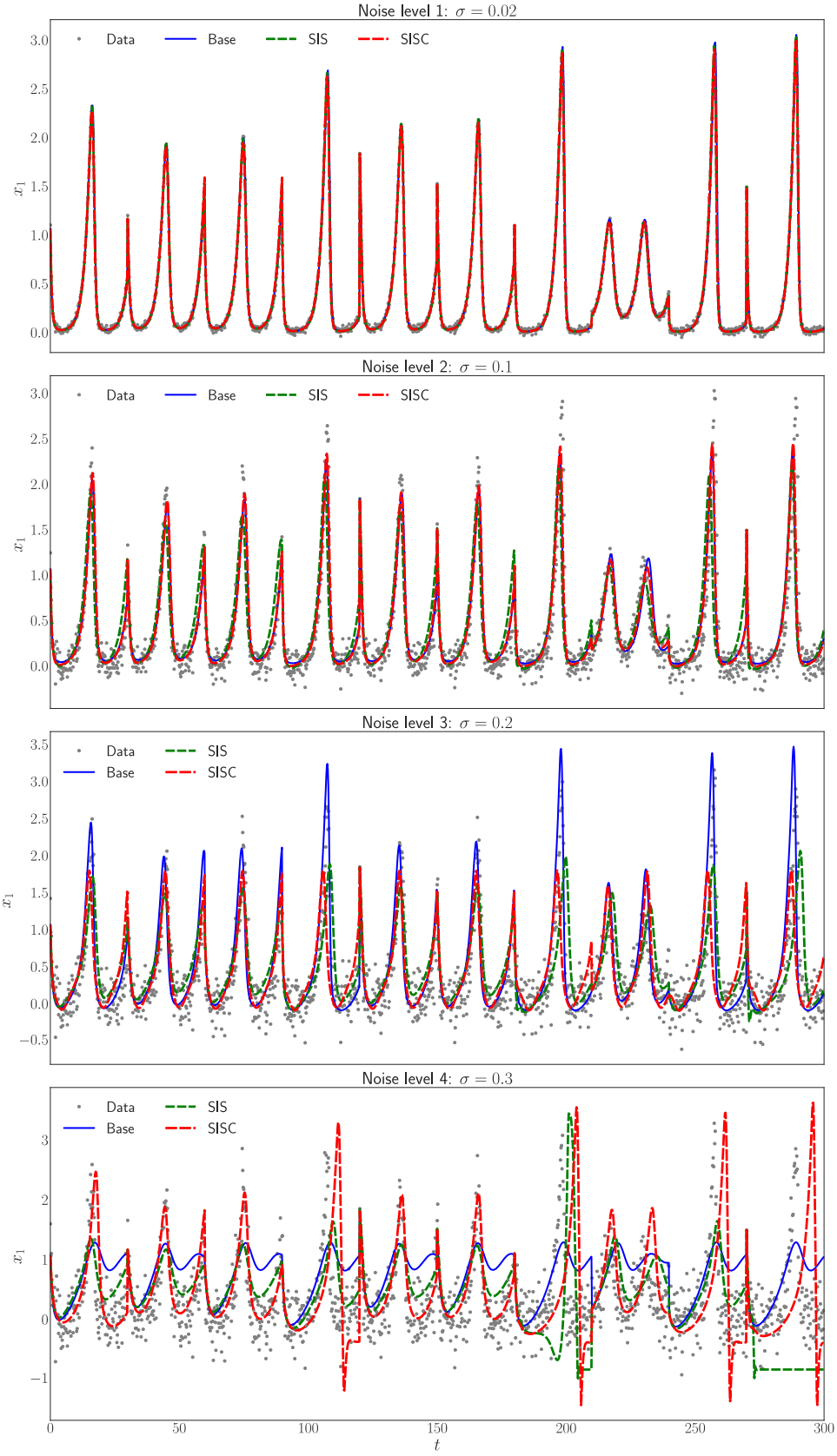

Fig. 2. Comparison of original noisy data (grey dots) with results from sparse identification without any subsampling (blue line), subsampling without co-teaching with $p = .2$ (green line), and subsampling with co-teaching and $p = .16, q = 0.04$ (red line) for the quantity x_1 in the predator-prey system. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

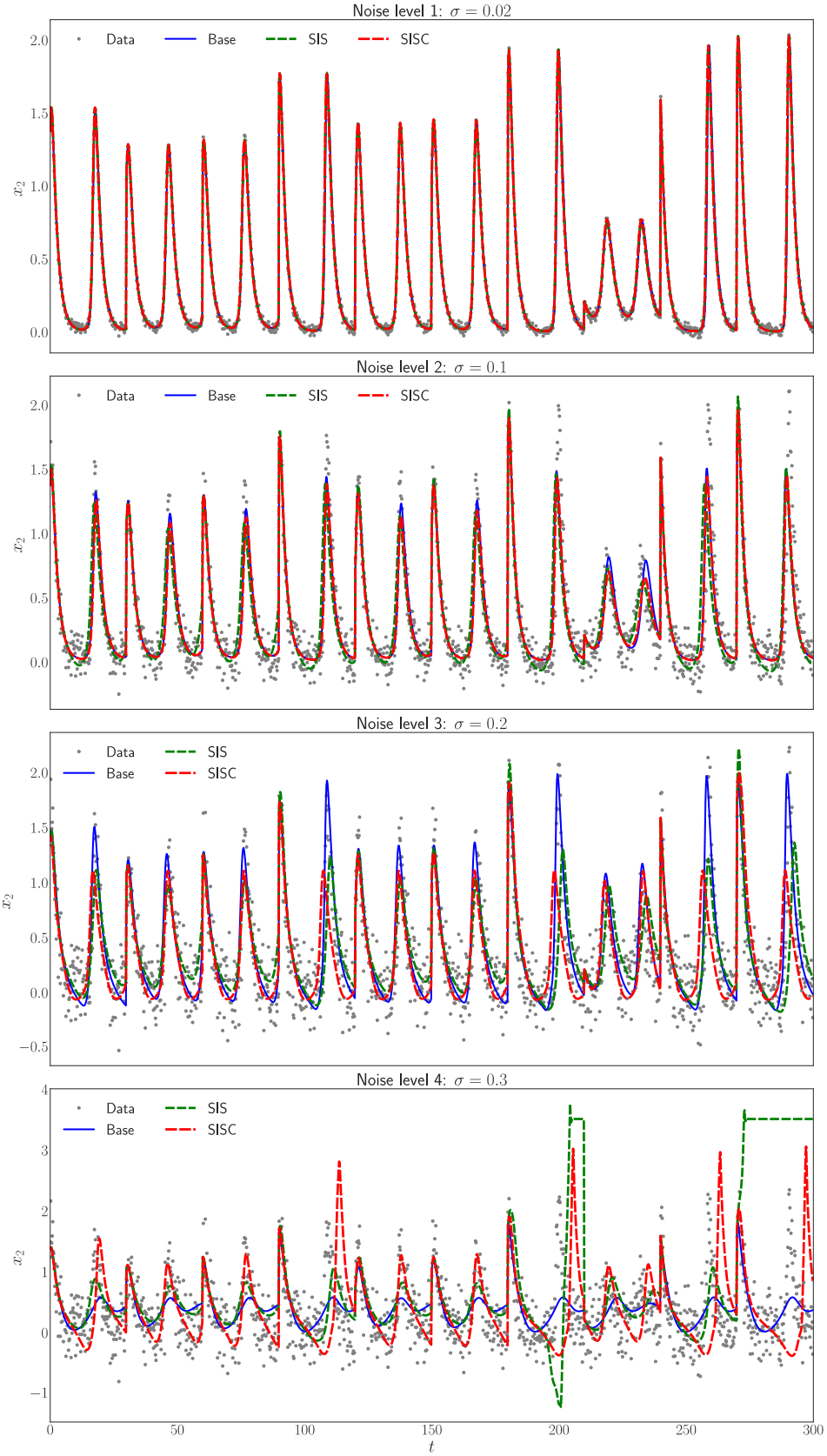

Fig. 3. Comparison of original noisy data (grey dots) with results from sparse identification without any subsampling (blue line), subsampling without co-teaching with $p = .2$ (green line), and subsampling with co-teaching and $p = .16, q = 0.04$ (red line) for the quantity x_2 in the predator-prey system. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

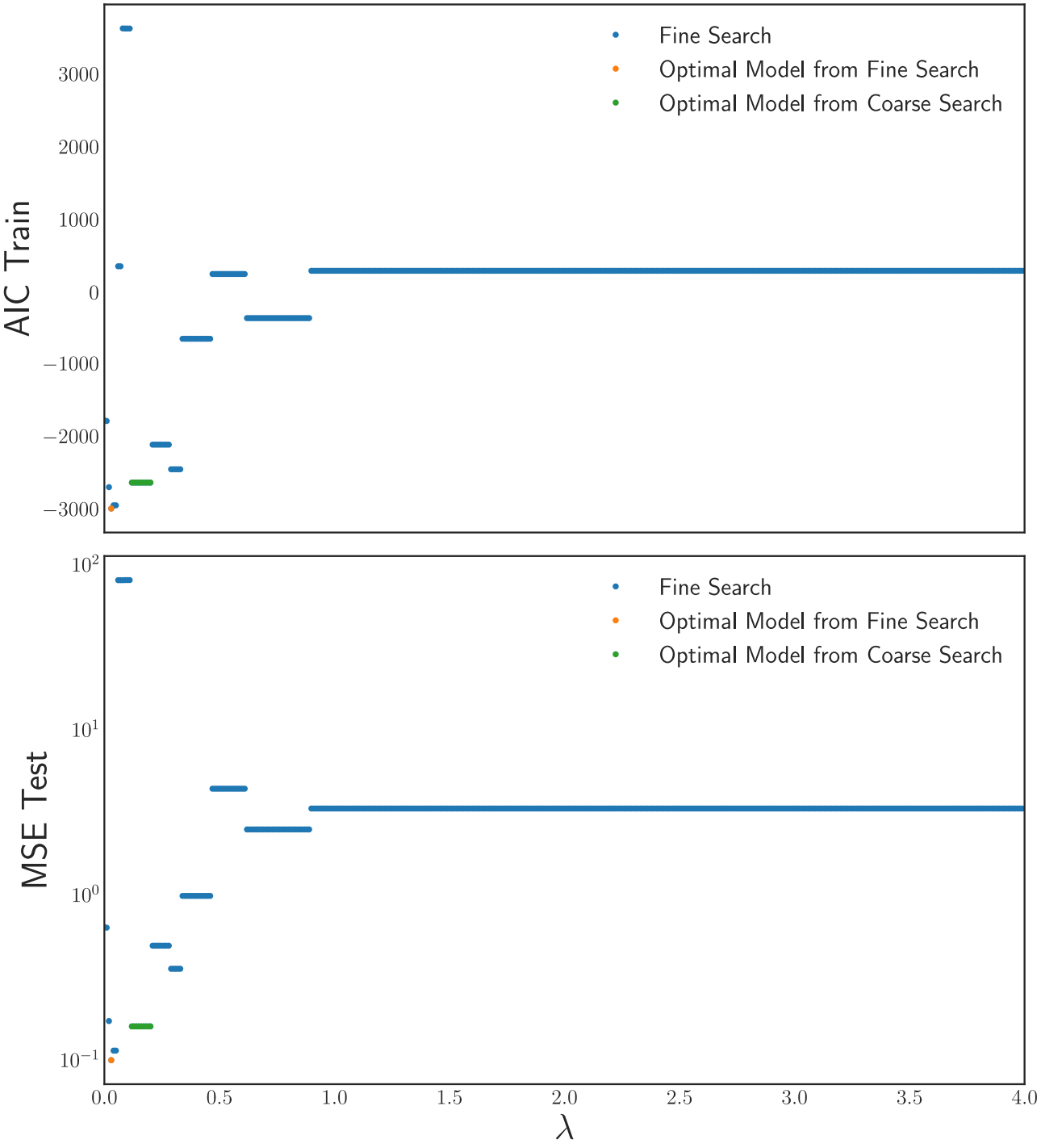

Fig. 4. Training set AIC and test set MSE for various $\lambda \in [0, 4]$ for the base case with no subsampling and $\sigma = 0.1$.

can yield a model very close to the best model possible and is sufficient for demonstration and comparison purposes.

3. At the noise level considered in this investigation of $\sigma = 0.1$, our optimal model was found to use the SR3 optimizer with the TVRD for derivative approximation and only a polynomial library. The best λ from our coarse search was 0.2. The final training AIC and test set MSE were -2617.451 and 0.15925, respectively.
4. In contrast, the best λ from the fine search is 0.03, which gives training AIC and test set MSE of -2973.28 and 0.09942, respectively. The decrease in the metrics from the optimal model from the coarse search is not very large. Moreover, the best λ from the fine search was only 0.03, requiring a grid spacing of 0.01 or smaller when searching for λ. The optimal model from the coarse search, however, has a relatively wide range of values over which the same model is obtained. Hence, a larger step can be used when searching for λ.
5. Due to the small differences in the metrics between the two optimal models in Fig. 4, it is possible that the optimal models would be reversed if the penalty on the number of terms were to be increased because the model from the coarse search only has 6 terms, while the model from the fine search consists of 15 terms. Specifically, in Eq. (12), the first term accounts for the model-fit or accuracy by using the MSE while

Table 2
Test set MSE for the predator-prey system using coarse and fine searches for λ in the *No subsampling* case.

σ	No subsampling (coarse)	No subsampling (fine)	SIS	SISC
0.02	0.00596	0.00596	0.00018	0.00021
0.1	0.15925	0.09942	0.05550	0.03851
0.2	0.34151	0.61213	0.08472	0.07614
0.3	0.95057	0.92768	1.70773	0.35808

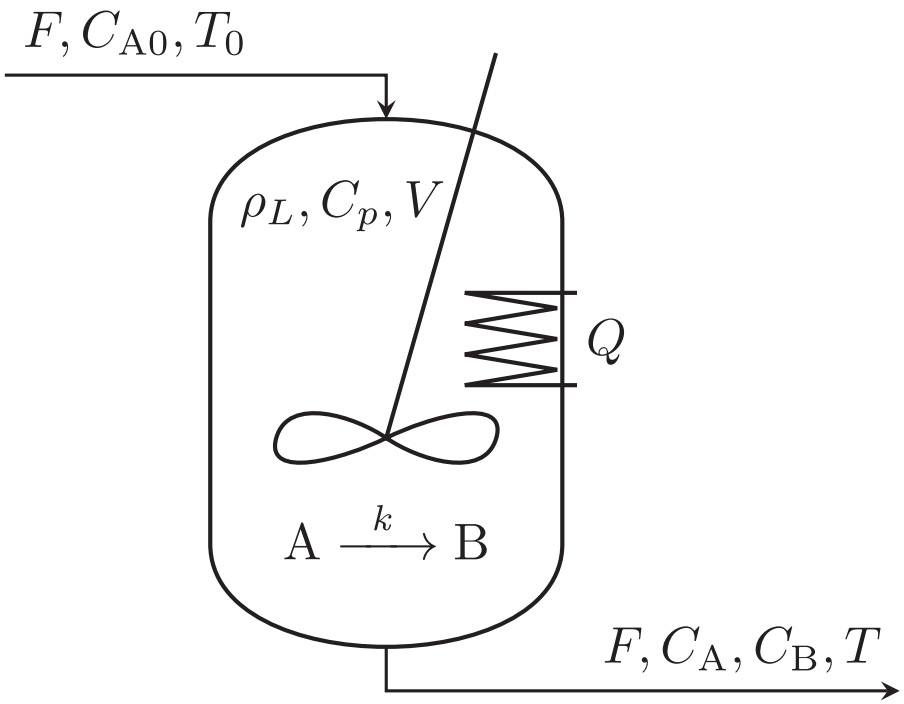

Fig. 5. A continuous-stirred tank reactor with a heating coil.

the second term penalizes the number of terms in the sparse-identified model. The pre-multiplier of "2" taken in Eq. (12) can be increased to penalize the number of terms more heavily and compromise on the accuracy. In fact, if the pre-multiplier on the number of terms is larger than 41.54, the model identified from the coarse search remains as the optimal model. Hence, a more elaborate scheme to select a search method for λ can take into consideration the balance between sparsity and accuracy. If a more sparse and less complex model is desired at the expense of some loss of accuracy, a coarse search may be preferred. If accuracy is crucial, a coarse-to-fine search may be the fastest method since a coarse search will determine the region for the fine search and reduce the number of computations required. In the case study of Fig. 4, a coarse search can reveal that values of λ below 0.5 require a finer search.

6. Finally, even the optimal model from the fine search for the base case is inferior in terms of the test set MSE to both the subsampling cases. Hence, even if the subsampling cases used a finer search for λ to yield a superior model, it would only increase the differences in performance. The key findings of this work are the clear, qualitative improvements between the three different methods rather than the exact quantitative improvement from one model to another.
7. As seen in Table 2, at the lowest noise level, the models produced by the two searches were identical. This is likely due to the optimal model being almost the exact model required to be found in both cases. Hence, a finer search is not required.
8. From Table 2, at the higher noise levels, which are the highlight of this work, it is observed that there is either an insignificant improvement in performance or no improvement at all. Therefore, in this work, a finer search for λ was omitted due to both the extremely heavy computational expense of optimizing such a continuous hyperparameter as well as the insignificant improvements that can be achieved from such an effort.

4.3. Example 2: CSTR

A chemical process example with noisy sensor data is considered. In particular, a single irreversible second-order exothermic reaction that converts a reactant A to a product B (A → B) takes place in a perfectly mixed non-isothermal continuous stirred tank reactor (CSTR) as shown in Fig. 5 described by the following set of ODEs:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{-\frac{E}{RT}} C_A^2 \tag{15a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{-\frac{E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{15b}$$

The states are the reactant concentration C_A and temperature T inside the reactor. The inlet contains pure reactant A with concentration C_{A0} and temperature T_0 at a flow rate F. A heating jacket surrounding the CSTR provides/removes energy at a rate Q to adjust the temperature. The fluid in the reactor is assumed to have a constant density of ρ_L with heat capacity C_p. The enthalpy of reaction, Arrhenius constant, activation energy of reaction, and the

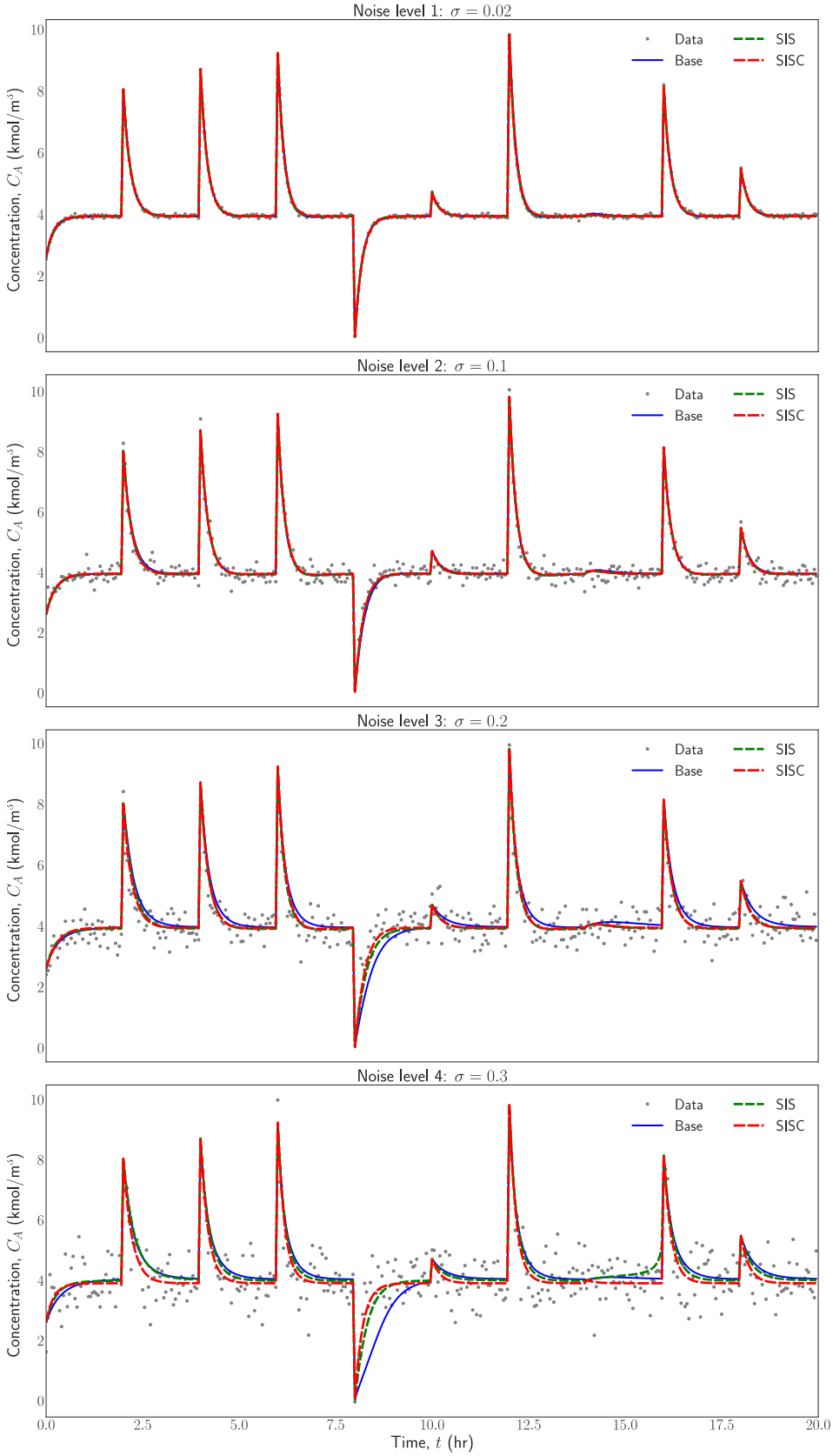

Fig. 6. Comparison of original noisy data (grey dots) with results from sparse identification without any subsampling (blue line), subsampling without co-teaching with $p = .2$ (green line), and subsampling with co-teaching and $p = .16, q = 0.04$ (red line) for the concentration C_A of the CSTR system. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

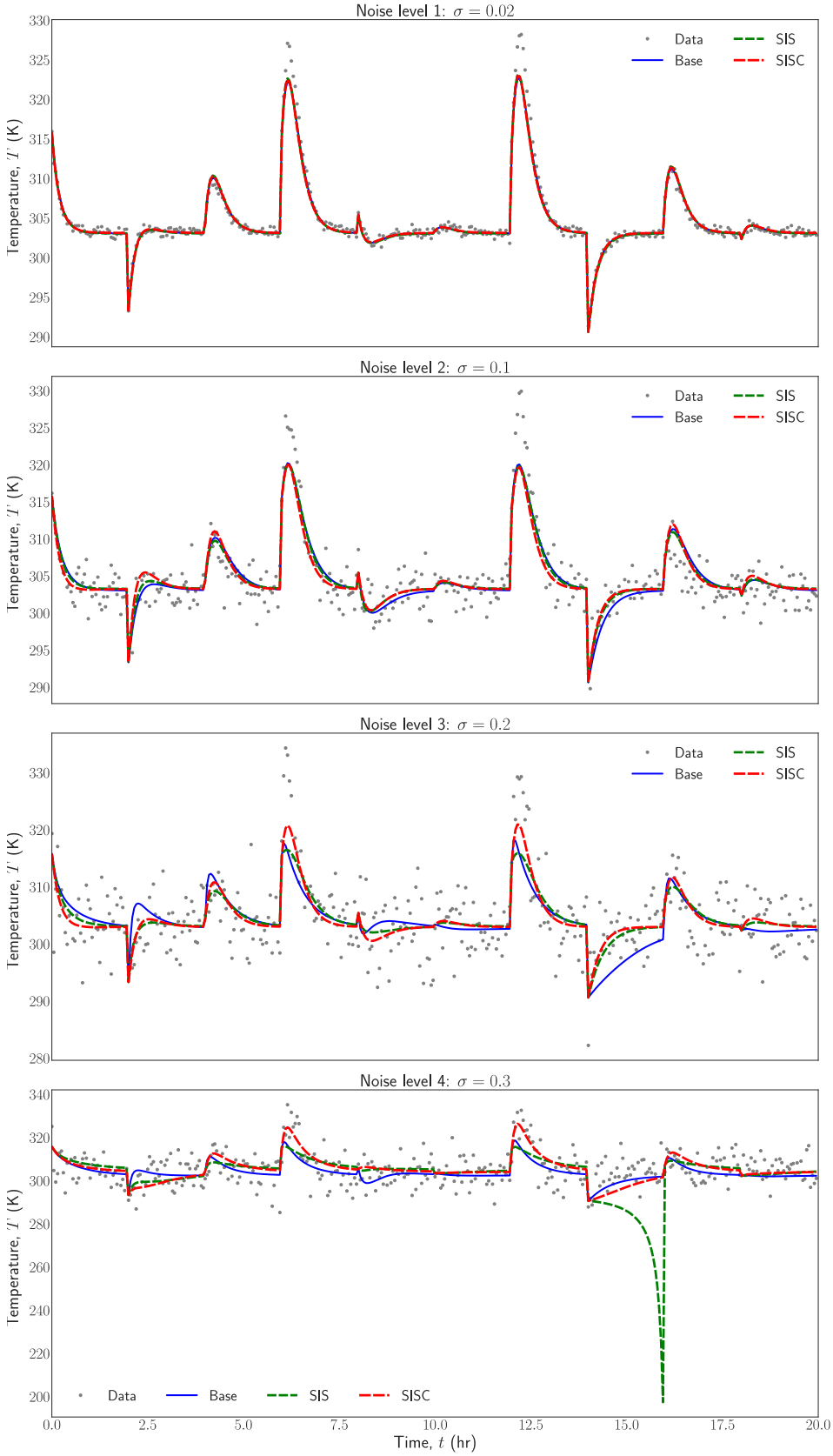

Fig. 7. Comparison of original noisy data (grey dots) with results from sparse identification without any subsampling (blue line), subsampling without co-teaching with $p = .2$ (green line), and subsampling with co-teaching and $p = .16, q = 0.04$ (red line) for the temperature T of the CSTR system. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Parameter values for chemical process example.

$F = 5.0\text{m}^3/\text{h}$	$V = 1.0\text{m}^3$
$k_0 = 8.46 \times 10^6 \text{m}^3\text{kmol}^{-1}\text{h}^{-1}$	$E = 5.0 \times 10^4 \text{kJ/kmol}$
$R = 8.314 \text{kJ kmol}^{-1}\text{K}^{-1}$	$\rho_L = 1000.0 \text{kg/m}^3$
$\Delta H_r = -1.15 \times 10^4 \text{kJ/kmol}$	$T_0 = 300.0\text{K}$
$Q = 0\text{kW}$	$C_{A0} = 4\text{kmol/m}^3$
$C_p = 0.231 \text{kJ kg}^{-1}\text{K}^{-1}$	

Table 4
Test set MSE for the CSTR system for four noise levels.

σ	No subsampling	SIS	SISC
0.02	0.01113	0.01059	0.01102
0.1	0.10510	0.09922	0.10370
0.2	0.49837	0.40037	0.36283
0.3	0.98210	1.89607	0.77613

ideal gas constant in appropriate units are denoted by $\Delta H, k_0, E$, and R, respectively. Process parameter values are given in Table 3. The total subsampling fractions considered in this example were 0.3, 0.4, 0.6, and 0.8 since 0.2 caused numerical issues.

The data generation for this process is carried out in a manner to ensure all 50 trajectories converge to the same steady-state. In particular, the initial concentration $C_A(0)$ is assigned values between 0.01mol/m^3 and 10mol/m^3, while the initial temperature $T(0)$ takes values between 290K to 320K. With each initial condition, the system of Eq. (15) is numerically integrated with explicit Euler with a step size of 10^{-4} hr from 0.0hr to 2.0hr and then sampled every 0.05hr to generate the clean data set for this process. In this example, all data generation was carried out until $t_f = 2.0\text{hr}$. After extensively tuning the hyperparameters, it was found that the sparse relaxed regularized regression (SR3) optimizer (Zheng et al., 2019) was consistently the superior optimizer for this system. The function library that yielded the optimal results was the library with only polynomial terms. In general, when selecting the function library, it is advisable to start with a small, less complex library, such as a polynomial library, and including trigonometric or other terms only when necessary.

The results for this system are plotted in Figs. 6 and 7 while the MSE are given in Table 4. It is observed that, at the lowest two noise levels of $\sigma = 0.02$ or 0.1, both methods satisfactorily capture the dynamics of the CSTR system, with small improvements from the base case to the SIS case and from the SIS case to the SISC case. However, as these differences are only noticeable when analyzing either Fig. 6 or Table 4 very closely, they are not significant. At the medium noise level of $\sigma = 0.2$, the differences become noticeable in some of the test trajectories. It is observed in Fig. 6 that the SIS prediction marginally outperforms the base case, but the SISC model significantly performs better than the SIS model, especially in the intervals $t \in (6, 10) \cup (12, 14)$. The MSE also decreases by roughly 20% from the base case to the SIS case and a further 10% between the SIS case and the SISC case. The largest differences, however, are seen at the highest level of noise *i.e.*, $\sigma = 0.3$. In this case, the SIS model actually fails to capture the dynamics completely and performs worse than the base case with almost twice the value of the MSE. In contrast, the SISC model outperforms the base case even using only 20% of the data in each subsample. There is a significant 21% drop in the MSE as seen from Table 4. However, the difference can also be observed in Fig. 7, particularly in x_1, which is the plotted variable, most strongly between $t = 8$ and $t = 10$.

In this example, as the total subsampling fraction is increased, there does not appear to be a consistent trend in the results. In fact, the results remain very similar for all the values of p and q tested. Therefore, it may be desirable to use lower values of the subsampling fraction and only increase it as required.

4.4. Advantages of explicit methods

The data generation in this paper is done via integrating the ordinary differential equation systems of the case studies considered in time using the Runge-Kutta numerical integration method with an integration time-step of $h_c = 10^{-4}$. As this is an explicit numerical integration method, the calculation time is on the order of milliseconds. To be precise, for the 50 total trajectories that the system is integrated along from the 50 initial conditions, the total time is around 0.1 seconds and 0.13 seconds for the predator-prey and CSTR systems, respectively.

The integration of the identified models is also on the same order: 0.1/50 or 0.13/50 seconds for a trajectory. The computational time for the 10 test trajectories was calculated and multiplied by 5 to make a fairer comparison to the times required for the first-principles system. The results had a wider range from 0.170 seconds to 0.482 seconds, with an average of 0.266 seconds. Two points must be emphasized here. Firstly, these times are slightly higher than those of the first-principles model data generation. This can be due to the higher complexity of the models. However, some of the models had the same number of terms as the first-principles model, with coefficients also very close to the original system. This was usually seen in the lower noise levels where a near-exact reconstruction was possible. These models should have required the same time to integrate as the first-principles models. Therefore, the reason for the higher times can be attributed to (a) the integration for the sparse-identified models being carried out by the PySINDy package's internal ODE integration function rather than Python's ODE solvers, and (b) some of the models in some of the subsamples might not be numerically stable, contributing to a large increase in the maximum and average values reported above. The most important fact to note, however, is that these times, especially for one trajectory, are still all below one second, and this rapid prediction is a key advantage of using ODE models with explicit nonlinearities in system identification as opposed to ODEs with neural network function approximators for the nonlinear basis functions, which is as a possibility if the identified model with the explicit functions is not accurate enough (not the case in our studies) at the expense of increasing computational burden.

Lastly, although the above range and average of integration times for the sparse-identified models are calculated from only the base (no subsampling) case of the predator-prey system, the results generalize to both of the other cases and also the CSTR system. This is because, once the ODE is identified, the data or procedure used to identify the ODE is irrelevant. Hence, even though the model identification step is more complex and time-consuming for the subsampling scenarios, once identified, the model is integrated using the same Runge-Kutta methods with the same step sizes. Moreover, there was no clear correlation between the number of terms in the identified ODE model and the integration time required, as long as the system was not unstable and did not diverge. Hence, the times reported above can be extended to all the sparse-identified models considered in this paper.

5. Conclusion

In this work, a novel algorithm was devised to build dynamical models that capture nonlinear process dynamics given only highly noisy sensor data. The noise was assumed to follow a white Gaussian distribution with different variances. A predator-prey model and a chemical process were used to demonstrate the performance and applicability of the new algorithm. It was shown that the basic sparse identification algorithm was inadequate in identifying the

model in the presence of high noise in the data, particularly above a variance of 0.01 for normalized data. However, when the subsampling technique was introduced, without co-teaching, by randomly subsampling to leave out the more noisy data in some iterations, it could identify the dynamics satisfactorily up to a noise variance of 0.04. Finally, the proposed algorithm combining subsampling with co-teaching, where the original data is subsampled but also mixed with some noise-free data from first-principles model simulations was used. Using the third algorithm, the performance improved slightly in the presence of noise with variance up to 0.04. However, at the highest noise level studied, which was characterized by a variance of 0.09, both the base case and the subsampling without co-teaching failed and could not identify the models using the extremely noisy data. The subsampling with co-teaching could accurately identify the models in this case, even when only 20% of the subsamples consisted of noise-free data generated from first-principles model simulations. The performance was evaluated based on plots of the outputs as well as the mean squared error (MSE) on the testing data sets. The results were qualitatively similar in both systems investigated, with more accurate models predicting the testing data set more accurately and yielding lower MSE values.

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.

CRediT authorship contribution statement

Fahim Abdullah: Conceptualization, Methodology, Software, Writing – review & editing. **Zhe Wu:** Conceptualization, Methodology, Writing – review & editing. **Panagiotis D. Christofides:** Writing – review & editing.

Acknowledgments

Financial support from the National Science Foundation and the Department of Energy is gratefully acknowledged.

References

Abdullah, F., Wu, Z., Christofides, P.D., 2021. Data-based reduced-order modeling of nonlinear two-time-scale processes. Chem. Eng. Res. Des. 166, 1–9.
Abdullah, F., Wu, Z., Christofides, P.D., 2021. Sparse-identification-based model predictive control of nonlinear two-time-scale processes. Computers & Chemical Engineering 153, 107411.
Aggelogiannaki, E., Sarimveis, H., 2008. Nonlinear model predictive control for distributed parameter systems using data driven artificial neural network models. Computers & Chemical Engineering 32, 1225–1237.
Al Seyab, R., Cao, Y., 2008. Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation. J Process Control 18, 568–581.
Ali, J.M., Hussain, M.A., Tade, M.O., Zhang, J., 2015. Artificial intelligence techniques applied as estimator in chemical process systems–a literature survey. Expert Syst Appl 42, 5915–5931.
AlMomani, A.A.R., Sun, J., Bollt, E., 2020. How entropic regression beats the outliers problem in nonlinear system identification. Chaos: An Interdisciplinary Journal of Nonlinear Science 30, 013107.
Aumi, S., Corbett, B., Clarke-Pringle, T., Mhaskar, P., 2013. Data-driven model predictive quality control of batch processes. AIChE J. 59, 2852–2861.
Bai, Z., Wimalajeewa, T., Berger, Z., Wang, G., Glauser, M., Varshney, P.K., 2015. Low-dimensional approach for reconstruction of airfoil data via compressive sensing. AIAA Journal 53, 920–933.
Boninsegna, L., Nüske, F., Clementi, C., 2018. Sparse learning of stochastic dynamical equations. J Chem Phys 148, 241723.
Bruckstein, A.M., Donoho, D.L., Elad, M., 2009. From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. 51, 34–81.
Brunton, S.L., Proctor, J.L., Kutz, J.N., 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences 113, 3932–3937.
Brunton, S.L., Tu, J.H., Bright, I., Kutz, J.N., 2014. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. SIAM J Appl Dyn Syst 13, 1716–1732.
Candès, E.J., 2008. The restricted isometry property and its implications for compressed sensing. C.R. Math. 346, 589–592.
Chaffart, D., Ricardez-Sandoval, L.A., 2018. Optimization and control of a thin film growth process: a hybrid first principles/artificial neural network based multiscale modelling approach. Computers & Chemical Engineering 119, 465–479.
Champion, K.P., Brunton, S.L., Kutz, J.N., 2019. Discovery of nonlinear multiscale systems: sampling strategies and embeddings. SIAM J Appl Dyn Syst 18, 312–333.
Chartrand, R., 2011. Numerical differentiation of noisy, nonsmooth data. ISRN Applied Mathematics 2011, 111.
Cohen, A., Davenport, M.A., Leviatan, D., 2013. On the stability and accuracy of least squares approximations. Foundations of Computational Mathematics 13, 819834.
Cortiella, A., Park, K.-C., Doostan, A., 2021. Sparse identification of nonlinear dynamical systems via reweighted $\ell 1$-regularized least squares. Comput Methods Appl Mech Eng 376, 113620.
Dam, M., Brøns, M., Juul Rasmussen, J., Naulin, V., Hesthaven, J.S., 2017. Sparse identification of a predator-prey system from simulation data of a convection model. Phys Plasmas 24, 022310.
Didonna, M., Stender, M., Papangelo, A., Fontanela, F., Ciavarella, M., Hoffmann, N., 2019. Reconstruction of governing equations from vibration measurements for geometrically nonlinear systems. Lubricants 7, 64.
Diversi, R., Guidorzi, R., Soverini, U., 2010. Identification of arx and ararx models in the presence of input and output noises. European Journal of Control 16, 242–255.
Doostan, A., Owhadi, H., 2011. A non-adapted sparse approximation of pdes with stochastic inputs. J Comput Phys 230, 3015–3034.
Efron, B., Stein, C., 1981. The jackknife estimate of variance. The Annals of Statistics 9, 586–596.
Fablet, R., Ouala, S., Herzet, C., 2018. Bilinear residual neural network for the identification and forecasting of geophysical dynamics. In: Proceedings of the 26th European Signal Processing Conference, pp. 1477–1481.
Garg, A., Mhaskar, P., 2018. Utilizing big data for batch process modeling and control. Computers & Chemical Engineering 119, 228–236.
González-García, R., Rico-Martínez, R., Kevrekidis, I., 1998. Identification of distributed parameter systems: a neural net based approach. Computers & Chemical Engineering 22, S965–s968.
Hadigol, M., Doostan, A., 2018. Least squares polynomial chaos expansion: a review of sampling strategies. Comput Methods Appl Mech Eng 332, 382–407.
Hampton, J., Doostan, A., 2015. Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression. Comput Methods Appl Mech Eng 290, 73–97.
Hampton, J., Doostan, A., 2015. Compressive sampling of polynomial chaos expansions: convergence analysis and sampling strategies. J Comput Phys 280, 363–386.
Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I.W., Sugiyama, M., 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, pp. 8536–8546.
Hesthaven, J.S., Gottlieb, S., Gottlieb, D., 2007. Spectral methods for time-Dependent problems. Cambridge University Press.
Huusom, J., Poulsen, N., Jørgensen, S., Jørgensen, J., 2012. Tuning siso offset-free model predictive control based on arx models. J Process Control 22, 1997–2007.
Juricek, B.C., Larimore, W.E., Seborg, D.E., 1998. Reduced-rank arx and subspace system identification for process control. IFAC Proceedings Volumes 31, 247–252.
Kaheman, K., Kutz, J.N., Brunton, S.L., 2020. Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 476, 20200279.
Kaiser, E., Kutz, J.N., Brunton, S.L., 2018. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474, 20180335.
Kosmatopoulos, E.B., Polycarpou, M.M., Christodoulou, M.A., Ioannou, P.A., 1995. High-order neural network structures for identification of dynamical systems. IEEE Trans. Neural Networks 6, 422–431.
Leylaz, G., Wang, S., Sun, J.-Q., 2021. Identification of nonlinear dynamical systems with time delay. International Journal of Dynamics and Control 1–12.
Lin, M., Cheng, C., Peng, Z., Dong, X., Qu, Y., Meng, G., 2021. Nonlinear dynamical system identification using the sparse regression and separable least squares methods. J Sound Vib 505, 116141.
Loiseau, J.-C., Brunton, S.L., 2018. Constrained sparse galerkin regression. J Fluid Mech 838, 4267.
Mackey, A., Schaeffer, H., Osher, S., 2014. On the compressive spectral method. Multiscale Modeling & Simulation 12, 1800–1827.
Mangan, N.M., Askham, T., Brunton, S.L., Kutz, J.N., Proctor, J.L., 2019. Model selection for hybrid dynamical systems via sparse regression. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 475, 20180534.
Mangan, N.M., Brunton, S.L., Proctor, J.L., Kutz, J.N., 2016. Inferring biological networks by sparse identification of nonlinear dynamics. IEEE Transactions on Molecular, Biological and Multi-Scale Communications 2, 52–63.
Mangan, N.M., Brunton, S.L., Proctor, J.L., Kutz, J.N., 2016. Inferring biological networks by sparse identification of nonlinear dynamics. IEEE Transactions on Molecular, Biological and Multi-Scale Communications 2, 52–63.

Mangan, N.M., Kutz, J.N., Brunton, S.L., Proctor, J.L., 2017. Model selection for dynamical systems via sparse regression and information criteria. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473, 20170009.
Menezes, J.M.P., Barreto, G.A., 2008. Long-term time series prediction with the NARX network: an empirical evaluation. Neurocomputing 71, 3335–3343.
Moore, C., 1986. Application of singular value decomposition to the design, analysis, and control of industrial processes. In: 1986 American Control Conference, pp. 643–650. Seattle, WA, USA
Nguyen, D., Ouala, S., Drumetz, L., Fablet, R., 2020. Assimilation-based learning of chaotic dynamical systems from noisy and partial data. In: ICASSP 2020 : International Conference on Acoustics, Speech, and Signal Processing, pp. 3862–3866. Barcelona, Spain
Ozolinš, V., Lai, R., Caflisch, R., Osher, S., 2013. Compressed modes for variational problems in mathematics and physics. Proceedings of the National Academy of Sciences 110, 18368–18373.
Patwardhan, S.C., Narasimhan, S., Jagadeesan, P., Gopaluni, B., L. Shah, S., 2012. Nonlinear bayesian state estimation: a review of recent developments. Control Eng Pract 20, 933–953.
Peng, J., Hampton, J., Doostan, A., 2016. On polynomial chaos expansion via gradient-enhanced ℓ1-minimization. J Comput Phys 310, 440–458.
Proctor, J.L., Brunton, S.L., Brunton, B.W., Kutz, J.N., 2014. Exploiting sparsity and equation-free architectures in complex systems. The European Physical Journal Special Topics 223, 2665–2684.
Quade, M., Abel, M., Nathan Kutz, J., Brunton, S.L., 2018. Sparse identification of nonlinear dynamics for rapid model recovery. Chaos: An Interdisciplinary Journal of Nonlinear Science 28, 063116.
Raissi, M., Perdikaris, P., Karniadakis, G.E., 2018. Multistep neural networks for data–driven discovery of nonlinear dynamical systems. arXiv:1801.01236.
Rauhut, H., Ward, R., 2012. Sparse legendre expansions via ℓ1-minimization. Journal of Approximation Theory 164, 517–533.
Rudin, L.I., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268.
Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N., 2017. Data-driven discovery of partial differential equations. Sci Adv 3, e1602614.
Rudy, S.H., Nathan Kutz, J., Brunton, S.L., 2019. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. J Comput Phys 396, 483–506.
Sarić, A.T., Sarić, A.A., Transtrum, M.K., Stanković, A.M., 2021. Symbolic regression for data-driven dynamic model refinement in power systems. IEEE Trans. Power Syst. 36, 2390–2402.
Schaeffer, H., 2017. Learning partial differential equations via data discovery and sparse optimization. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473, 20160446.
Schaeffer, H., Caflisch, R., Hauck, C.D., Osher, S., 2013. Sparse dynamics for partial differential equations. Proceedings of the National Academy of Sciences 110, 6634–6639.
Schaeffer, H., McCalla, S.G., 2017. Sparse model selection via integral terms. Phys. Rev. E 96, 023302.
Schaeffer, H., Tran, G., Ward, R., 2018. Extracting sparse high-dimensional dynamics from limited data. SIAM J Appl Math 78, 3279–3295.
Schaeffer, H., Tran, G., Ward, R., Zhang, L., 2020. Extracting structured dynamical systems using sparse optimization with very few samples. Multiscale Modeling & Simulation 18, 1435–1461.
Siegelmann, H., Horne, B., Giles, C., 1997. Computational capabilities of recurrent narx neural networks. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 27, 208–215.
de Silva, B.M., Higdon, D.M., Brunton, S.L., Kutz, J.N., 2020. Discovery of physics from data: universal laws and discrepancies. Frontiers in Artificial Intelligence 3, 25.
Tran, G., Ward, R., 2017. Exact recovery of chaotic systems from highly corrupted data. Multiscale Modeling & Simulation 15, 1108–1129.
Trischler, A.P., D'Eleuterio, G.M., 2016. Synthesis of recurrent neural networks for dynamical system simulation. Neural Networks 80, 67–78.
Van Overschee, P., De Moor, B., 1994. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica 30, 75–93.
Wang, W.-X., Yang, R., Lai, Y.-C., Kovanis, V., Grebogi, C., 2011. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. Phys. Rev. Lett. 106, 154101.
Wong, W., Chee, E., Li, J., Wang, X., 2018. Recurrent neural network-based model predictive control for continuous pharmaceutical manufacturing. Mathematics 6, 242.
Wu, P., Pan, H., Ren, J., Yang, C., 2015. A new subspace identification approach based on principal component analysis and noise estimation. Industrial & Engineering Chemistry Research 54, 5106–5114.
Wu, Z., Luo, J., Rincon, D., Christofides, P.D., 2021. Machine learning-based predictive control using noisy data: evaluating performance and robustness via a large-scale process simulator. Chem. Eng. Res. Des. 168, 275–287.
Wu, Z., Rincon, D., Christofides, P.D., 2020. Real-time adaptive machine-learning-based predictive control of nonlinear processes. Industrial & Engineering Chemistry Research 59 (6), 2275–2290.
Wu, Z., Rincon, D., Luo, J., Christofides, P.D., 2021. Machine learning modeling and predictive control of nonlinear processes using noisy data. AlChE J. 67, e17164.
Xie, W., Bonis, I., Theodoropoulos, C., 2015. Data-driven model reduction-based nonlinear MPC for large-scale distributed parameter systems. J Process Control 35, 50–58.
Yeo, K., Melnyk, I., 2019. Deep learning algorithm for data-driven simulation of noisy dynamical system. J Comput Phys 376, 1212–1231.
Zeng, J., Gao, C., Su, H., 2010. Data-driven predictive control for blast furnace ironmaking process. Computers & Chemical Engineering 34, 1854–1862.
Zhang, L., Schaeffer, H., 2019. On the convergence of the sindy algorithm. Multiscale Modeling & Simulation 17, 948–972.
Zhang, S., Lin, G., 2018. Robust data-driven discovery of governing physical laws with error bars. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474, 20180305.
Zhang, S., Lin, G., 2021. Subtsbr to tackle high noise and outliers for data-driven discovery of differential equations. J Comput Phys 428, 109962.
Zhang, Z., Wu, Z., Rincon, D., Christofides, P.D., 2019. Real-time optimization and control of nonlinear processes using machine learning. Mathematics 7 (10).
Zheng, P., Askham, T., Brunton, S.L., Kutz, J.N., Aravkin, A.Y., 2019. A unified framework for sparse relaxed regularized regression: SR3. IEEE Access 7, 1404–1423.