Computers and Chemical Engineering 174 (2023) 108247

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Data-based modeling and control of nonlinear process systems using sparse identification: An overview of recent results

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

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

ARTICLE INFO

Keywords:
Nonlinear processes
Sparse identification
Subsampling
Two-time-scale processes
Singular perturbations
Co-teaching
Dropout
Ensemble learning
Model predictive control

ABSTRACT

This paper discusses recent developments in the data-based modeling and control of nonlinear chemical process systems using sparse identification of nonlinear dynamics (SINDy). SINDy is a recent nonlinear system identification technique that uses only measurement data to identify model dynamical systems in the form of first-order nonlinear differential equations. In this work, the challenges of handling time-scale multiplicities and noisy sensor data when using SINDy are addressed. Specifically, a brief overview of novel methods devised to overcome these challenges are described, along with modeling guidelines for using the proposed techniques for process systems. When applied to two-time-scale systems, to overcome model stiffness, which leads to ill-conditioned controllers, a reduced-order modeling approach is proposed where SINDy is used to model the slow dynamics, and nonlinear principal component analysis is used to algebraically "slave" the fast states to the slow states. The resulting model can then be used in a Lyapunov-based model predictive controller with guaranteed closed-loop stability provided the separation of fast and slow dynamics is sufficiently large. To handle high levels of sensor noise, SINDy is combined with subsampling and co-teaching to improve modeling accuracy. The challenges of modeling and controlling large-scale systems using noisy industrial data are then addressed by using ensemble learning with SINDy. After summarizing the advances, a nonlinear chemical process is used to provide an end-to-end demonstration of process modeling using sparse identification with guidelines for chemical engineering practitioners. Finally, several future research directions for the incorporation of SINDy into process systems engineering are proposed.

1. Introduction

A central objective of scientific and engineering research is the derivation of the laws governing physical systems in the form of equations. With the explosion in data and computational power over the last two decades, the construction of these equations empirically from data has become more tractable than deriving physics-based first-principles models, especially for highly complex systems, and is gaining momentum in the literature. For many physical systems, the laws governing their dynamics take the form of ordinary differential equations (ODE) or partial differential equations (PDE) with time and/or space as independent variables. Common examples include the Boltzmann equation in thermodynamics and the Navier–Stokes equations in fluid dynamics (Zhang and Lin, 2018). The development of such time-varying predictive models is often a prerequisite for other objectives in a plant/system engineering context, such as predictive maintenance in operations engineering and advanced control system design in any closed-loop system with strict product requirements.

In chemical process systems, model predictive control (MPC) is an advanced control system that has been implemented and accepted widely in the industry (Holkar and Waghmare, 2010). As the name suggests, MPC uses a dynamical model such as an ODE to predict the process states (outputs) over a user-defined prediction horizon to be able to take the optimal control action based on anticipated possible future trajectories. A large body of literature on data-driven modeling in MPC can be found in Aggelogiannaki and Sarimveis (2008). Two of the most common, classical system identification algorithms include the singular value decomposition (Moore, 1986) and Numerical algorithms for Subspace State Space System Identification (N4SID) (Van Overschee and De Moor, 1994). However, machine learning (ML) methods, a type of data-driven modeling with numerous parameters and tunable hyper-parameters, have demonstrated highly accurate results when applied to complex systems with multiple interacting nonlinearities due to their high degree of freedom. Some examples of ML methods include support vector regressors, extreme gradient boosting, and,

* Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.compchemeng.2023.108247
Received 17 February 2023; Received in revised form 27 March 2023; Accepted 28 March 2023
Available online 31 March 2023
0098-1354/

particularly of interest in recent years, artificial neural networks. For example, in Wu et al. (2019a,b), recurrent neural networks were used to model nonlinear processes, and subsequent closed-loop stability results under recurrent neural network-based model predictive control were derived. Autoencoders, which are feedforward neural networks (FNN) that replicate the input at their output, have been the subject of several studies. While linear autoencoders correspond exactly to PCA, Kramer (1991) proposed the use of nonlinear autoencoders, which use nonlinear activation functions, as a form of nonlinear PCA. Autoencoders, particularly undercomplete autoencoders, are a powerful tool for dimensionality reduction due to the enforced reduction of the dimension in the intermediate or hidden layers of the network. Tsay and Baldea (2020) used undercomplete autoencoders to carry out nonlinear dimensionality reduction and build reduced-order models for integrated scheduling and control of chemical process operations. When the system identification and optimal scheduling computations were conducted in the latent variables with reduced dimensionality, it was found that the computational efficiency as well as the level of dynamic information provided were improved. In Schulze et al. (2022), Koopman theory was used to derive a Wiener-type formulation for handling multiple-input multiple-output (MIMO) input-affine dynamical systems. Specifically, reduced-order surrogate models were developed by combining autoencoders with linear dynamic blocks. The models were hypothesized to be particularly useful in control applications due to the high accuracy and dimensionality reduction capabilities of the proposed Wiener-type Koopman models. The integration of a Gaussian process model with MPC was proposed in Likar and Kocijan (2007) and applied to a gas–liquid separation process. The simple model structure of the Gaussian process model and, more importantly, the statistical information such as the prediction uncertainty provided by such a model were found to be desirable qualities for control-centric applications.

A potential drawback of traditional ML models has been their black-box nature, which limits their applicability and adoption in process systems engineering. Therefore, the field of hybrid modeling, sometimes referred to as "gray-box" modeling, which aims to combine *a priori* first-principles knowledge or domain expertise with black-box approaches such as ML models to improve both the accuracy and interpretability of the overall model, has recently attracted significant attention. Bikmukhametov and Jäschke (2020) outlines a number of approaches to incorporating physics into data-driven modeling including but not limited to:

(1) feature engineering, which refers to domain experts selecting and/or creating physically meaningful features from the data set obtained from sensors rather than using the raw measurements directly,
(2) residual modeling, which refers to building an ML model to model the residual between the known first-principles model and sensor measurements in order to build a model that captures the plant-model mismatch, and
(3) linear meta-model of models, where the solutions from multiple sub-models, which correspond to various parts of the overall system and are obtained using feature engineering, are combined into a linear meta-model by taking a weighted linear combination of all the models to represent the overall system accurately once the weights are tuned.

Alhajeri et al. (2021) investigated the use of FNNs to build state estimators in the absence of full-state feedback. Specifically, an FNN was used to model the nonlinear terms in the dynamics such as those corresponding to chemical reactions. In Alhajeri et al. (2022a,b), the links between layers of a recurrent neural network (RNN) were disconnected (i.e., corresponding weights zeroed) based on the process structure, leading to the elimination of erroneous model predictions and improved overall model accuracy. Sansana et al. (2021) provides a detailed overview of hybrid modeling and its evolution over the last three decades since it became a subject of interest in the scientific community. Sansana et al. (2021) reports that the *a priori* knowledge to be incorporated into hybrid modeling has typically been in the form of equations, and other forms of data/information such as plant floor experience and process flow sheets have not been investigated exhaustively. It was also found that data-driven modeling has typically been used to enhance previously known or derived mechanistic models, but the reverse, i.e., using mechanistic models to improve or constrain data-driven models, is largely unexplored. In the field of process monitoring and fault diagnostics, in particular, Sansana et al. (2021) highlighted the benefit of knowledge of causality that can be inferred via hybrid modeling.

The area of surrogate modeling in process systems engineering has, in parallel with the above directions, increased in research intensity. McBride and Sundmacher (2019) provides a comprehensive overview of advances in surrogate modeling in chemical engineering over the past three decades. A primary reason for the surge of interest in surrogate models is the increasing complexity of modern, highly accurate models used to simulate or model the nonlinear processes, scheduling problems, and complex thermodynamics that are ubiquitous in chemical engineering. Despite their increasing accuracy, such models encounter a number of challenges in downstream optimization and control applications. Due to the model complexity, the computational expense in terms of both processing power and time required to evaluate such models is exorbitantly large in many cases. While single function evaluations may be feasible in a practical setting, if the models are to be embedded into an optimization problem, such as set-point optimization or closed-loop control under an MPC, the computational demand becomes prohibitive due to the large number of function evaluations (typically hundreds or thousands) required to find such solutions. This is further complicated by black-box models since no simplification, such as omission of a term or otherwise, may be performed to find a compromise between model complexity and accuracy. If the type of model used is noisy or has discontinuities, this further complicates the problem, especially since, in this case, finite-differences cannot be used to estimate derivatives, which are crucial in optimization. To overcome these challenges, mathematically simpler models known as surrogate models have been proposed to approximate the input/output relationship of the complex models using much fewer model parameters and with much lower computational costs.

Although surrogate models can be used to approximate more complex models, another approach is to start with simpler model structures to model the desired system of interest and only add complexity as required. While methods such as N4SID and MOESP have been widely used over the past decades with varying degrees of success depending on the application and severity of the nonlinearities present, sparse identification for nonlinear dynamical systems (SINDy) is a recent method that aims to identify nonlinear ODEs directly from data, which are explicit and in closed-form, allowing them to be directly incorporated into MPC or any other optimization problem. Due to the availability of efficient differential equation solvers, the computational cost of integrating such models is generally low, especially if the models are well-conditioned. In the field of chemical engineering, SINDy has been used to identify reaction networks (Hoffmann et al., 2019) and to build reduced-order models for modeling and controlling a hydraulic fracturing process (Narasingam and Kwon, 2018). Despite the application of SINDy to several chemical process examples in the literature, a number of specific issues encountered in the modeling and control of chemical processes and plants remain to be addressed adequately, based on our review of the literature. Therefore, this paper provides a unified summary of recent advancements and novel extensions to SINDy to overcome numerous challenges that are encountered when applying SINDy to the domain of chemical engineering. Besides providing general guidance with respect to basis functions and numerical concerns, more specifically, the difficulties of modeling multiscale systems, noisy sensor data, and industrial processes are discussed.

In this manuscript, we apply SINDy to model and control three types of process systems:

(1) processes with time-scale multiplicities,
(2) simulated processes with high levels of sensor noise, and
(3) large-scale processes corrupted with high levels of industrial noise.

Each category of systems has associated challenges and are addressed using different improvements upon the original SINDy algorithm, the details of which will be discussed in the respective section. We note that, although SINDy was introduced with the intent of identifying the governing physical laws as closed-form differential equations consistent with known physics of the system of interest, the application of SINDy is not limited to such cases. As the product of SINDy is a closed-form ODE model with explicit nonlinearities, the resulting model can be directly incorporated into an MPC for efficient computations. Therefore, in this work, we use SINDy as a system identification algorithm with the ultimate goal of building dynamical models for controllers. The rest of this manuscript is outlined as follows: in Section 2, the general class of nonlinear process systems under consideration is described. Section 3 details the SINDy algorithm along with general guidelines and tuning considerations for building SINDy models, and its formulation in a model predictive controller. In Section 4, the challenges of two-time-scale systems are discussed, while Sections 5 and 6 address the challenges of noisy data for simulated processes and large-scale industrial processes, respectively. A detailed, end-to-end practical demonsration of applying SINDy to a highly nonlinear chemical process is given in Section 7. Finally, Section 8 provides a number of research directions for furthering the application of SINDy for process modeling and control.

2. Class of nonlinear process systems

We consider the class of nonlinear process systems described by the following first-order ODE:

$$\dot{x}(t) = f(x) + g(x)u + w, \qquad x(t_0) = x_0 \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^r$ is the manipulated input vector, and $w \in \mathbb{R}^n$ is the noise vector. The unknown vector and matrix functions $f \in \mathbb{R}^n$ and $g \in \mathbb{R}^{n\times r}$, respectively, constitute the process model representing the inherent physical laws constraining the system and are assumed to be locally Lipschitz vector and matrix functions of their arguments with $f(0) = 0$. The manipulated input is constricted to be in r nonempty convex sets defined as $\mathcal{U}_i \subseteq \mathbb{R}, i = 1,\dots,r$. The sensor noise w is assumed to be bounded within the set $W := w \in \mathbb{R}^n : \|w\|_2 \leq \theta, \theta > 0$. The class of systems of the form of Eq. (1) is further restricted to the family of stabilizable nonlinear systems, i.e., there exist a sufficiently smooth control Lyapunov function $V(x)$ and a control law $\Phi(x) = [\Phi_1(x)\cdots\Phi_r(x)]^\top$ that renders the nominal ($w \equiv 0$) closed-loop system of Eq. (1) asymptotically stable under $u = \Phi(x)$. The stability region Ω_ρ is defined as the largest level set of V where $\dot{V}$ is rendered negative. Without loss of generality, the initial time t_0 is taken to be 0 throughout the article.

3. Methodology: Sparse identification of nonlinear dynamics

3.1. Overview of the sparse identification method

Based on sparse regression and compressive sensing, sparse identification of nonlinear dynamics (SINDy) is a novel method in the field of system identification (Bai et al., 2015; Brunton et al., 2016) and has been applied to a diverse array of engineering problems (Bhadriraju et al., 2020). The aim of SINDy is to use only input/output data from a system to represent the dynamics in the form of the nominal system of Eq. (1),

$$\dot{\hat{x}}(t) = \hat{f}(\hat{x}) + \hat{g}(\hat{x})u \tag{2}$$

where $\hat{x} \in \mathbb{R}^n$ is the state vector of the sparse-identified model, and $\hat{f}$ and $\hat{g}$ are the model parameters that capture the physical laws governing the system.

Since most physical systems contain only a few terms in the right-hand side of Eq. (2), if a large number of nonlinear basis functions are considered as possible terms in $\hat{f}$ and $\hat{g}$, the space of all candidate functions considered is rendered sparse. Hence, SINDy aims to identify the small number of active functions in $\hat{f}$ and $\hat{g}$ using algorithms that leverage sparsity. We first obtain a discrete set of full-state measurements from open-loop simulations or experiments and concatenate them into a data matrix X and an input matrix U,

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

$$U = \begin{bmatrix} u_1(t_1) & u_2(t_1) & \cdots & u_r(t_1) \\ u_1(t_2) & u_2(t_2) & \cdots & u_r(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(t_m) & u_2(t_m) & \cdots & u_r(t_m) \end{bmatrix} \tag{3b}$$

where $x_i(t_\ell)$ and $u_j(t_\ell)$ represent the measurement of the i^{th} state and j^{th} input at the ℓ^{th} sampling time, respectively, where $i = 1,\dots,n$, $j = 1,\dots,r$, and $\ell = 1,\dots,m$. $\dot{X}$, the time-derivative of X, is a required matrix in the sparse identification algorithm and is either measured if possible (e.g., velocity) or otherwise estimated from X. Subsequently, a function library $\Theta(X,U)$ is constructed with s nonlinear functions of X and U. These s functions are the candidate nonlinear functions that may be zero or nonzero in the right-hand side of Eq. (2). The sparse identification algorithm exploits sparsity to calculate the coefficients associated with the terms in the library, Θ. Given the universality of mononomials, polynomials, and trigonometric functions in engineering systems (Brunton et al., 2016), they are often selected as the initial library in Θ. An example of an augmented library is

$$\Theta(X,U) = \begin{bmatrix} | & | & | & | & | & | \\ \mathbf{1} & X & \sin(X) & e^X & U & UX^2 \\ | & | & | & | & | & | \end{bmatrix} \tag{4}$$

The goal of sparse identification is to find each of the s coefficients associated with the s nonlinear functions considered in Θ for each row of Eq. (2). Each state x_i corresponds to a sparse vector of coefficients, $\xi_i \in \mathbb{R}^s$, that represent the nonzero terms in $\hat{f}_i$ and $\hat{g}_i$ in the respective ODE, $\dot{\hat{x}}_i = \hat{f}_i(\hat{x}_i) + \hat{g}_i(\hat{x}_i)u$. Consequently, there are n such coefficient vectors that must be calculated. In matrix notation, the unknown quantity is

$$\Xi = \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_n \end{bmatrix} \tag{5}$$

which is found by solving the following equation:

$$\dot{X} = \Theta(X,U)\Xi \tag{6}$$

Eq. (6) may be solved using standard least-squares after reformulating the problem as such by setting all coefficients in Ξ below a certain threshold λ to zero. Specifically, the least-squares problem takes the form,

$$\Xi = \arg\min_{\Xi'} \left\|\dot{X} - \Theta(X,U)\Xi'\right\|_2 + \lambda \left\|\Xi'\right\|_1 \tag{7}$$

where the first term maximizes the fidelity of the model to the data, while the second term is an L_1 regularization term that ensures sparsity of Ξ. In Eq. (7), Ξ' is a notational substitute for Ξ. To solve Eq. (7), the least-squares problem is written in the following form, which may be solved using a standard solver for a linear system of equations:

$$\Xi = \arg\min_{\Xi''} \left\|\dot{X} - \Theta(X,U)\Xi''\right\|_2 \tag{8}$$

where the matrix Ξ'' is Ξ' with all coefficients having an absolute value below λ set to zero. Eq. (8) is repeatedly solved until convergence of the non-zero coefficients. The iterations typically converge rapidly due to the sparse structure of Ξ. An alternate algorithm to solve Eq. (6) is known as Sparse Relaxed Regularized Regression (SR3), which is based on the well-known LASSO operator (Zheng et al., 2019). After finding Ξ using either method, the identified model can be formulated as the continuous-time differential equation,

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

where $\Theta(x^{\top}, u^{\top})$ is a column vector containing symbolic functions of x and u from the chosen function library, and $x^{\top}$ represents the transpose of x.

3.2. Data generation and SINDy modeling considerations

When applying SINDy to an engineering problem, a number of factors affect the results and must be carefully considered before and during the construction of a SINDy model.

3.2.1. Data generation

Data for system identification methods is typically obtained from either open-loop simulations or open-loop experiments. The sampling period used to record the data, the variation of system inputs and outputs considered for data generation, and the distribution of the data set are some of the properties that affect the amount of dynamic information contained in the data set and, as a result, the model quality.

Firstly, as information is lost when continuous data is sampled into discrete data, a higher sampling rate (lower sampling period) generally leads to better system identification for any method including SINDy, especially since SINDy requires estimates of the time-derivative of the states using finite differences or some variant thereof. However, it is important to consider practical limitations in terms of sampling. While an extremely small sampling period of 10^{-5} units may produce a data set with high information density, from which derivative estimations can likely be made very accurately, leading to the identification of better models, such a high sampling rate is typically not possible to achieve in a chemical process application or even in many other engineering disciplines. Instead, the sampling period should be chosen to be as small as reasonably possible, which would also be desirable in practice. Manufacturer specifications of the relevant type of sensors for process variables may be used as a lower bound on the sampling period for simulations-based studies.

Secondly, the dynamic information captured in a data set is dependent on the initial conditions chosen, the input signal variation, and the total simulation duration. The chosen combinations of initial conditions and input variables must cover as much of the operating region of interest as possible, and the simulation should be run until it reaches the desired steady-state of operation, in order to maximize the dynamic information captured in the data set. In contrast, if the data collection is carried out using a narrow range of initial conditions and/or inputs, or if a large part of the trajectories are zero values at the steady-state due to excessive run time, the data set may be large but contain little dynamic information to build an adequate model from. Furthermore, based on our studies, SINDy modeling works best with longer trajectories, even if from fewer initial conditions, rather than an exponential number of extremely small trajectories from many random initial conditions. The open-loop runs, whether experimental or simulations-based, should also reflect the various types of actions that are relevant in a control setting. For example, a number of trajectories should use a nonzero input to drive the system to various regions of the state space, which will assist the model in identifying the input dynamics. However, a few runs should also initiate the system away from the steady-state and let the states approach the steady-state under no control action, provided that the steady-state of interest is a stable one. If the data set is generated following the above best practices, it should yield an independent and identically distributed (i.i.d.) data set with maximum dynamic information and the least redundancy/repetition.

Lastly, when dealing with a specific type of a system, any unique characteristics of the system that may hinder or facilitate data generation and quality should be considered. For example, since the goal is to capture as much dynamic information as possible and not collect redundant data over a large period of the simulation with constant values for all variables (i.e., after the system reaches a steady state), when dealing with multiscale systems, techniques such as "burst sampling" have been proposed in Champion et al. (2019). Burst sampling refers to the use of a short sampling time in regions with higher gradients and faster dynamics, such as the fast transient of the fast subsystem(s) of a multiscale system, while reducing the sampling rate once the fast states converge to the slow manifold. Such advanced sampling strategies greatly reduce data storage requirements, and allow the user to retain only the most informative bits of data to be used for modeling and control. Such advanced data acquisition strategies should be used instead of mere iterative procedures. On the other hand, if the system operates at or near an unstable steady-state, integrating the system for extended periods of time may lead to the states diverging (if the system does not have another steady-state that is stable), which will cause errors during run time and hinder the data generation. Hence, for unstable operating points, it is desirable to use multiple shorter trajectories. Such facilitation and difficulties of data generation must be considered on a case-to-case basis for the system being studied.

3.2.2. Data preprocessing

In any machine learning (ML) application, it is essential to preprocess the data before training a model. The two preprocessing steps required to apply SINDy are the train/test split and the normalization of the data set.

With respect to the split, the data set must first be split into the training and test sets. Most of the training data set is used to regress the model coefficients, while a small fraction of the training set is reserved as the validation set, which is used to tune the hyper-parameters. Once the optimal set of hyper-parameters is found, the model is finalized on the entire training data with the selected hyper-parameters. The final model is then bench-marked against the unseen data, which is the test set, also referred to as open-loop tests in control applications. The train–validation–test split ratio is arbitrary to an extent, although general rules and best practices exist. The training set should generally be the largest because the model performance is mostly related to the training data set, which is used to find the model parameters, while the test set is only used to gauge the model accuracy post-training. In fact, as long as the data set is i.i.d., increasing the size of the training set will always lead to an improvement of the model accuracy. The train–validation–test proportions are also determined by the application. For example, for methods with a large number of hyper-parameters to tune, such as neural networks, where usually large volumes of data are usually available, it may be more valuable to have a larger validation set, e.g., a 50-25-25 train–validation–test split. On the other hand, for SINDy, since there is only one key hyper-parameter to tune, a larger training set may be warranted, such as a 60-20-20 or 70-15-15 split. When the data set poses additional challenges such as noise, it may warrant an 80-10-10 split, since training noisy data is often a difficult task and requires significant amounts of data, especially if any smoothing or other, additional preprocessing steps are required.

A number of methods exist to normalize, i.e., center and scale the data set. Three common methods for normalization are the z-score scaler, Min-Max scaler, and Max-Abs scaler, of which the first two methods both center and scale the data, while the Max-Abs scaler only scales it. Specifically, the z-score scaler first centers the data set to its mean value by subtracting the mean, and then scales the data set to have unit variance by dividing by the standard deviation. The Min-Max scaler divides each number by the range of the data after subtracting the minimum value of the data set and then adds the minimum value

back to the scaled number in order to transform all data points to values between a lower and upper limit, usually 0 and 1, respectively. The Max-Abs scaler only scales the data to be between ±1 by dividing the data set by its maximum absolute value, without any subtraction or centering. While the methods are described for a single variable, for the multivariate case, the above operations are independently carried out on each variable or column of the data set. For chemical processes, as the process inputs and outputs are often written in deviation form from their steady-state values, further centering may not be as crucial; all variables will attain a value of zero at the steady-state. However, due to the large differences in the orders of magnitudes between the variables, such as between concentrations and temperatures, scaling the data set is necessary in most cases. When using SINDy, where the sign of the coefficients associated with certain terms can contain information on the process dynamics (e.g., an increase in the input heating rate should lead to an increase in the temperature), methods that scale without centering such as the Max-Abs scaler can be a reasonable starting point when deciding on a normalization method, as was also observed in some of our results.

3.2.3. Hyper-parameter tuning

In the basic SINDy algorithm, the model structure and accuracy are simultaneously controlled and balanced by a single hyper-parameter, λ. Therefore, tuning it is essential and usually carried out via a fine search or coarse-to-fine search. The latter is computationally efficient and used in this work. A coarse-to-fine search can be justified by the fact that, for appropriately scaled data, for most systems, no nonzero terms will remain in the SINDy model for large values of λ, such as λ that is an order of magnitude greater than the scaled data set. In contrast, extremely small values of λ will yield dense models that are prone to instability as well as redundant in the basis functions. Hence, a coarse search can be used to bound the region where a finer search can be carried out to identify the optimal model that yields the lowest loss or error metric. Fig. 1 demonstrates how this process can be used to select the optimal model (corresponding to the orange point) through a very fine search or even models very close to the optimal in terms of accuracy by a much coarser search (corresponding to the green region). As expected, it can be observed that values of $\lambda > 1.0$ zero all terms in the SINDy model, leading to a constant error for all such λ. At the lower extreme of values of λ, the model is no longer sparse, and some terms that may even lead to an unstable model can start to have nonzero coefficients, in which case the MSE rapidly increases even beyond the case of all the terms being zero. This is especially the case since Fig. 1 is based on the work in Abdullah et al. (2022a), where the case of noisy data is considered.

The basis functions chosen for the candidate library are another central element of the SINDy method, which may be treated similarly as a hyper-parameter, in the sense that it is not entirely arbitrary and may require addition/removal of basis functions as necessary. Expanding it without computational considerations is not recommended as the overall optimization problem will then suffer from the curse of dimensionality, while also rendering the model more prone to instabilities due to dense model structures. Therefore, if there is any physical insight on the type of nonlinearities that are potentially relevant to the system of interest, this physical insight should be incorporated into the optimization search (e.g., biasing the order with which the nonlinearities are considered in the optimization search in an approach similar to the ALAMO modeling technique (Wilson and Sahinidis, 2017)). For chemical processes with nonlinear reaction terms, a common consideration may be to include exponential terms involving the temperature as the Arrhenius rate law is widely used in deriving mass and energy balances for reactors.

For estimating the time-derivative $\dot{X}$ in the right-hand side of Eq. (6), which is typically unavailable from sensor measurements, the ideal method to be used depends on the nature of the data set. For clean data, any finite difference-based approach such as forward, backward, or centered finite difference is usually adequate and will eventually yield similar results for the model coefficients at the end of the SINDy algorithm. However, if the data is noisy, finite differences are unstable even at low noise levels. Hence, methods robust to noise such as the total variation regularized derivative (TVRD) and the smoothed finite-difference (SFD) have been proposed (Brunton et al., 2016). TVRD is based on the total-variation regularization, which has been widely used in image processing applications. In TVRD, the derivative is computed as the minimizer of a functional using gradient descent. In contrast, in SFD, the data set is first presmoothed using a filter, which may be a low-pass filter or the Savitzky-Golay filter, and finite-differences are then computed from the resulting, smoothed data set. As no gradient descent is involved, computationally, it is generally faster than TVRD. However, when both methods were used in Abdullah et al. (2022b), each method yielded some of the final, optimal models for the various cases studied, making them both reasonable choices to test.

3.3. Incorporation of SINDy within MPC

Model predictive control is an advanced control methodology that utilizes a model of the process to predict the states/output over a prediction horizon to compute the optimal control actions by solving an online optimization problem. The formulation of a Lyapunov-based model predictive controller (LMPC) that uses a sparse-identified ODE, $F_{si}(\cdot)$, as the process model is presented below:

$$\mathcal{J} = \min_{u\in S(\Delta)} \int_{t_k}^{t_{k+N}} C(\tilde{x}(t), u(t))\, \mathrm{d}t \tag{9a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{si}(\tilde{x}(t), u(t)) \tag{9b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{9c}$$

$$u(t) \in \mathcal{U},\ \forall\, t \in [t_k, t_{k+N}) \tag{9d}$$

$$\dot{\hat{V}}(x(t_k), u) \le \dot{\hat{V}}\big(x(t_k), \Phi_{si}(x(t_k))\big),\ \text{if } x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{si}} \tag{9e}$$

$$\hat{V}(\tilde{x}(t)) \le \rho_{si},\ \forall\, t \in [t_k, t_{k+N}),\ \text{if } x(t_k) \in \Omega_{\rho_{si}} \tag{9f}$$

where $\tilde{x}$ is the predicted state trajectory, $S(\Delta)$ represents the set of piece-wise constant functions with a period of Δ, and N is the number of sampling periods within each prediction horizon. $\dot{\hat{V}}(x,u)$ is the time-derivative of the Lyapunov function and is equal to $\frac{\partial \hat{V}(x)}{\partial x} F_{si}(x,u)$. $u = u^*(t),\ t \in [t_k, t_{k+N})$ denotes the optimal input sequence over the prediction horizon, which is provided by the optimizer. The LMPC applies only the first value in $u^*(t_k)$ over the next sampling period $t \in [t_k, t_{k+1})$, and solves the optimization again at the next sampling time t_{k+1}.

In the MPC formulation, Eq. (9a) is the objective function to be minimized and is chosen to be equal to the integral of $C(\tilde{x}(t), u(t))$ over the prediction horizon. A typical cost function that achieves a value of zero at the steady-state in the absence of manipulated input action, while simultaneously weighing the deviation in both state and input from the origin is the quadratic stage cost, which is often used in LMPC and is formulated as follows:

$$C(\tilde{x}(t), u(t)) = x^\top Q_1 x + u^\top Q_2 u \tag{10}$$

Eq. (9b) describes the sparse-identified model that is used to predict the closed-loop states over the prediction horizon starting from the initial condition of Eq. (9c) while u is varied within the constraints defined by Eq. (9d). The last two constraints of Eq. (9e) based on the Lyapunov function, $V = x^\top P x$, guarantee that the closed-loop state either moves towards the origin at the next sampling time if the state is outside $\Omega_{\rho_{si}}$ or is contained within $\Omega_{\rho_{si}}$ for the entire prediction horizon once the state enters $\Omega_{\rho_{si}}$.

The generally nonlinear, non-convex optimization problem of Eq. (9) is solved at every sampling period, and the first entry of the optimal u^* calculated is sent to the actuator, following which the optimization is re-solved at the next sampling period using the new

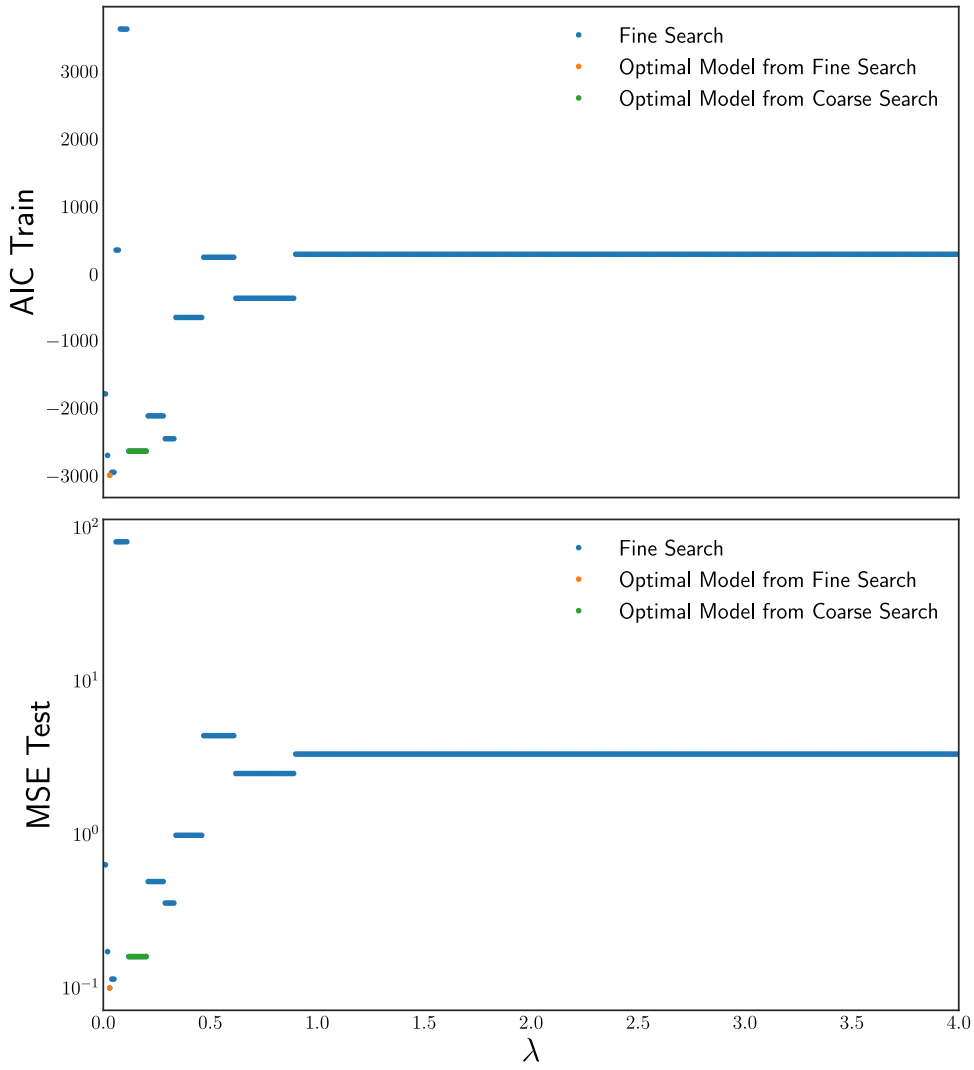

Fig. 1. Values of two error metrics, the Akaike Information Criterion (AIC) and the mean-squared error, as functions of λ for model selection.

state measurement. The optimization is solved using the numerical solver Ipopt (Wächter and Biegler, 2006) with its Python front-end named PyIpopt. For the contractive constraint of Eq. (9e), the universal Sontag control law (Lin and Sontag, 1991) or a well-tuned, stabilizing proportional-only controller may be used. It is important to note that the matrices P, Q_1, and Q_2 must be tuned for the LMPC to achieve the best results, and poorly tuned weight matrices may lead to the solver not converging to a solution within the sampling period or the maximum allowed number of iterations.

4. Reduced-order modeling for two-time-scale systems

Time-scale separation is a common phenomenon found in chemical processes such as distillation columns and catalytic continuous stirred-tank reactors (CSTRs) (Chang and Aluko, 1984). If the time-scale separation is not accounted for in a standard nonlinear feedback controller, the controller may be ill-conditioned or even unstable in closed-loop (Kokotović et al., 1999). Due to the distinct slow and fast dynamics in such systems, the process will be represented by stiff ODEs in time when using SINDy without any modification. Such stiff ODEs, when integrated with an explicit integration method such as forward Euler, require a very small integration step size to prevent divergence and yield sufficiently accurate solutions. Hence, Abdullah et al. (2021a) used the mathematical framework of singular perturbations to propose the decomposition of the original two-time-scale system into two lower-order subsystems, each separately modeling the slow and fast dynamics of the original multiscale system. Specifically, following a short transient period, the fast states converge to a slow manifold and can be algebraically related to the slow states using nonlinear functional representations. In Abdullah et al. (2021a), we applied nonlinear principal component analysis (NLPCA) developed by Dong and McAvoy (1996) to capture the nonlinear relationship between the slow and fast states, while using sparse identification to derive well-conditioned, reduced-order ODE models for only the slow states that could then be integrated with much larger integration time steps due to their numerical stability. Once the slow states are predicted with the ODE model, it is possible to use NLPCA to algebraically predict the fast states without any integration.

Nonlinear principal component analysis is a nonlinear extension of principal component analysis (PCA). PCA is a commonly used dimensionality reduction technique that finds a linear mapping between a higher-dimensional space (of the data) and a lower-dimensional space with minimal loss of information by minimizing the squared sum of orthogonal distances between the data points and a straight line. NLPCA attempts to generalize this to the nonlinear case in two steps: first, a 1-D curve that passes through the “middle” of the data points known as the “principal curve” is found; second, the principal curve is parametrized in terms of distance of each point along the curve by using a feedforward neural network with nonlinear activation functions. Overall, to make a prediction of the state of the two-time-scale system,

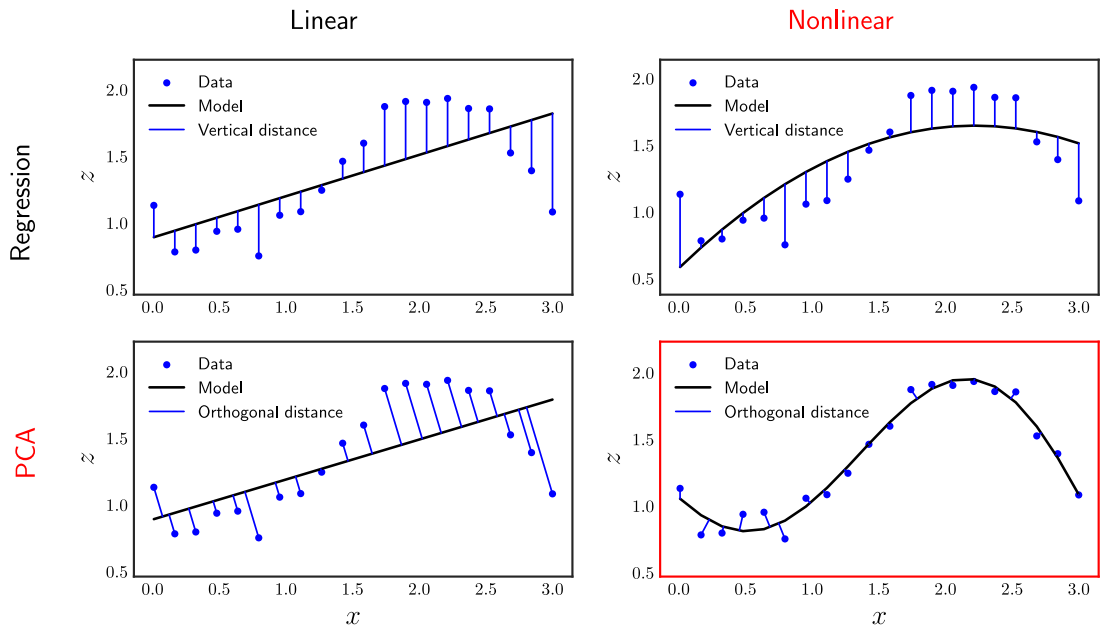

Fig. 2. Demonstration of the evolution of NLPCA based on PCA and its relation to nonlinear regression.

the measurement of the slow states at the current sampling time is passed to an explicit integrator (such as a Runge–Kutta scheme) that integrates the sparse-identified model to predict the slow states over the prediction horizon, which are then sent to the FNN to yield a prediction of the fast states.

Two-time-scale systems can be written in the form,

$$\dot{x}_s = f_s(x_s, x_f, u, \epsilon) \tag{11a}$$

$$\epsilon \dot{x}_f = f_f(x_s, x_f, u, \epsilon) \tag{11b}$$

where $x_s \in \mathbb{R}^{n_s}$ and $x_f \in \mathbb{R}^{n_f}$ denote the slow and fast states, respectively, with $n_s + n_f = n$. ϵ is a small positive parameter that represents the ratio of slow to fast dynamics of the original system. By making standard assumptions from the singular perturbation framework, the slow subsystem of Eq. (11a) can be rewritten in the form required for sparse identification,

$$\dot{\hat{x}}_s = F_{si}(\hat{x}_s, u) := \hat{f}(\hat{x}_s) + \hat{g}(\hat{x}_s)u, \qquad \hat{x}_s(t_0) = x_{s0} \tag{12}$$

where F_{si} is the sparse-identified slow subsystem.

In the first step of NLPCA, we capture the unidimensional principle curve in the n-dimensional state space to find the nonlinear algebraic relationship between the slow and fast states as shown in Fig. 2. The curve, denoted by $\mathcal{P}(\mu)$ is parametrized in terms of the ordered arc-length along the curve, μ. If $\bar{x} \in \mathbb{R}^n$ is the full-state vector, we can define the projection index $\mu_{\mathcal{P}} : \mathbb{R}^n \to \mathbb{R}$ as:

$$\mu_{\mathcal{P}}(\bar{x}) = \sup_{\mu \in \mathbb{R}} \{\mu : \|\bar{x} - \mathcal{P}(\mu)\| = \inf_{\mu' \in \mathbb{R}} \|\bar{x} - \mathcal{P}(\mu')\|\} \tag{13}$$

with μ' being a notational substitute for μ. Based on the above definition and denoting the expectation of a random variable by $\mathbb{E}$, the curve can be defined as:

$$\mathbb{E}\big(\bar{x} | \mu_{\mathcal{P}}(\bar{x}) = \mu\big) = \mathcal{P}(\mu) \tag{14}$$

where the expectation operator is approximated using a combination of scatter-plot smoothing and locally weighted regression when only discrete time-series data is available. Since the output of the first step of NLPCA, the principal curve, is a non-queryable model, an FNN is used to capture the identified principal curve.

With respect to the structure of the FNN, it is necessary to use at least one hidden layer with a sigmoid activation function, $\sigma(x) = 1/(1 + e^{-x})$, to exploit the universal approximation property of neural networks (Hornik et al., 1990; Hornik, 1991). To improve the network capability, a two-hidden-layer FNN was used in this work, as depicted in Fig. 3. The learning rate, which is the most influential hyper-parameter, requires careful tuning to obtain the optimal FNN model in the second step of NLPCA.

An LMPC that uses Eq. (12) as the process model of Eq. (9b) may be constructed. Such an LMPC will predict the slow states of the two-time-scale system and optimize the cost function based on the predicted slow states. Due to the coupled nature of the states, it is sufficient to stabilize the slow states to guarantee asymptotic stability for the entire system. However, if computational resources are available, the FNN may be used to predict the fast states, and the LMPC can then account for the full-state of the system. In Abdullah et al. (2021b), only the slow subsystem was used to ensure the LMPC optimization can be solved within every sampling period.

The primary advantage of the reduced-order model in LMPC is that the lower computational cost of the SINDy model inference, with nearly zero loss in model accuracy, directly impacts the difficulty of the optimization required to be solved by the LMPC. Hence, the LMPC based on the reduced-order SINDy model can use a longer prediction horizon, which has the potential to improve closed-loop performance in terms of faster convergence to the origin and a lower total cost function over the simulation duration, the former of which is demonstrated most clearly in the concentration profile in Fig. 4, which is based on the work in Abdullah et al. (2021b).

5. Subsampling and co-teaching in the presence of high sensor noise

A key step in the sparse identification procedure is the estimation of the time-derivatives of the states when it cannot be measured directly, as is the case in most process systems. From a survey of the literature, since the conceptualization of SINDy in Brunton et al. (2016), several advancements in the algorithm have been proposed to handle noisy data. However, most articles that investigate the effect of noise on SINDy add noise to the pre-computed derivatives (from clean data) and/or use very low levels of noise that can be easily smoothened. One example is the SINDy-PI algorithm proposed by Mangan et al. (2016) and improved by Kaheman et al. (2020). Through case studies, Kaheman et al. (2020) demonstrated that even the improved algorithm could only handle noise with a maximum

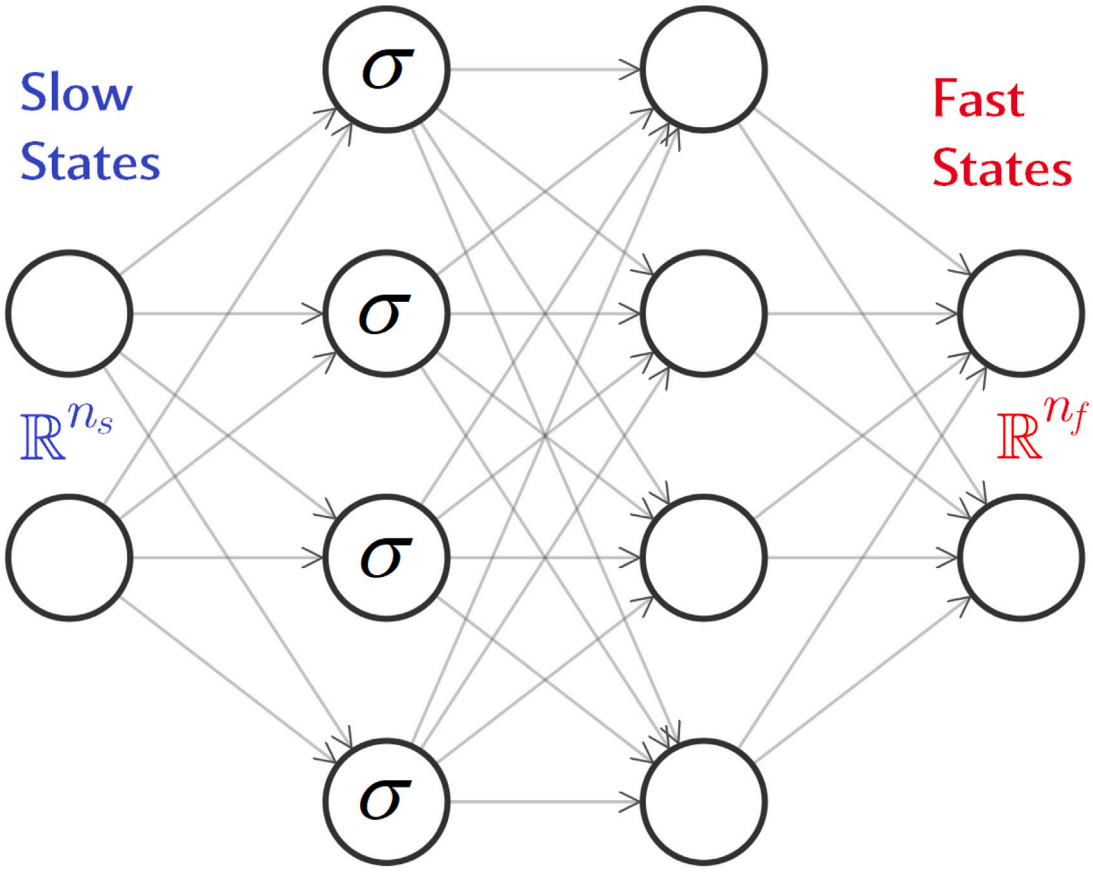

Fig. 3. Structure of the neural network used for NLPCA-SI.

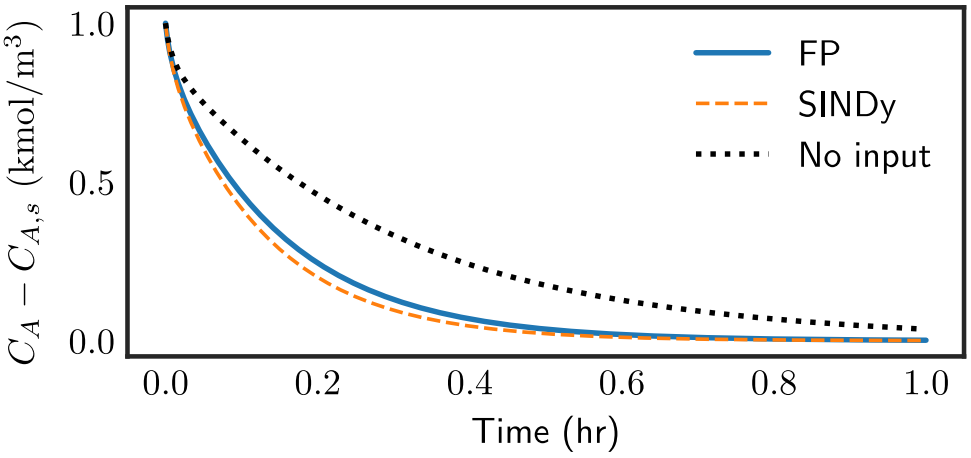

Fig. 4. Concentration (state) profile for a CSTR in closed-loop under the LMPC utilizing the first-principles (FP) model with $N = 16$ (blue line) and the SINDy slow model with $N = 24$ (orange line).

variance of 10^{-4}, which is very small in the context of process systems. Although a number of works can be found that focus on alternate approaches to build dynamical models in the presence of noise, such as Runge–Kutta time-steppers with embedded neural networks to handle the nonlinear elements (González-García et al., 1998; Fablet et al., 2018; Raissi et al., 2018; Rudy et al., 2019), these are alternatives to SINDy rather than improvements upon the original SINDy method. As a result, the methods to assist the modeling of noisy data as well as the subsequent results are largely different from SINDy and its extensions. For example, the unexpected results of Raissi et al. (2018) when using Runge–Kutta time-steppers were later explained using well-known characteristics of neural networks. More advanced time-steppers such as the work of Rudy et al. (2019) also emphasize their limitations when integrating the models from new initial conditions or attempting to capture dynamics away from a steady state, both of which are relevant in control-centric applications. While a detailed discussion of the comprehensive literature can be found in Abdullah et al. (2022b), in summary, one paper proposed an improvement upon the SINDy algorithm in the presence of moderate noise that demonstrated promise and could be developed further. This method, proposed by Zhang and Lin (2021), termed subsampling-based threshold sparse Bayesian regression (SubTSBR), involved randomly subsampling a fraction of the entire data set multiple times and selecting the best model by using a model-selection criterion. The issue of noisy data has also been

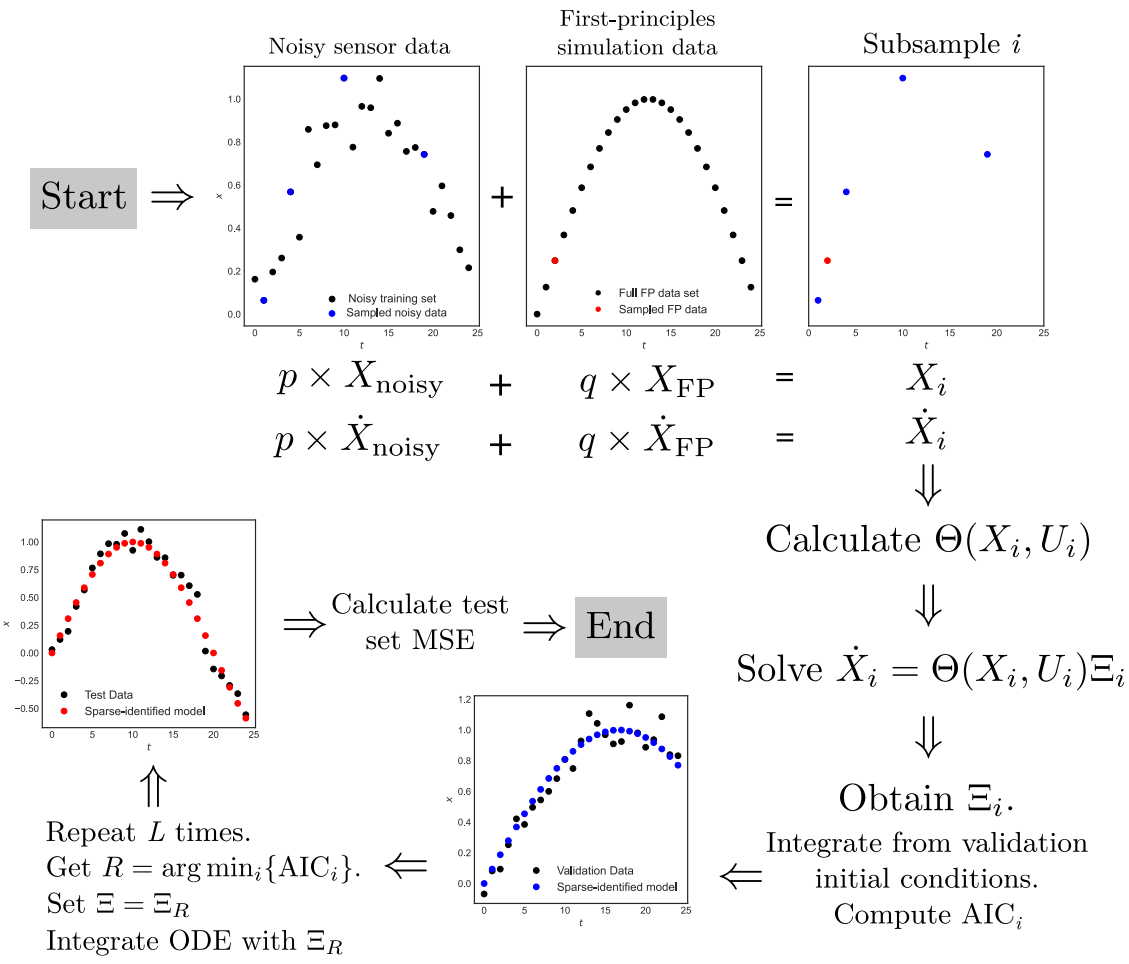

Fig. 5. Data flow diagram of subsampling with co-teaching for noisy data.

studied in the field of computer science, where fitting a neural network to noisy data often leads to the neural network overfitting the data and capturing the noisy pattern instead. A recent technique proposed to overcome this challenge is co-teaching, where a simplified first-principles process model is used to generate noise-free training data to assist the model training step by reducing overfitting. In this section, we propose a novel extension to SINDy by combining it with subsampling and co-teaching to handle highly noisy sensor data.

Subsampling is a classical statistical technique where a fraction of the total number of samples in a data set are randomly extracted and analyzed to estimate statistical parameters (Efron and Stein, 1981) or speed up algorithms (Rudy et al., 2017). However, subsampling can also be used to instead improve the modeling accuracy of SINDy when the data set is highly noisy. This is because common regression methods such as least squares utilize the complete data set by assuming that only a small fraction of the data samples are highly noisy or outliers. As a result, if the entire data set is used, the higher percentage of "good" data samples should smooth the large noise present in the data set. However, this assumption breaks down if the noise is either very high or uniformly present throughout the data set. In such a case, there are insufficient "good" data samples to smooth out the noise from the very highly corrupted data samples. In the context of SINDy, subsampling refers to selecting random fractions of the data set multiple times in order to sample only the less noisy data points for carrying out the sparse regression. The key requirement for subsampling is that the number of unknown weights to be estimated in the SINDy procedure have to be fewer than the number of total data samples available (i.e., the problem has to be overdetermined), which is the case for most practical data sets. Although as a standalone improvement, subsampling greatly improves the performance of SINDy under moderate noise levels, it is insufficient at higher noise levels, where co-teaching becomes incumbent.

Co-teaching is a method that has been used in the field of computer science, primarily in image recognition, where neural networks are trained to categorize images into pre-defined classes. However, often, a small proportion of the images in the training data set may be mislabeled, greatly deteriorating the performance of the neural network. As manually relabeling vast amounts of images is not feasible, the method of co-teaching was proposed wherein newly generated noise-free data is fed during model training to reduce the impact of the noisy data. The concept has recently been extended to regression problems, specifically the modeling of dynamical systems using long short-term memory (LSTM) networks (Wu et al., 2021a,b). The central idea of co-teaching, which was highlighted in Wu et al. (2021b), is that neural networks fit simpler patterns in the early iterations of model training, which implies that noise-free data will yield low values of the loss function, while noisy data will tend to produce high loss function values. Therefore, the training can be made more robust to noise and overfitting if the noisy data is augmented with a nonzero proportion of noise-free data from simulations of simplified, approximate first-principle models that can be derived for the complex, original nonlinear system.

Improving the sparse identification algorithm with both subsampling and co-teaching enables it to tackle consistently noisy data sets where subsampling alone is insufficient. This is because subsampling only subsamples, in the best case scenario, the least noisy data points, which are still too noisy to yield an adequate model. In the proposed method, first, a random subset of the entire data set X_{noisy} and its

corresponding $\dot{X}_{\text{noisy}}$ are sampled, which are then mixed with noise-free data generated from approximate first-principles models of the process, X_{FP} and $\dot{X}_{\text{FP}}$. The resulting mixed data set is used to solve for the unknown weights of the s terms in the SINDy function library. Once a model is identified, a model-selection criterion is used to evaluate the model performance. Three parameters must be specified in the algorithm: $p \in (0, 1)$ or the subsampling fraction, $q \in (0, 1)$ or the noise-free subsampling fraction, and $L \geq 1$, which is the number of times to independently subsample and identify a SINDy model. The algorithm randomly subsamples and mixes $p \times m$ data points from the noisy data set with $q \times m$ data points from the noise-free data set to produce the data and derivative submatrices, X_i and $\dot{X}_i$, respectively, for subsample i with $i = 1, 2, \ldots, L$. U_i are the corresponding $(p + q) \times m$ points from the input matrix U. The sparse regression equation to be solved is then

$$\dot{X}_i = \Theta(X_i, U_i)\Xi_i \tag{15}$$

where Ξ_i are the coefficients associated with each library function that is identified using the data subset $\{X_i, \dot{X}_i, U_i\}$. Once Ξ_i is determined and, therefore, the i^{th} ODE model is found, the process is repeated L times until all L models are found, following which the model selection criterion is used to extract the optimal model. An example of a model selection criterion that balances the error with the model sparsity, which is crucial for SINDy, is the Akaike Information Criterion given by the expression,

$$\text{MSE} = \frac{1}{m}\sum_{j=1}^{m}\left(x(t_j) - \hat{x}(t_j)\right)^2 \tag{16}$$

$$\text{AIC} = m \log \text{MSE} + 2L_0 \tag{17}$$

where MSE is the mean-squared error, and L_0 denotes the zeroth norm, which is equal to the number of nonzero terms in the sparse-identified model.

The hyper-parameters unique to the subsampling with co-teaching algorithm, besides the ones described in Section 3.2.3, are the values of p, q, and L. It should be noted that the goal is to capture the original noisy data rather than the noise-free data from first-principles simulations. Hence, the fraction q should generally be small, while p can be any real number between 0 and 1 as long as both metrics satisfy $p + q \leq 1$. While increasing L will generally improve the model performance because a larger number of sub-models are identified for the optimal model to be chosen from, the computational costs of increasing L must be considered. Fig. 5 shows the flow of the data throughout the algorithm.

Open-loop modeling results for a CSTR system are shown in Fig. 6, where the base SINDy model is observed to deteriorate in performance at the level of noise considered (Gaussian noise with a standard deviation of $\sigma_T = 4\,\text{K}$ in the temperature). Subsampling, even by itself, greatly improves the SINDy model performance, while subsampling with co-teaching further improves the performance. The improvement using co-teaching is most significant at the highest levels of noise considered (Gaussian noise with a standard deviation of $\sigma_T = 6\,\text{K}$ for the temperature) since models constructed using only subsampling even diverged in some cases (Abdullah et al., 2022b). Visually, the models can be assessed in terms of how close the model predictions are to the data as well as whether the states evolve in the correct direction. Although this is difficult to do for the entire simulation domain at the higher levels of noise, analyzing specific time domains in Fig. 6 can reveal differences between the models. In Fig. 6, in the ranges $t \in [2, 4] \cup [14, 16]$, the base SINDy model clearly deteriorates and deviates from the other models and the data, which is mostly concentrated much higher, near the steady-state, indicating the poor performance of the base SINDy model in these regions. The models using subsampling with and without co-teaching can be further differentiated in the regions $t \in [6, 10] \cup [12, 14]$, where the subsampling-only model predicts smaller deviations from the steady-state, but the data deviates further from the steady-state than predicted by either subsampling-based model. Therefore, the co-teaching-based subsampling model is closer to the data than the subsampling-only model in these ranges where the states deviate further from the steady-state. However, especially when dealing with noisy data, the modeling performance is best characterized quantitatively in terms of the MSE, which are shown in Table 1. The MSE for subsampling with co-teaching is consistently the lowest across all noise levels except the lowest noise level, where all methods show very similar MSE and the differences are insignificant because of the superior performance of the models from all three methods. At higher noise levels, the differences become more significant, with the subsampling-only based model even diverging when $\sigma_T = 6\,\text{K}$. At low to moderate noise levels, however, the MSE of the models using subsampling, whether with co-teaching or not are very similar. Therefore, co-teaching should be used once the model performance from using only subsampling deteriorates.

Table 1
Test set MSE for the CSTR system for four noise levels.

σ_T (K)	Base	Only subsampling	Subsampling + Co-teaching
0.4	0.01113	0.01059	0.01102
2	0.10510	0.09922	0.10370
4	0.49837	0.40037	0.36283
6	0.98210	1.89607	0.77613

6. Ensembled-based dropout-SINDy to model highly noisy industrial data sets

While subsampling with co-teaching is a viable option to tackle the issue of high sensor noise in the data measurements, the primary drawback of co-teaching is its requirement for a first-principles process model that is at least similar to the original system with respect to the dynamics and the steady-state values. However, in the case of industrial data, the dynamics may be far too complex for any theoretically derived ODE to adequately capture the system. Therefore, for the case of dealing with high levels of industrial noise, a new direction and improvement on SINDy is proposed, which is a form of ensemble learning that we term "dropout-SINDy".

Ensemble learning refers to the use of multiple models in place of one model. Homogeneous ensemble learning involves the use of the multiple models of the same type, while heterogeneous ensemble learning strategies use a combination of different types of models to improve the predictive performance. In this work, only homogeneous ensemble learning is considered. However, in the context of SINDy, even the terminology, "homogeneous ensemble learning", can refer to two distinct methods: either the data set can be subsampled to produce multiple models with the same underlying model structure, or multiple models with varying function libraries may be built using the same data set. The subsampling method described in Section 5 is an example of the former, but it was shown in Abdullah et al. (2022b) that subsampling, by itself, cannot improve the SINDy algorithm under high noise levels. In contrast, for the case of industrial noise, the proposed dropout-SINDy method uses only a fraction of the function library Θ to identify each submodel. Hence, multiple models can be identified, each with a random subset of the library. Similarly to co-teaching, this can reduce the impact of noisy data and, additionally, improve the stability properties of the SINDy models because a large number of nonzero terms (a dense coefficient matrix Ξ) can often lead to instabilities. The sparse regression equation to be solved for dropout-SINDy is similar to Eq. (15), but the state, input, and derivative data sets remain as X, U and $\dot{X}$, respectively, while only the library Θ_i and coefficients Ξ_i are varied between the n_{models} models in the ensemble, where $i = 1, 2, \ldots, n_{\text{models}}$:

$$\dot{X} = \Theta_i(X, U)\Xi_i \tag{18}$$

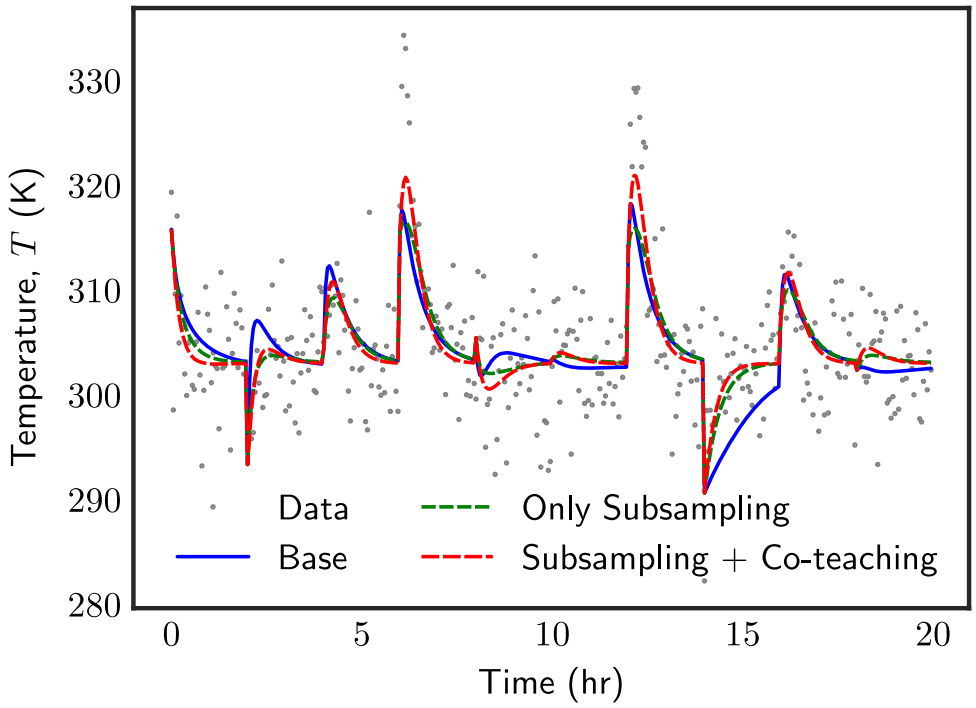

Fig. 6. Comparison of original noisy data (gray dots) with results from sparse identification without any subsampling (blue line), subsampling without co-teaching with $p = 0.2$ (green line), and subsampling with co-teaching and $p = 0.16, q = 0.04$ (red line) for the temperature T of a CSTR system.

Eq. (18) is solved using a different subset of the computed library Θ_i each time to find the corresponding set of model coefficients for the nonzeroed terms, Ξ_i. In each Θ_i, n_{dropout} library functions are randomly dropped out, with the corresponding entries in Ξ_i also being zeroed before solving Eq. (18). Once all the sub-models are found, the final model must be selected from the $\Xi_1, \ldots, \Xi_{n_{\text{models}}}$. In this case, although the mean, median, and mode are all possible methods to find the central tendency of all the Ξ_i, the mean is likely to yield dense models because even one nonzero value for a certain coefficient in any one of the sub-models will cause the coefficient to be nonzero. In contrast, the median and mode do not suffer from this. However, the mode may not be useful since even two sub-models with a zero coefficient for a library term will lead to the term being zeroed if none of the other nonzero values are repeated exactly equally, leading to excessive sparsity. Hence, the median is determined to be the most reasonable measure of central tendency for dropout-SINDy.

The number of functions of the candidate library to be omitted in each model, n_{dropout}, as well as the number of models, n_{models}, must be tuned when building a dropout-SINDy model. A very small value of n_{dropout} implies that the sub-models in the ensemble are very similar to the base SINDy model without any dropout, negating most if not all performance gains of the proposed method. But if n_{dropout} is too large, excessive sparsity will lead to models that lack the complexity required to capture the dynamics. Similarly, a small value of n_{models} may lead to the optimal model not being identified as the search is conducted over a smaller set, but increasing n_{models} also increases computational costs and might even promote instability if the median of the model coefficients is shifted by a larger proportion of poor models. Hence, this balance between computational cost, model improvement, model complexity, and stability must be considered when tuning n_{models} and n_{dropout} when using dropout-SINDy. The data flow throughout the algorithm is outlined in Fig. 7.

In this section, "industrial" data refers not to an experimental data set but data generated from a chemical process simulated in the high-fidelity chemical process simulator, Aspen Plus Dynamics, which is a widely used simulator in the chemical sector that has been used to build steady-state and dynamic simulations of chemical processes to aid chemical engineers in process design and optimization. Chemical process simulators have several advantages over first-principles models as they contain numerous built-in packages to handle most common unit operations, thermodynamic properties, molecular interactions, etc., which result in significantly more accurate models that more closely represent the plant process dynamics. In Abdullah et al. (2022a), Aspen Plus Dynamics was used to build the process flow diagram shown in Fig. 8, which was then used for both data generation as well as closed-loop simulations in order to imitate the industrial process.

Table 2
Parameter values for nonisothermal CSTR example.

$F = 5.0\,\text{m}^3/\text{h}$	$V = 1.0\,\text{m}^3$
$k_0 = 8.46 \times 10^6\,\text{m}^3\,\text{kmol}^{-1}\,\text{h}^{-1}$	$E = 5.0 \times 10^4\,\text{kJ/kmol}$
$R = 8.314\,\text{kJ}\,\text{kmol}^{-1}\,\text{K}^{-1}$	$\rho_L = 1000.0\,\text{kg/m}^3$
$\Delta H_r = -1.15 \times 10^4\,\text{kJ/kmol}$	$T_0 = 300\,\text{K}$
$C_{A0_s} = 4\,\text{kmol/m}^3$	$Q_s = 0\,\text{MJ/h}$
$C_{A_s} = 1.95\,\text{kmol/m}^3$	$T_s = 402\,\text{K}$
$C_p = 0.231\,\text{kJ}\,\text{kg}^{-1}\,\text{K}^{-1}$	

When using the basic SINDy algorithm to model the highly noisy industrial data from Aspen Plus Dynamics, it is found that basic SINDy is unable to model the dynamics or even correctly predict the final steady-state of the open-loop system, the latter of which greatly affects the performance of a controller. However, when dropout-SINDy is used on the industrial data set, it is able to capture most of the dynamics and correctly predict the final steady-state values of the states. When an MPC is designed with the dropout-SINDy model, it can be demonstrated to achieve closed-loop stability and converge to the steady-state faster and with less energy and overshoot than a corresponding proportional-controller as shown in Fig. 9.

7. Demonstration of the use of SINDy to model a nonlinear chemical process

In this section, the modeling of a highly nonlinear CSTR operating at an unstable steady-state using SINDy is considered.

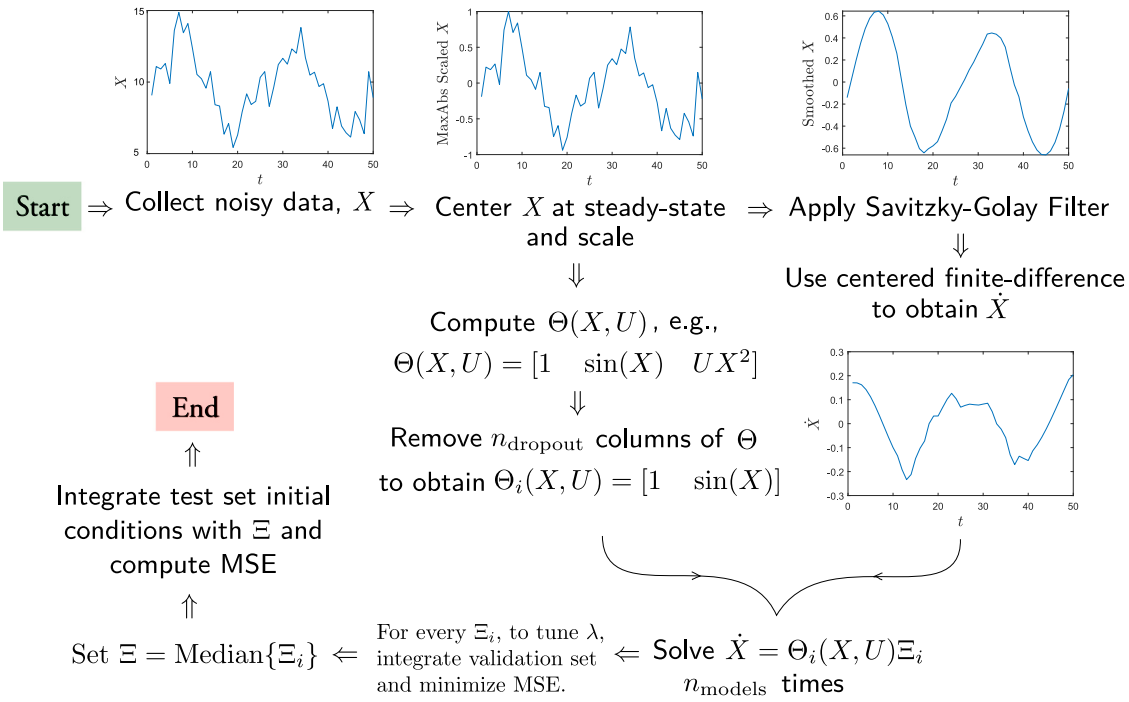

Fig. 7. Data flow diagram of Dropout-SINDy for noisy, industrial data.

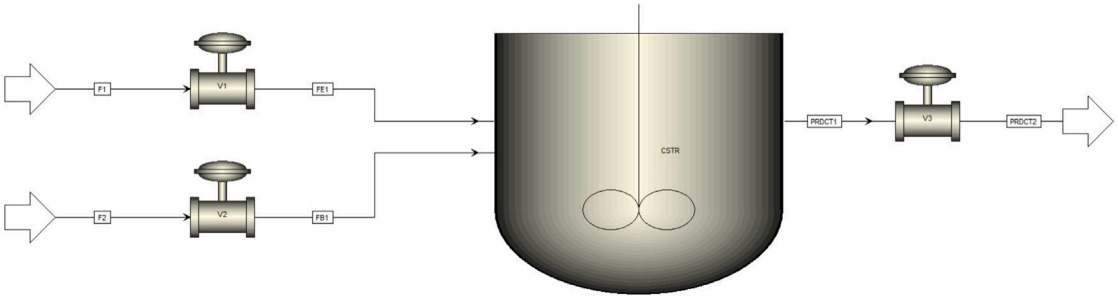

Fig. 8. Aspen Plus model process flow diagram of an ethylbenzene production process.

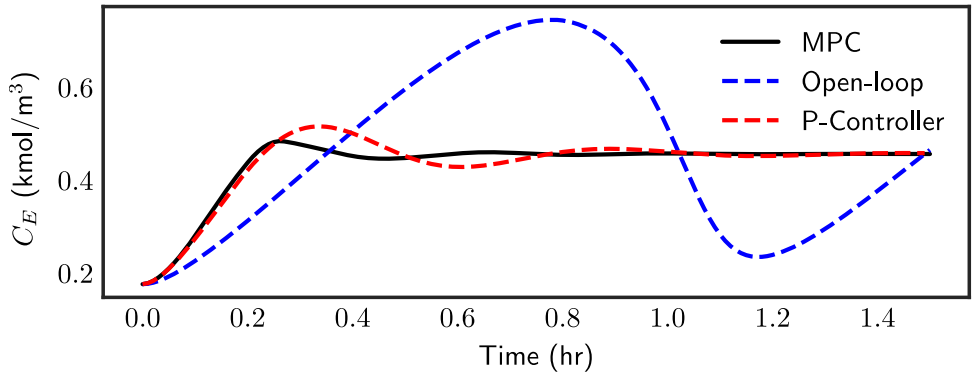

Fig. 9. State and input profiles for a CSTR in closed-loop under no control (blue line), a P-controller (red line), and the LMPC utilizing the dropout-SINDy model (black line) throughout the simulation period $t_p = 1.5$ h.

Specifically, a perfectly mixed, nonisothermal CSTR where an irreversible, exothermic reaction with second-order kinetics, $A \xrightarrow{k} B$, takes place is studied. The rate constant of the reaction, k, is not assumed to be constant and, instead, an Arrhenius relation of the following form is used to determine the rate constant as a function of the Kelvin temperature, T:

$$k = k_0 e^{-\frac{E}{RT}} \tag{19}$$

where k_0, E, and R represent the pre-exponential constant, activation energy of the reaction, and the ideal gas constant, respectively. Using material and energy balances, the differential equation model describing the CSTR dynamics is derived as follows:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{-\frac{E}{RT}} C_A^2 \tag{20a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{-\frac{E}{RT}} C_A^2 + \frac{10^3 Q}{\rho_L C_p V} \tag{20b}$$

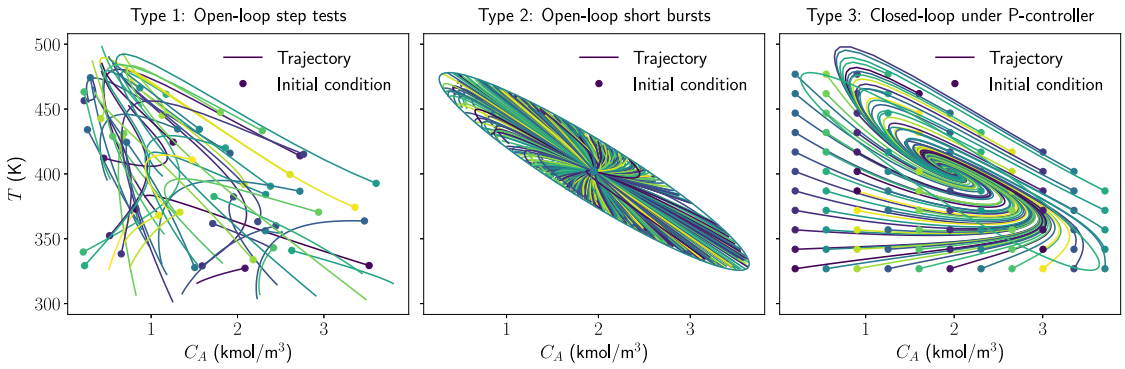

Fig. 10. Three types of data generation for the nonisothermal CSTR operating at an unstable steady-state.

where C_A, V, and T denote the concentration of reactant A in the reactor, the volume of the reacting liquid inside the reactor, and the time-varying absolute temperature of the reactor. The concentration of species A in the inlet stream, the inlet temperature, and the volumetric flow rate fed to the reactor are represented by C_{A0}, T_0, and F, respectively. A heating jacket supplies/removes heat to/from the CSTR at a rate of Q. The density and heat capacity of the reacting liquid are assumed to have constant values of ρ_L and C_p, respectively, while ΔH denotes the enthalpy of the reaction. The values of the process parameters are provided in Table 2. With the values from Table 2 substituted into Eq. (20), the exact ODE model to be identified using SINDy can be found to be

$$\frac{dC_A}{dt} = 5C_{A0} - 5C_A - 8.46\times 10^6 e^{-\frac{6013.95236949723}{T}} C_A^2 \tag{21a}$$

$$\frac{dT}{dt} = 1500 - 5T + 4.211688\times 10^8 e^{-\frac{6013.95236949723}{T}} C_A^2 + 4.3Q \tag{21b}$$

The objective is to build a SINDy model for the CSTR system of Eq. (20), ideally for the entire state-space or at least a large region of the state-space around the desired operating point, which is the unstable steady-state, $(C_{A_s}, T_s) = (1.95\,\text{kmol/m}^3, 402\,\text{K})$. The factors that most significantly impact the quality of the SINDy model for this system were found to be the data generation and the candidate library of basis functions considered for $\Theta(X, U)$, both of which are discussed in detail over the next two subsections. To compare models quantitatively, for the sake of brevity, rather than reporting every model obtained from each data generation method or candidate library considered, in the rest of this section, the maximum absolute error in the Kelvin temperature will often be reported because the errors in the temperature are larger in terms of absolute value and intuitively understood.

Remark 1. Due to the explicit nature of SINDy models, once the ODE models are obtained from SINDy, incorporating them into an MPC is generally straightforward. The challenge of SINDy-based MPC, however, lies in the modeling rather than MPC implementation, as opposed to entirely black-box approaches such as recurrent neural networks and other deep learning models, which can approximate practically any input/output data if provided with sufficient data and tuned thoroughly, but can encounter computational and technical challenges when implemented in closed-loop MPC. Hence, this section focuses solely on the modeling of the nonlinear CSTR, since past works (Abdullah et al., 2021b, 2022b) have already demonstrated the application of SINDy models in MPC with open- and closed-loop simulation results for a diverse array of systems. The goal of this section is to familiarize the reader with the intricacies of building SINDy models in a chemical engineering paradigm.

7.1. Data generation

For system identification, the data set used to identify the system is a crucial element. Hence, the data generation must be carried out in a practical method while also providing sufficient dynamic information for an algorithm to capture. Therefore, all simulations of Eq. (20) were carried out using an integration step size of $h_c = 10^{-4}$ h and sampled every $\Delta = 0.01$ h (36 s), which is a reasonable sampling period for such a chemical process. The simulations were carried out for a duration of $t_f = 1$ h since most trajectories reached a steady-state within 1 h of simulation duration.

Due to the various ways that one may generate or obtain data for this system, three types of data generation were carried out, and each data set was then used to attempt to build SINDy models. Representative trajectories for each data set are shown in Fig. 10. The types of data generation and their advantages/disadvantages are summarized as follows:

1. Method: Open-loop step tests are carried out using numerous, random initial conditions and input signals until a steady-state is reached.
   - 1000 such trajectories were generated in this data set.
   - Initial conditions were randomly selected with the following restrictions on the initial states: $C_A \in [0.2, 3.7]$ kmol/m^3 and $T \in [327, 477]$ K
   - Input signals were randomly generated with the following restrictions on the inputs: $C_{A0} \in [0.5, 7.5]$ kmol/m^3 and $Q \in [-500, 500]$ MJ/h
   - This is a standard method of data generation within chemical engineering in simulations-based applications as well as experimental practices. As an established method, data generation via this method is easily conducted, a wide area of the state space can be covered by exciting the input signals as desired, and a large amount of dynamic information is present in the data set.
   - Due to the operating region being the unstable steady-state, the trajectories, being in open-loop, will settle at the stable steady-states. However, this was not found to deteriorate the performance, likely because the dynamics of the reactor itself are independent of the region.
   - As the states may achieve extreme values when the input signals are varied too widely (such as temperatures as high as 1000 K or as low as 1 K for certain excessively large/small values of Q), the best practice is to limit the range of input signals when using this method of data generation. This is particularly important when using finite-differences to estimate $\dot{X}$, which is the only estimation method available in a practical setting. It was found that when data generated indiscriminately including trajectories that settle at 1000 K or 1 K were included in the training data set, i.e., all 1000 trajectories were used in training, SINDy had difficulties identifying the correct model if the derivatives were estimated with finite-differences. If the exact derivatives were provided (which

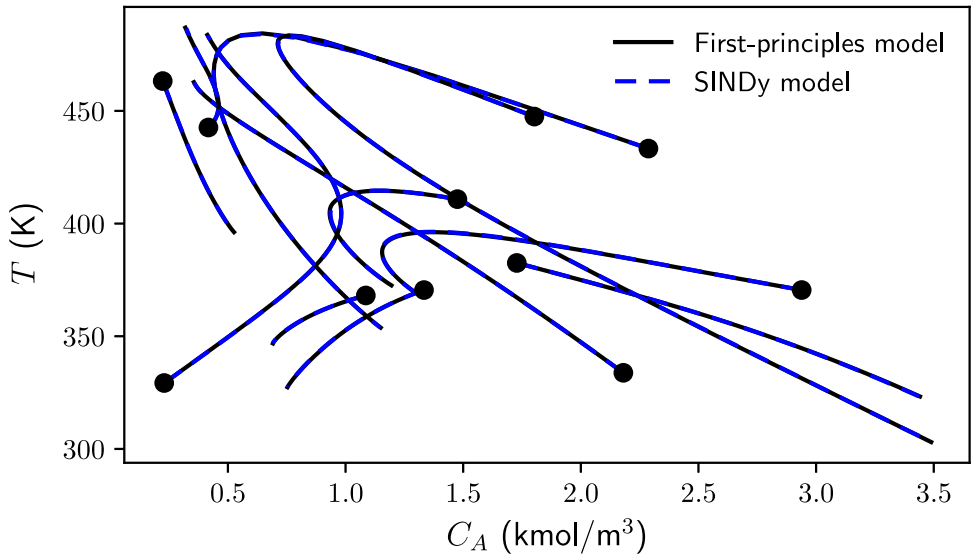

Fig. 11. State-space profiles for open-loop simulation using the first-principles model of Eq. (20) and the SINDy model obtained using type 1 data generation, respectively, for various sets of inputs and initial conditions (marked as solid dots) x_0 in the vicinity of the desired operating point.

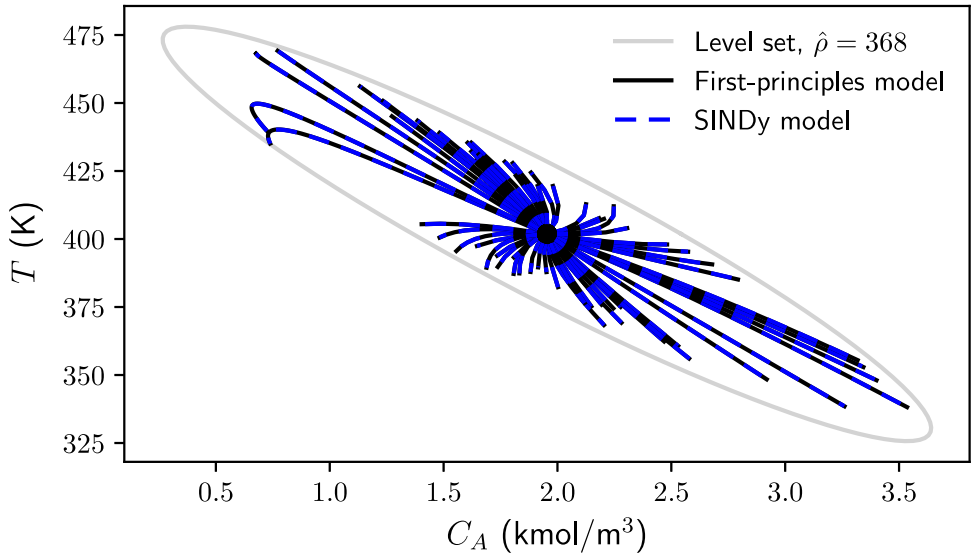

Fig. 12. State-space profiles for open-loop simulation using the first-principles model of Eq. (20) and the SINDy model obtained using type 2 data generation, respectively, for various sets of inputs, starting from the steady-state.

would not be available in most chemical engineering applications), then SINDy was able to identify the model correctly. Upon further analysis of the derivatives at the regions of the fastest dynamics, it was found that the states changed very abruptly within the sampling period of $\Delta = 0.01$ h, causing numerical instabilities in the derivative estimation. Hence, providing the exact derivatives resolved the issue. As expected, the issue was also resolved if the data was sampled ten times as frequently, i.e., with $\Delta = 0.001$ h. However, using all the 1000 trajectories is not necessary to capture the dynamics of this system, as described next.

- When the data set was truncated to only retain trajectories that never exceeded a temperature or 500 K or dropped below 300 K, i.e. $T \in [300, 500]$ K $\forall\ t$, in order to only select trajectories close to the desired steady-state, 53 out of the initial 1000 trajectories were retained. However, SINDy was able to identify the best model with these 53 trajectories, producing a model with a maximum absolute error in the temperature of only 0.5 K. A few representative open-loop simulations (i.e., the test set) for the first-principles model and this sparse-identified model are shown in Fig. 11, demonstrating close agreement throughout the region of state-space. Hence, it can be concluded that 53 trajectories contain sufficient dynamic information to build a highly accurate SINDy model, and there is no necessity to use all 1000 trajectories, which introduce faster and more complex dynamics in certain regions, which eventually require a finer sampling to be practically useful.

2. Method: Open-loop step tests are carried out with the system initiated from the desired steady-state and excited using various input signals only until the desired level set, $\Omega_{\hat{\rho}}$, or operating region is excited.

    - 1000 such trajectories were generated in this data set.

- The initial condition was fixed to be the unstable steady-state, $(C_A, T) = (C_{A_s}, T_s) = (1.95\,\text{kmol/m}^3, 402\,\text{K})$
- Input signals were randomly generated with the following restrictions on the inputs: $C_{A0} \in [0.5, 7.5]$ kmol/m^3 and $Q \in [-500, 500]$ MJ/h
- As data is generated only within the operating region of interest, an advantage is that almost the entire region of the state-space that is of interest can be captured via a large number of simulations.
- Due to the unstable nature of the steady-state, one disadvantage is that a very large number of the 1000 trajectories in the data set are incomplete and too short to provide sufficient dynamic information, especially for SINDy, which generally performs better with longer trajectories rather than short bursts of trajectories. Out of the 1000 trajectories, only 14 trajectories are able to be simulated until $t_f = 1$ h. Since second-order finite-differences are used for the gradient approximation in our work as well as due to the internal mechanisms of the integrator used, at least 4 data points are required to be able to use a trajectory for model identification. Only 381 of the 1000 trajectories had at least 4 data points and could be used to build a SINDy model. However, the data set of 381 trajectories did not contain enough dynamic information, producing a SINDy model with a maximum temperature prediction error of 10.4 K. However, if the size of the data set was increased to 2000 trajectories, 807 trajectories with at least 4 data points remained, which then produced a highly accurate SINDy model with a maximum temperature prediction error of 0.6 K.
- State-space profiles for some open-loop simulations are shown in Fig. 12 for the first-principles model and the identified SINDy model, showing close agreement throughout.

3. Method: Closed-loop simulations are carried out under a proportional-only controller with the state initialized from various initial conditions.

    - Two data sets were used to attempt to build a SINDy model using this method of data generation, one data set with 121 trajectories, spanning an 11 × 11 grid for x_0 in the state-space, while the second data set consisted of 961 trajectories, covering a 31 × 31 grid for x_0 in the state-space.
    - Initial conditions were selected within the grid, $C_A \times T = [0.2, 3.7]$ kmol/m^3 × $[327, 477]$ K with each range uniformly spaced into 10 or 30 intervals with 11 or 31 points, respectively.
    - Input signals were calculated using the equation for a proportional controller, $Q = -1000(T - T_s)$, where 1000 represents the controller gain, and C_{A0} was fixed at its steady-state value of $C_{A0_s} = 4\,\text{kmol/m}^3$.
    - A purported advantage of this method of data generation is that, due to the presence of the controller, the state can be driven to the desired unstable steady-state from any initial condition, providing dynamic information for trajectories from any point in the state-space up to the unstable steady-state.
    - The models obtained using this data set could very accurately predict the derivatives within the test set, i.e., the right-hand side of the model evaluates to the correct value of the derivative of the test set trajectories. However, all the simulations diverged from the steady-state after a short period at the initial stages of the simulation duration. This phenomenon was also observed in Brunton et al. (2016) with the glycolytic oscillator model and attributed to the identification of wrong basis terms for some of the variables. In the models obtained for Eq. (20) using SINDy with the data set generated using closed-loop simulations, the heat input rate, Q, erroneously appeared with a relatively large coefficient in the first ODE representing $\dot{C}_A$, which may be the cause of the divergence. While further analyses may allow such data to be used for SINDy model identification, based on our current results, this method of data generation was not found to produce accurate SINDy models.

Based on the above analysis, the first method of data generation was found to be the optimal method of data generation when using SINDy. Since the optimal results were obtained with limited trajectories that were able to be integrated to t_f and also stayed relatively close to the steady-state of interest, the best method of data generation for this system, based on the above analysis, seems to be conduct a modest number of step tests near the desired region. However, the second method can also be used if a much larger data set is used and caution is taken to only use trajectories with at least 4 data points when using a second-order finite-difference method for estimating the time-derivative of the states. The use of closed-loop data to identify SINDy models was not found to yield satisfactory results, and further analyses should be carried out in the future to assess the viability of such data for SINDy modeling of chemical processes.

7.2. Candidate library of basis functions

Since its inception, multiple studies have reported the central role of the candidate library, Θ, in the SINDy algorithm (Brunton et al., 2016; Kaheman et al., 2020). In Brunton et al. (2016), for example, a standard benchmark problem for system identification, the glycolytic oscillator model, could only be partially identified, i.e., the dynamics of only four out of the seven states could be correctly identified. The reason was attributed to the presence of rational functions in the right-hand sides of the ODEs corresponding to the remaining three states, which were not considered in the polynomial basis set used. Hence, choosing the correct basis functions is critical to the success of SINDy. For the remainder of the section, the data set used to study the effect of the candidate library is the data set generated using the first type of data generation described in Section 7.1 (53 trajectories from open-loop step tests).

In the absence of any *a priori* knowledge, the nonisothermal CSTR of Eq. (20) is a particularly challenging system to obtain the correct basis for and, hence, model. This is primarily due to the fact that, by design, SINDy can only regress the pre-multiplying coefficients for each basis function, which appear linearly in the right-hand side of the ODE. The basis functions themselves must be selected and the Θ calculated before carrying out the regression step for identifying the pre-multiplying coefficients by solving Eq. (8). Since the activation energy is generally unknown, the exponential term must be carefully selected. For the set of parameters chosen, from Eq. (21), it can be observed that the numerator of the argument of the exponential term is −6013.95236949723. Due to the extreme dissimilarity between $e^{-\frac{1}{T}}$ and $e^{-\frac{6013.95236949723}{T}}$, using the former exponential term as a basis function will not yield an accurate SINDy model. The dynamics of the $e^{-\frac{6013.95236949723}{T}}$ term cannot be captured by any linear multiple of $e^{-\frac{1}{T}}$. However, choosing large numbers over a wide range is also inadvisable since only a narrow range of the exponent can yield an accurate model with a maximum absolute error in the temperature below 1 K as shown in Fig. 13. While it may be possible to tune the exponent in this particular case by conducting a fine search for the exponent over a wide range with shortly spaced intervals of approximately 10 units, this is generally intractable when the exponent is even larger in magnitude (increasing the required search region) or if there are multiple reactions, in which case tuning each exponent term using a multidimensional grid search

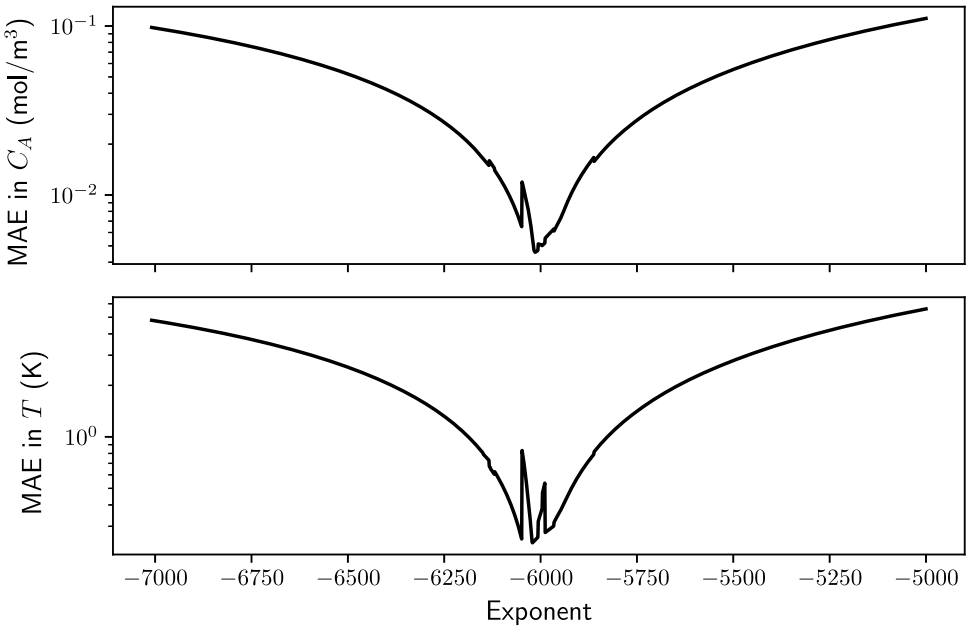

Fig. 13. Validation error as a function of the numerator of the argument of the exponential function in the candidate library for the original data set, (C_A, T).

at such a high resolution becomes prohibitively expensive. Therefore, two approaches are proposed to overcome this challenge, both of which are shown to yield accurate SINDy models.

Remark 2. This challenge has been overcome in some past studies by assuming that the activation energy is known *a priori*, and the exact term, $e^{-\frac{6013.95236949723}{T}}$, is included in the candidate library, largely simplifying the modeling problem (e.g., Bhadriraju et al., 2019, 2020). In other studies using SINDy to model reaction networks, the objective was to identify the reactions rather than investigate any temperature dependence (Hoffmann et al., 2019). Hence, the specific challenge of obtaining an appropriate basis for SINDy to model nonisothermal reactors is considered here.

7.2.1. Non-dimensionalization of the temperature

The first approach we consider is non-dimensionalizing the temperature by scaling it by a reference temperature, T_{ref}. We consider, for simplicity and without loss of generality, $T_{\text{ref}} = T_s$, and define the new dimensionless temperature as $\bar{T} = T/T_s$. Hence, the ODE system of Eq. (20), after the appropriate substitutions, takes the form,

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{\gamma}{\bar{T}}} C_A^2 \tag{22a}$$

$$\frac{d\bar{T}}{dt} = \frac{F}{V}\left(\frac{T_0}{T_s} - \bar{T}\right) + \frac{-\Delta H}{\rho_L C_p T_s} k_0 e^{\frac{\gamma}{\bar{T}}} C_A^2 + \frac{10^3 Q}{\rho_L C_p V T_s} \tag{22b}$$

where $\gamma = -E/RT_s$. For the set of process parameters and reference temperature chosen, $\gamma = -14.96$. Due to the much smaller value of γ and the lower sensitivity of γ, it is possible to conduct a fine search for a value of γ that produces an accurate SINDy model. The maximum absolute errors in the variables for the validation set for $\gamma \in [-20, 0]$ are shown in Fig. 14. A value of $\gamma = -15$ yields the highly accurate model,

$$\frac{dC_A}{dt} = 5.051 C_{A0} - 5.058 C_A - 8.8 \times 10^6 e^{-\frac{15}{\bar{T}}} C_A^2 \tag{23a}$$

$$\frac{d\bar{T}}{dt} = 3.768 - 5.046\bar{T} + 1.09 \times 10^6 e^{-\frac{15}{\bar{T}}} C_A^2 + 0.011 Q \tag{23b}$$

where every coefficient is within 5% of the true values. There are two further advantages of non-dimensionalization in this case. Firstly, when multiple reactions are present, in many practical cases, since the reference temperature is similar to the specific process temperatures, all the γ will often be approximately of the same order of magnitude or within an order of magnitude difference (e.g., Alanqar et al., 2017a). Therefore, a "mean" or representative value of the γ values of all the reactions will produce an accurate SINDy model, owing to the greatly reduced sensitivity of the basis functions to γ. Secondly, even if the (nearly) exact value of $\gamma = -15$ is not found using the search methodology described, simply using every integer value of $\gamma \in [-20, -10]$ to create 11 basis functions also yields an accurate (but dense) SINDy model with a maximum absolute error in the temperature of 0.4 K. In contrast, in the original variables, if multiple basis functions, for example, the set $\gamma \in \{-7000, -6000, -5000, -4000, -3000, -2000, -1000\}$ is chosen to be in the candidate library, the results are poor due to the large dissimilarities between successive basis functions as noted previously. We reiterate that the goal of using SINDy in this work is not to capture the underlying ODE but to use SINDy as a system identification method. Although the original system was nearly reproduced when $\gamma = -15$ was correctly identified, this need not be the case to apply SINDy, especially when the models will subsequently be used for model-based feedback control. Hence, the "bruteforce" approach of using all 11 basis functions with $\gamma \in [-20, -10]$ is considered a satisfactory model as well. This latter approach may also handle multiple reactions more easily since it is likely that the correct value of γ for each reaction is captured in the candidate library.

Remark 3. To apply non-dimensionalization to the system when applying SINDy, the only change that must be made is that the temperature data must be scaled by T_s before providing the data set to the SINDy algorithm. Since finite-differences are used to estimate the time-derivative, $\dot{X}$, the derivative estimates will scale accordingly once the data set itself is scaled.

7.2.2. Higher-order Taylor series approximation

A possibly more general approach that can handle any value of the activation energy or any number of reactions is to express the exponential term using its Taylor series expansion such that the activation energy appears as a pre-multiplier, which can then be regressed using SINDy. As SINDy is a nonlinear method, any order of the Taylor series can be retained. If multiple reactions are present, the pre-multiplier should account for all the reactions since the temperature variable is independent of the activation energy, and all the approximated terms

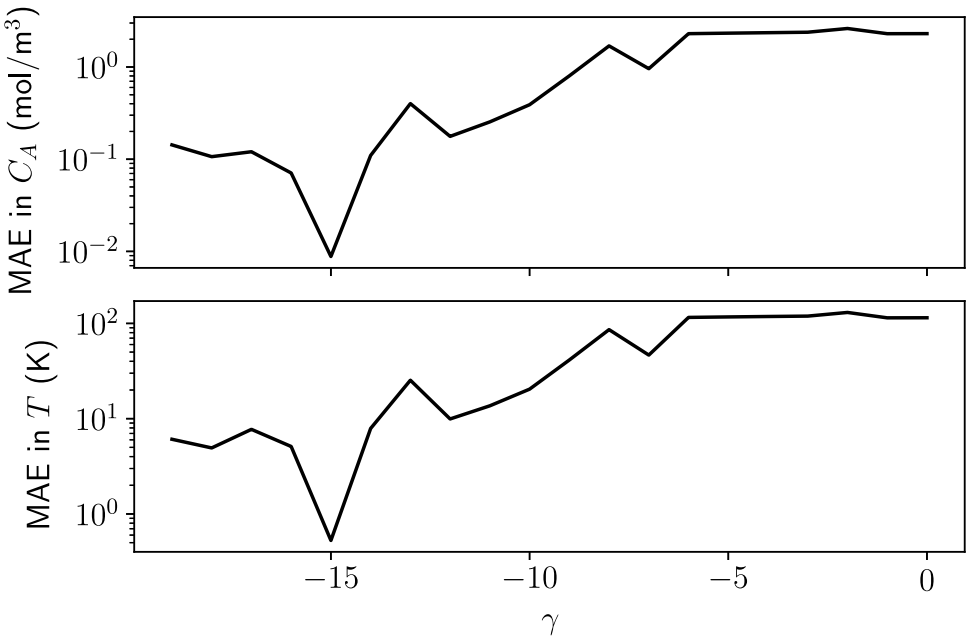

Fig. 14. Validation error as a function of γ for the data set with dimensionless temperature, $(C_A, \bar{T})$.

can be summed to yield one final pre-multiplying coefficient value for each term of the Taylor expansion.

Due to the length of the models involving Taylor expansion, only error metrics and discussions are provided for this method. When the candidate library includes up to 5th-order terms of the Taylor expansion, i.e., $(T - 402)^5$, an accurate SINDy model with a maximum absolute error in the temperature of 0.4 K is obtained, with open-loop test results nearly identical to Fig. 11. When the sparse-identified model is compared to the original ODE of Eq. (21) with parameters substituted in and the exponential term replaced by 5th-order Taylor series, it is found that the SINDy model neglects terms above third-order, which are of the order of 10^{-8} and 10^{-6} for C_A and T, respectively, in the actual equation (i.e., when a 5th-order Taylor expansion of the exponential term is used in the first-principles system of Eq. (21)). As for terms up to third-order, the SINDy model correctly identifies all terms for C_A and identifies the terms in T correctly as well, but also identifies a few erroneous terms such as linear C_A and C_{A0} terms. However, the contribution of the extra terms are extremely minor and do not affect the accuracy, as seen in the extremely low maximum absolute error.

Remark 4. While this method may be reminiscent of linearization of a nonlinear ODE, there are two key differences. Firstly, a nonlinear higher-order Taylor expansion is used to approximate the exponential function rather than a linear approximation. This greatly affects the region of accurate model predictions compared to a linearized model. When the open-loop tests shown in Fig. 12 were repeated with the linear state-space model obtained for this system in Wu et al. (2019b) using N4SID, all trajectories were found to diverge, while Fig. 12 demonstrates the high accuracy of the nonlinear SINDy model. Secondly, only the exponential term in the Arrhenius relationship is approximated using the Taylor expansion, but the remaining terms in the ODE model and candidate library remain in their original, nonlinear forms. Hence, all other nonlinear terms can still be identified exactly without any approximation, while model linearization includes linearizing even such polynomial and trigonometric terms.

7.3. Summary of data generation and candidate library guidelines and final steps to build the SINDy model

Based on extensive results from using the various types of data generation and basis functions considered, the following points can be summarized:

1. Open-loop step tests were found to be the optimal method of data generation for obtaining a SINDy model for the system studied, although short bursts within the desired stability region can yield a good model if a sufficient number of trajectories with at least 4 data points are obtained.
2. Data from closed-loop simulations did not yield an accurate model for the system studied.
3. A sampling period of $\Delta = 0.01$ h or 36 seconds is sufficient for obtaining an accurate SINDy model as long as enough dynamic information is captured via open-loop tests with a large number of input signals.
4. Due to the sensitivity of the argument of the exponential term, $-E/R$ or γ, the exponential basis term should be selected carefully.
    - Specifically, if *a priori* knowledge of the reaction (such as an estimate of the activation energy) is available, the system may be modeled directly without any modifications as long as the correct values of the activation energy are used to build the candidate library.
    - In the event that the no *a priori* knowledge is available, the system should be either non-dimensionalized with respect to temperature or a higher-order Taylor series used to approximate the exponential terms.
5. Non-dimensionalization of the temperature has potential to reproduce the exact system.
6. Using Taylor series approximations of the exponential term can yield highly accurate SINDy models, but their performance is expected to deteriorate when sufficiently far from the point of expansion. However, since a nonlinear, higher-order approximation is used, the region where the model performs accurately will be significantly larger than any model obtained from a linearization of the original system, and likely large enough for any practical application.

Once the data set and method of handling the exponential term are finalized based on the aforementioned guidelines, the SINDy model is obtained by using the PySINDy package in Python (de Silva et al., 2020; Kaptanoglu et al., 2022). Specifically, the data set is loaded into Python and split into an 70%/10%/20% training/validation/test set. The time-derivative of the states, $\dot{X}$, is estimated using second-order central finite differences (except the first and last points, which use second-order forward and backward finite differences, respectively). The optimizer is chosen to be the sequential thresholded least squares described in Brunton et al. (2016), with λ tuned via a coarse search to a value of 5.0, although similar results were obtained for the SR3 optimizer as well. The candidate library, for both the non-dimensionalization and Taylor series approaches, was chosen to include up to second-order polynomial terms for the concentration C_A, the bias term, and linear input terms. The remaining terms for the non-dimensionalization method included a linear temperature term, the exponential term with $\gamma = -15$, and interaction terms between the polynomial C_A terms and the exponential term. Specifically, the candidate library for the non-dimensionalization approach takes the following form:

$$\Theta(C_A, \bar{T}, C_{A0}, Q) = [1 \quad C_A \quad C_A^2 \quad \bar{T} \quad C_{A0} \quad Q \quad e^{-\frac{15}{\bar{T}}} \quad C_A e^{-\frac{15}{\bar{T}}} \quad C_A^2 e^{-\frac{15}{\bar{T}}}] \tag{24}$$

For the Taylor series approach, the only change is that the exponential term is replaced with $(T - 402)$, $(T - 402)^2$, …, $(T - 402)^5$. Hence, the last three functions and the $\bar{T}$ function in Eq. (24) are replaced by 15 terms (five exponential approximation terms and ten interaction terms with C_A), producing a library of 20 functions. Once all of the above selections are made, the SINDy model can be obtained by calling the model fitting method in PySINDy. The SINDy model of Eq. (23), for example, is obtained by using the first type of data generation (53 open-loop step tests) and the candidate library of Eq. (24).

8. Future directions

8.1. Neural network basis functions

For highly complex systems, it may be possible that the initially chosen nonlinear basis functions do not produce adequate results, but no prior knowledge is available to intelligently expand the function library. Moreover, adding random, additional nonlinear candidate functions may fail to improve the SINDy model performance if the functions added are completely dissimilar to the relevant functions that are required to model the system. An example is the challenge of the exponential basis term encountered and discussed in Section 7.2 with the nonisothermal CSTR example. In such cases, one option is to add more powerful and general function approximators such as feedforward neural networks, which are well-known for their universal approximation property, which dictates that they can approximate any static nonlinear function if they are designed with enough neurons and at least one sigmoidal hidden layer (Hornik et al., 1990; Hornik, 1991). Such hybrid models consisting of partly first-principles/ODE models and partly data-based black-box models are increasingly being used (Porru et al., 2000; Oliveira, 2003; von Stosch et al., 2014; Zendehboudi et al., 2018; Bangi and Kwon, 2020; Lee et al., 2020). Specifically, hybrid models involving ODE models and FNNs have been successfully applied to state estimation problems in the recent work of Alhajeri et al. (2021). Therefore, a similar approach may be proposed for SINDy, where the right-hand side of the SINDy model of Eq. (2) may be modified to

$$\dot{\hat{x}}(t) = \hat{f}(\hat{x}) + \hat{g}(\hat{x})u + \text{FNN}(x, u) \tag{25}$$

where FNN denotes a feedforward neural network model that can capture any nonlinearities not modeled by the function library. One advantage of such a model, as opposed to a purely FNN model for the right-hand side of Eq. (25), may include reduced computational time due to the requirement of simpler models with fewer parameters, since only a fraction of the model must be captured by an FNN. Moreover, neural network training generally requires large volumes of data with wide variation and coverage of the operating region, which may be difficult to obtain in an experimental or plant setting. In contrast, when only a fraction of the overall model requires an FNN to be modeled, the data acquisition may be eased as well.

Once a model of the form of Eq. (25) is identified, if possible, converting the FNN part of the SINDy model back to symbolic functions will greatly improve the model inference time as explicit nonlinearities are computationally desirable. Such advances have already been initiated in recent papers on modeling biological systems (Rackauckas et al., 2020).

8.2. Real-time model updates

In the presence of disturbances or changes in process behavior due to, for example, catalyst deactivation or feed stream disruptions, the process model in a model-based controller such as MPC must be updated in real-time to reflect the changes. Much of the research in model re-identification is concentrated on the mathematical details of the algorithms used for the model update, such as recursive least-squares or recursive singular value decomposition (Moonen et al., 1989; Lovera et al., 2000; Mercere et al., 2004) rather than developing a rigorous framework for the triggering of the model re-identification procedure. Research on the triggering procedure include error-triggered as well as event-triggered model re-identification (Alanqar et al., 2017a,b; Wu et al., 2020), but mostly use first-principles process models. In the context of SINDy, Quade et al. (2018) proposed a model re-identification procedure, where the SINDy model coefficients could be updated or terms could be added or deleted as required. The trigger for re-identification was a significant divergence between the local Lyapunov exponent and the prediction horizon estimate (although the definition of "prediction horizon" in Quade et al. (2018) differs from its usage in this manuscript). However, the results of Quade et al. (2018) were only in the context of modeling. Hence, a future direction for research in sparse identification would be to consider real-time updates to a data-based SINDy model based on the error- or event-triggering mechanism of Wu et al. (2020).

9. Conclusions

In this paper, we have provided an overview of several recent advancements in the sparse identification for nonlinear dynamics (SINDy) method to overcome the challenges of modeling and controlling two-time-scale systems and noisy data. The methods considered included combining SINDy with nonlinear principal component analysis, feedforward neural networks, subsampling, co-teaching, and ensemble learning. The novel methods were described in detail, and best practices, tuning guidelines, as well as common pitfalls to avoid, for their successful application in process systems engineering were provided for control practitioners. To demonstrate their effectiveness, results from applying the proposed algorithms to chemical processes were provided. Subsequently, SINDy was used to model a nonlinear chemical process to provide a demonstration of its application as well as to highlight specific challenges faced when applying SINDy in process systems engineering. Finally, a number of future research directions were outlined.

CRediT authorship contribution statement

Fahim Abdullah: Conceptualization, Methodology, Software, Writing. **Panagiotis D. Christofides:** Supervision, Reviewing and editing.

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.

Data availability

Data will be made available on request.

References

Abdullah, F., Alhajeri, M.S., Christofides, P.D., 2022a. Modeling and control of nonlinear processes using sparse identification: Using dropout to handle noisy data. Ind. Eng. Chem. Res. 61 (49), 17976–17992.

Abdullah, F., Wu, Z., Christofides, P.D., 2021a. 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., 2021b. Sparse-identification-based model predictive control of nonlinear two-time-scale processes. Comput. Chem. Eng. 153, 107411.

Abdullah, F., Wu, Z., Christofides, P.D., 2022b. Handling noisy data in sparse model identification using subsampling and co-teaching. Comput. Chem. Eng. 157, 107628.

Aggelogiannaki, E., Sarimveis, H., 2008. Nonlinear model predictive control for distributed parameter systems using data driven artificial neural network models. Comput. Chem. Eng. 32 (6), 1225–1237.

Alanqar, A., Durand, H., Christofides, P.D., 2017a. Error-triggered on-line model identification for model-based feedback control. AIChE J. 63, 949–966.

Alanqar, A., Durand, H., Christofides, P.D., 2017b. Fault-tolerant economic model predictive control using error-triggered online model identification. Ind. Eng. Chem. Res. 56, 5652–5667.

Alhajeri, M.S., Abdullah, F., Wu, Z., Christofides, P.D., 2022a. Physics-informed machine learning modeling for predictive control using noisy data. Chem. Eng. Res. Des. 186, 34–49.

Alhajeri, M.S., Luo, J., Wu, Z., Albalawi, F., Christofides, P.D., 2022b. Process structure-based recurrent neural network modeling for predictive control: A comparative study. Chem. Eng. Res. Des. 179, 77–89.

Alhajeri, M.S., Wu, Z., Rincon, D., Albalawi, F., Christofides, P.D., 2021. Machine-learning-based state estimation and predictive control of nonlinear processes. Chem. Eng. Res. Des. 167, 268–280.

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 J. 53 (4), 920–933.

Bangi, M.S.F., Kwon, J.S.-I., 2020. Deep hybrid modeling of chemical process: Application to hydraulic fracturing. Comput. Chem. Eng. 134, 106696.

Bhadriraju, B., Bangi, M.S.F., Narasingam, A., Kwon, J.S.-I., 2020. Operable adaptive sparse identification of systems: Application to chemical processes. AIChE J. 66 (11), e16980.

Bhadriraju, B., Narasingam, A., Kwon, J.S.-I., 2019. Machine learning-based adaptive model identification of systems: Application to a chemical process. Chem. Eng. Res. Des. 152, 372–383.

Bikmukhametov, T., Jäschke, J., 2020. Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models. Comput. Chem. Eng. 138, 106834.

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

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.

Chang, H.C., Aluko, M., 1984. Multi-scale analysis of exotic dynamics in surface catalyzed reactions–I: Justification and preliminary model discriminations. Chem. Eng. Sci. 39 (1), 37–50.

de Silva, B., Champion, K., Quade, M., Loiseau, J.C., Kutz, J., Brunton, S., 2020. PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data. J. Open Source Softw. 5 (49), 2104.

Dong, D., McAvoy, T., 1996. Nonlinear principal component analysis–Based on principal curves and neural networks. Comput. Chem. Eng. 20 (1), 65–78.

Efron, B., Stein, C., 1981. The Jackknife estimate of variance. Ann. Statist. 9 (3), 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.

González-García, R., Rico-Martínez, R., Kevrekidis, I., 1998. Identification of distributed parameter systems: A neural net based approach. Comput. Chem. Eng. 22, S965–s968.

Hoffmann, M., Fröhner, C., Noé, F., 2019. Reactive SINDy: Discovering governing reactions from concentration data. J. Chem. Phys. 150 (2), 025101.

Holkar, K.S., Waghmare, L.M., 2010. An overview of model predictive control. Int. J. Control Autom. 3 (4), 47–63.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Netw. 4 (2), 251–257.

Hornik, K., Stinchcombe, M., White, H., 1990. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Netw. 3 (5), 551–560.

Kaheman, K., Kutz, J.N., Brunton, S.L., 2020. SINDy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics. Proc. R. Soc. A: Math., Phys. Eng. Sci. 476 (2242), 20200279.

Kaptanoglu, A.A., de Silva, B.M., Fasel, U., Kaheman, K., Goldschmidt, A.J., Callaham, J., Delahunt, C.B., Nicolaou, Z.G., Champion, K., Loiseau, J.-C., Kutz, J.N., Brunton, S.L., 2022. PySINDy: A comprehensive Python package for robust sparse system identification. J. Open Source Softw. 7 (69), 3994.

Kokotović, P., Khalil, H.K., O'Reilly, J., 1999. Singular Perturbation Methods in Control: Analysis and Design. Society for Industrial and Applied Mathematics, pp. 93–156.

Kramer, M.A., 1991. Nonlinear principal component analysis using autoassociative neural networks. AIChE J. 37 (2), 233–243.

Lee, D., Jayaraman, A., Kwon, J.S., 2020. Development of a hybrid model for a partially known intracellular signaling pathway through correction term estimation and neural network modeling. PLoS Comput. Biol. 16, 1–31.

Likar, B., Kocijan, J., 2007. Predictive control of a gas–liquid separation plant based on a Gaussian process model. Comput. Chem. Eng. 31 (3), 142–152.

Lin, Y., Sontag, E.D., 1991. A universal formula for stabilization with bounded controls. Systems Control Lett. 16, 393–397.

Lovera, M., Gustafsson, T., Verhaegen, M., 2000. Recursive subspace identification of linear and non-linear Wiener state-space models. Automatica 36 (11), 1639–1650.

Mangan, N.M., Brunton, S.L., Proctor, J.L., Kutz, J.N., 2016. Inferring biological networks by sparse identification of nonlinear dynamics. IEEE Trans. Mol., Biol. Multi-Scale Commun. 2 (1), 52–63.

McBride, K., Sundmacher, K., 2019. Overview of surrogate modeling in chemical process engineering. Chem. Ing. Tech. 91 (3), 228–239.

Mercere, G., Lecoeuche, S., Lovera, M., 2004. Recursive subspace identification based on instrumental variable unconstrained quadratic optimization. Internat. J. Adapt. Control Signal Process. 18, 771–797.

Moonen, M., De Moor, B., Vandenberghe, L., Vandewalle, J., 1989. On-and off-line identification of linear state-space models. Internat. J. Control 49, 219–232.

Moore, C., 1986. Application of singular value decomposition to the design, analysis, and control of industrial processes. In: 1986 American Control Conference. Seattle, WA, USA, pp. 643–650.

Narasingam, A., Kwon, J.S.I., 2018. Data-driven identification of interpretable reduced-order models using sparse regression. Comput. Chem. Eng. 119, 101–111.

Oliveira, R., 2003. Combining first principles modelling and artificial neural networks: A general framework. In: Kraslawski, A., Turunen, I. (Eds.), European Symposium on Computer Aided Process Engineering-13. In: Computer Aided Chemical Engineering, vol. 14, Elsevier, pp. 821–826.

Porru, G., Aragonese, C., Baratti, R., Servida, A., 2000. Monitoring of a CO oxidation reactor through a grey model-based EKF observer. Chem. Eng. Sci. 55 (2), 331–338.

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

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., Edelman, A., 2020. Universal differential equations for scientific machine learning. arXiv preprint arXiv:2001.04385.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2018. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. arXiv:1801.01236.

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

Sansana, J., Joswiak, M.N., Castillo, I., Wang, Z., Rendall, R., Chiang, L.H., Reis, M.S., 2021. Recent trends on hybrid modeling for industry 4.0. Comput. Chem. Eng. 151, 107365.

Schulze, J.C., Doncevic, D.T., Mitsos, A., 2022. Identification of MIMO Wiener-type Koopman models for data-driven model reduction using deep learning. Comput. Chem. Eng. 161, 107781.

Tsay, C., Baldea, M., 2020. Integrating production scheduling and process control using latent variable dynamic models. Control Eng. Pract. 94, 104201.

Van Overschee, P., De Moor, B., 1994. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica 30 (1), 75–93.

von Stosch, M., Oliveira, R., Peres, J., Feyo de Azevedo, S., 2014. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. Comput. Chem. Eng. 60, 86–101.

Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106, 25–57.

Wilson, Z.T., Sahinidis, N.V., 2017. The ALAMO approach to machine learning. Comput. Chem. Eng. 106, 785–795.

Wu, Z., Luo, J., Rincon, D., Christofides, P.D., 2021a. 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. Ind. Eng. Chem. Res. 59 (6), 2275–2290.

Wu, Z., Rincon, D., Luo, J., Christofides, P.D., 2021b. Machine learning modeling and predictive control of nonlinear processes using noisy data. AIChE J. 67 (4), e17164.

Wu, Z., Tran, A., Rincon, D., Christofides, P.D., 2019a. Machine learning-based predictive control of nonlinear processes. Part I: Theory. AIChE J. 65, e16729.

Wu, Z., Tran, A., Rincon, D., Christofides, P.D., 2019b. Machine learning-based predictive control of nonlinear processes. Part II: Computational implementation. AIChE J. 65, e16734.

Zendehboudi, S., Rezaei, N., Lohi, A., 2018. Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review. Appl. Energy 228, 2539–2566.

Zhang, S., Lin, G., 2018. Robust data-driven discovery of governing physical laws with error bars. Proc. R. Soc. A: Math., Phys. Eng. Sci. 474 (2217), 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.

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.