Contents lists available at ScienceDirect

Chemical Engineering Research and Design

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

Data-based reduced-order modeling of nonlinear two-time-scale processes

Fahim Abdullah[a], *Zhe Wu*[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

Article history:
Received 5 October 2020
Received in revised form 6 November 2020
Accepted 10 November 2020

Keywords:
Two-time-scale processes
Nonlinear processes
Singular perturbations
Nonlinear principal component analysis
Sparse identification
Chemical processes

ABSTRACT

This work focuses on data-based reduced-order modeling of nonlinear processes that exhibit time-scale multiplicity. Using time-series data from all the state variables of a nonlinear process, an approach that involves nonlinear principal component analysis and neural network function approximators is employed to identify the fast and slow process state variables as well as compute a nonlinear slow manifold function approximation in which the fast variables are "slaved" in terms of the slow variables. Subsequently, a nonlinear sparse identification approach is employed to calculate a dynamic model of nonlinear first-order ordinary differential equations that describe the temporal evolution of the slow process states. The method is applied to two chemical process examples, and the advantages of the proposed method over using only sparse identification for the original two-time-scale process are discussed. The approach can be thought of as a data-based analogue of the classical singular perturbation modeling of two-time-scale processes where explicit time-scale separation is assumed (and expressed in terms of a small positive parameter ε multiplying the time derivative of the fast states), and the reduced-order slow subsystem is calculated analytically.

© 2020 Institution of Chemical Engineers. Published by Elsevier B.V. All rights reserved.

1. Introduction

Advanced control methods such as model-predictive control (MPC) require a process model for predicting the states and/or outputs for the control and optimization of chemical processes. However, most chemical processes exhibit nonlinear behavior and, often, also multiple-time-scale dynamics. A number of such processes that have been studied include biochemical reactors (Christofides and Daoutidis, 1996; Heineken et al., 1967; Merrill, 1978), catalytic continuous stirred-tank reactors (CSTRs) (Chang and Aluko, 1984), fluidized catalytic crackers (Chang and Aluko, 1984; Monge and Georgakis, 1987; Georgakis, 1977), and distillation columns (Lévine and Rouchon, 1991). If the time-scale separation in such systems is neglected, directly implementing standard nonlinear feedback control methods may lead to controller ill-conditioning or loss of closed-loop stability (Kokotović et al., 1999).

As nonlinear process models are in the form of ordinary differential equations (ODEs) in time, they need to be integrated to predict the states and/or outputs. However the presence of fast dynamics in such processes result in stiff ODEs. Such stiff ODEs require a very small integration step size when using explicit integration methods to avoid numerical instability and to produce accurate solutions. An alternative is to compute the temporal evolution of the fast states using a different method than numerical integration of a stiff ODE.

Specifically, utilizing the mathematical framework of singular perturbations (Kokotović et al., 1999), the stiff ODE system is written in the standard singularly perturbed form, where a small positive parameter, ε, is used to multiply the time-derivative of the fast states. Using singular perturbations, the original system is decomposed into two lower-order subsystems by utilizing the inherent two-time-scale property of the system. Each lower-order subsystem separately rep-

* *Corresponding author*: Tel.: +1 (310) 794-1015; fax: +1 (310) 206-4107.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.cherd.2020.11.009

0263-8762/© 2020 Institution of Chemical Engineers. Published by Elsevier B.V. All rights reserved.

resents the slow and fast dynamics of the original stiff ODE model and may be simpler to study individually. The asymptotic behavior (for small ε) of the original system can then be inferred based on the behavior of the lower-order subsystems. In particular, once the fast states converge to the slow manifold after a short transient period, there exists a nonlinear relationship between the slow and fast states, which can be taken advantage of.

To capture nonlinear relationships in data, Ref. Kramer (1991) suggested a nonlinear generalization of principal component analysis, termed nonlinear principal component analysis (NLPCA). This was developed further in Ref. Dong and McAvoy (1996), where it was proved that this new method could accurately capture significant nonlinear relationships between the variables in a time-series data set. Furthermore, for the reconstruction of ODEs from data, a procedure known as sparse identification was devised in Ref. Brunton et al. (2016).

In a two-time-scale system, if the nonlinear relationships between the slow and fast states can be derived using NLPCA, then the fast subsystem may be predicted using a deterministic neural network (NN) model obtained from NLPCA with the slow subsystem as its input, with the results being valid after a short transient period. The evolution of the system on the slow manifold, on the other hand, can be predicted by integrating the sparse identified model with a larger integration step size compared to the one needed to integrate the original system of stiff ODEs. The objective of this work is to investigate an approach of combining NLPCA with sparse identification to efficiently predict the full state of the original stiff ODE system with minimal loss of accuracy except for a short transient period. Neural networks have already been used effectively in another two-time-scale process, chemical vapor deposition, for parameter estimation under uncertainty (Kimaev and Ricardez-Sandoval, 2020a), dynamic optimization (Kimaev and Ricardez-Sandoval, 2020b), and nonlinear control using MPC (Kimaev and Ricardez-Sandoval, 2019).

2. Preliminaries

2.1. Class of systems

The general class of continuous-time nonlinear systems with m states considered in this work has the following general form:

$$\dot{\bar{x}} = f(\bar{x}) \tag{1}$$

where $\bar{x} \in \mathbb{R}^m$ is the state vector, and the vector function $f(\bar{x})$ contains the dynamic modeling constraints that are inherently present in the system due to its physics. We assume that the system of Eq. (1) exhibits time-scale separation in the sense that it involves coupled slow and fast states where $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$ are the slow and fast state vectors, respectively, with $n + p = m$.

Traditionally, the system of Eq. (1) has been studied within the framework of singular perturbations where a small positive parameter ε representing the speed ratio of the slow to the fast dynamics of the system is used to write the system of Eq. (1) in the following standard singularly perturbed form:

$$\begin{aligned} \dot{x} &= f_1(x, z) \\ \varepsilon \dot{z} &= f_2(x, z) \end{aligned} \tag{2}$$

where $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$ are the slow and fast state vectors, respectively, with $n + p = m$. The vector functions $f_1(x, z)$ and $f_2(x, z)$ are sufficiently smooth vector functions in $\mathbb{R}^n$ and $\mathbb{R}^p$, respectively. In the system of Eq. (2), after a short transient period, the fast states, z, converge to a slow manifold (provided such an isolated manifold exists) and can be expressed by a nonlinear algebraic expression in x, the slow states. In this work, nonlinear principal component analysis (NLPCA) is utilized to capture this nonlinear manifold relationship between the slow and fast states, while sparse identification is used to reconstruct the slow dynamic model for the slow states.

Remark 1. Traditionally, two-time-scale systems of the form of Eq. (2) have been analyzed by setting ε to zero in Eq. (2) and calculating the slow manifold and the slow subsystem analytically. This requires knowing the vector functions $f_1(x, z)$ and $f_2(x, z)$. The objective of this work is to calculate the slow manifold and the slow subsystem using only time series data of $\bar{x}(t)$.

2.2. Nonlinear principal component analysis

Principal component analysis (PCA) is a well-established dimensionality reduction technique for linearly mapping higher-dimensional data onto a lower-dimensional space with marginal loss of information by minimizing the sum of orthogonal deviations from a line. It is applicable to chemical processes as they contain voluminous data but much less information. However, as chemical processes are usually nonlinear, this can introduce inaccuracies (Paluš and Dvorák, 1992) such as the minor components containing more than just noise or insignificant variance (Xu et al., 1992). Therefore, for nonlinear dynamical systems such as chemical processes, NLPCA was developed.

2.2.1. Principal curve

In NLPCA as defined in Ref. Dong and McAvoy (1996), a "principal curve" (Hastie and Stuetzle, 1989) passing through the "middle" of the data set is first fitted to the data. This curve captures the nonlinear relationship between the variables by minimizing the sum of orthogonal deviations between the data and the curve. If not, analogously to PCA, the procedure can be carried out successively on the residuals to compute more components as required to capture the desired variance.

The principal curve is a 1-D curve in the m-D space of the states. It is represented by a vector function, $f(\mu)$, consisting of m functions of only the ordered arc-length along the curve, μ. If $\bar{x} \in \mathbb{R}^m$ is a continuous random vector with probability distribution, h, and $\|\cdot\|$ denotes the Euclidean norm of a vector, then the projection index, $\mu_f : \mathbb{R}^m \to \mathbb{R}$, can be defined as

$$\mu_f(\bar{x}) = \sup_{\mu \in \mathbb{R}} \{\mu : \|\bar{x} - f(\mu)\| = \inf_{\nu \in \mathbb{R}} \|\bar{x} - f(\nu)\|\} \tag{3}$$

where ν is a substitute for μ, and also represents the arc length. Hence, μ_f is the value of μ for which the curve, $f(\mu)$, is nearest to $\bar{x}$. In the case of multiple such values, the largest is used. A principal curve with distribution h can now be defined as the curve, f, such that

$$\mathbb{E}(\bar{x} | \mu_f(\bar{x}) = \mu) = f(\mu) \tag{4}$$

where $\mathbb{E}$ is the expectation operator.

The procedure to estimate the principal curve represented by Eq. (4) is described next. The challenge is that the expectation operator $\mathbb{E}$ in Eq. (4) can be computed only if the distribution of $\bar{x}$ is known. However, in most engineering applications, including the systems of interest in this work, $\bar{x}$ is a discrete, multivariate data set with unknown distribution sampled from a process. Hence, as suggested in Ref. Hastie and Stuetzle (1989), we estimate the principal curve, given by the left-hand side of Eq. (4), $\mathbb{E}(\bar{x}|\mu_f(\bar{x}) = \mu)$, using scatter-plot smoothing with locally weighted regression. This is implemented through a two-step iteration procedure. To initialize the iterations, the first linear PCA line, f^0, is used as the initial guess. Then, in each iteration, first, the data points are orthogonally projected onto the current estimate of the curve, f^i, and their ordered arc lengths are calculated from the "first" projected point on the curve. Since the points are joined by line segments in this work, this is equivalent to a cumulative sum of successive Euclidean distances. The "first" data point is taken as the data point with the lowest value of $\bar{x}_1$ (first dimension of the data set) after projection onto f^i. In the second step, the new curve, $f^{i+1} = \mathbb{E}(\bar{x}|\mu_{f^i}(\bar{x}) = \mu)$, is estimated by using locally weighted regression (Cleveland, 1979) on the data set where the neighborhood of each point is based on the ordering from the first step. The number of points used for smoothing is based on the "span" parameter, $0 < s \leq 1$, representing the percentage of points to be used. Specifically, for r data points, sr rounded to the nearest integer would be the number of points used for smoothing. The weights are assigned to the neighboring points using the tri-cube weight function,

$$w_{ij}(\mu) = \begin{cases} \left(1 - \left|\frac{\mu_j - \mu_i}{d_i}\right|^3\right)^3 & \text{if } |\mu_j - \mu_i| < d_i \\ 0 & \text{if } |\mu_j - \mu_i| \geq d_i \end{cases} \tag{5}$$

where w_{ij} is the weight assigned to point j when computing the new point i with $i, j = 1, \ldots, r$, and d_i is the distance from point i to the furthest neighboring point considered. For further details regarding locally weighted regression, the reader is referred to Ref. Cleveland (1979). This second step yields the new curve, f^{i+1}. The two steps described are then repeated with the new curve in the next iteration until a stopping criterion is met. The criterion used in this work is the relative change in the Euclidean distance from a point $\bar{x}$ to its projection on the principal curve, f. Specifically, if ε is a small, positive number, we iterate until

$$\left|\frac{\|\bar{x} - f^i(\mu_f)\| - \|\bar{x} - f^{i+1}(\mu_f)\|}{\|\bar{x} - f^i(\mu_f)\|}\right| < \varepsilon$$

The principal curve, however, is not a model and cannot generate an output from a new input sample. Therefore, the principal curve is approximated using a feedforward neural network, which is then used as the model to query new data.

2.2.2. Neural network

A two-layer feedforward neural network model that approximates a nonlinear function of the form, $y = f(x)$, can be written in the general form,

$$y = \sigma^{(3)}(W^{(3)}h^{(2)} + b^{(3)}) \tag{6}$$

$$h^{(2)} = \sigma^{(2)}(W^{(2)}h^{(1)} + b^{(2)}) \tag{7}$$

$$h^{(1)} = \sigma^{(1)}(W^{(1)}x + b^{(1)}) \tag{8}$$

where $x \in \mathbb{R}^n$ is the NN input vector, $y \in \mathbb{R}^p$ is the NN output vector. Each layer has a bias vector $b^{(i)}$ with $i = 1, 2, 3$, and an activation function $\sigma^{(i)}$ with $i = 1, 2, 3$, which is a nonlinear activation function such as the sigmoid function, $\sigma(x) = 1/(1 + e^{-x})$. The hidden layers have output vectors $h^{(i)}$ with $i = 1, 2$. Every pair of units in two consecutive layers has an associated weight, which is stored in the weight matrix $W^{(i)}$.

2.3. Sparse identification

Sparse identification is a technique developed to identify dynamical systems made possible due to recent advances in sparsity algorithms. Sparsity methods have been applied to dynamic system modeling in recent works (Wang et al., 2011; Schaeffer et al., 2013; Ozolinš et al., 2013; Mackey et al., 2014; Brunton et al., 2014; Proctor et al., 2014; Bai et al., 2015; Arnaldo et al., 2015). Sparse identification attempts to reconstruct the original ODE using only measured data from the system and identifies dynamical systems expressed in the form given in Eq. (1).

Sparse identification takes advantage of the fact that $f(\bar{x})$ in Eq. (1) typically contains very few nonzero terms, which makes it sparse in a higher-dimensional space of many candidate nonlinear functions. This sparsity allows calculation of the nonzero terms using scalable convex methods while avoiding a prohibitively expensive brute-force search.

In sparse identification, time-series data of the state x is first collected by sampling the process with sampling period, Δ, over a range of initial and operating conditions. The concatenated data matrix, X, is then, of the form,

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \tag{9}$$

where each x_i is a column of time-series data for state i for $i = 1, \ldots, m$. The matrix of the derivative of X,

$$\dot{X} = \begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \cdots & \dot{x}_m \end{bmatrix} \tag{10}$$

is estimated in this work using second-order central finite differences (except the first and last points, which use second-order forward and backward finite differences, respectively). Next, a library, $\Theta(X)$, of q nonlinear functions of the columns of X is created. These functions are candidate terms for f, the right-hand side of Eq. (1). The objective is to find which of these terms are active by leveraging sparsity. An example of an augmented library or $\Theta(X)$ is

$$\Theta(X) = \begin{bmatrix} | & | & | & | & | & | \\ \mathbf{1} & X & X^{P_2} & \cdots & \sin X & \tanh X \\ | & | & | & | & | & | \end{bmatrix} \tag{11}$$

where, for example, X^{P_2} denotes all quadratic nonlinearities, given by

$$X^{P_2} = \begin{bmatrix} x_1^2 & x_1x_2 & \cdots & x_2^2 & x_2x_3 & \cdots & x_n^2 \end{bmatrix} \tag{12}$$

The sparse identification problem is to determine the q coefficients associated with the q candidate nonlinear functions considered in $\Theta(X)$ for each of the m states. The m

coefficient vectors, each denoted by ξ, can be compactly written as a matrix,

$$\Xi = \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_m \end{bmatrix} \tag{13}$$

Each $\xi_i \in \mathbb{R}^q$ is a sparse vector of coefficients indicating the active terms in the dynamical equation of the corresponding row, $\dot{x}_i = f_i(x)$. The equation that must now be solved to determine Ξ can be set up as

$$\dot{X} = \Theta(X)\Xi \tag{14}$$

The above equation is solved using least-squares after zeroing all coefficients in Ξ that are smaller than a threshold, λ, known as the sparsification knob since this single parameter controls the sparsity of Ξ. This is repeated until the non-zero coefficients converge, which is very rapid in practice.

As expected, the effectiveness of sparse identification is intrinsically related to the candidate nonlinear functions used to compute the sparse dynamics. However, as discussed in Ref. Brunton et al. (2016), polynomials and trigonometric functions are a natural basis for many systems and may be used as an initial bank of basis functions. More nonlinear functions may be added to the bank based on physical knowledge of the specific system such as exponential terms for systems with chemical reactions.

3. NLPCA-SI implementation

In this work, the two methods described—NLPCA and sparse identification—are used together to identify the original two-time-scale system dynamics. The objective is to use NLPCA for deriving nonlinear relationships between the slow and fast dynamics and use sparse identification to reconstruct the slow dynamics. Thus, the final system is in the form of ODEs for the slow states and a neural network for the fast states. The method is abbreviated as NLPCA-SI.

The NLPCA-SI procedure is as follows: (1) a large data set is generated through open-loop simulations of the original nonlinear system in Eq. (1) using different initial conditions in a reasonable operating region, (2) a time-series plot of the states is used to separate the slow and fast states, (3) the collected data is then auto-scaled by subtracting the mean and dividing by the standard deviation since NNs have activation functions with relatively small ranges, (often 0 to 1), making them inaccurate or difficult to train if the data varies too widely, (4) the principal curve is computed using the entire m-dimensional data set, (5) and a feedforward NN as described in Section 2.2.2 is built using only the slow states as the inputs and the fast states as the outputs. Two remarks are made regarding the procedure. In the second step, another, possibly more robust, method to identify the slow and fast states *a priori* is to plot the time-series gradients of the data. For the fast states, due to the stiffness, the magnitudes of the gradients are extremely large initially. For every simulation of the ODE systems for data generation, the initial gradients of the fast states were at least one order of magnitude greater than that of the slow states. In most runs, the differences were two orders of magnitudes. Therefore, the ratio of initial gradients of the states can be used as a criterion to identify the slow and fast states *a priori*. This idea can be further developed as well. For example, norms of the ratios at multiple time instances can be used, where several orders of magnitudes separate the slow and fast states. In the final step of building the NN, an open-source platform for machine learning, Tensorflow, is used to solve the required optimization problems and obtain the optimal weight matrices, $W^{(i)}$, which minimize the loss function used: the mean-squared error between the predicted and actual outputs.

The NN used in the NLPCA is a two-hidden-layer network with a linear output layer. Ref. Dong and McAvoy (1996) used an overall five-layer network with the principal scores (defined as the arc length, μ, along the principal curve) as a middle layer, based on the initial structure proposed by Ref. Kramer (1991). However, it treated the network as two three-layer networks, with the third layer of the first network being the first layer of the second network. This was done because five-layer networks were considered difficult to train. However, with the recent advances in data science and machine learning, this is not intractable anymore. Hence, a four or even five-layer network may be trained without splitting it into two networks. However, the number of hidden layers is not arbitrary. Ref. Hornik et al. (1990), Hornik (1991) proved that at least one sigmoidal hidden layer is required for a feed-forward neural network to have the universal approximation property. However, this is in theory and the parameters may be extremely specific to achieve the required accuracy. In addition, for nonlinear control systems, two layers may be required for closed-loop stabilization (Sontag, 1992). Finally, the function to be approximated by the NN may have regions where it behaves differently from its behavior in most of its domain. This can lead to the failure of a one-hidden-layer network and might even require a separate NN for distinct regions. Due to these considerations, two hidden layers are used in this NN with the number of hidden neurons in each hidden layer being determined by a grid search and cross-validation as in Ref. Dong and McAvoy (1996). However, it is noted that these two parameters—the number of hidden layers and the number of neurons in each hidden layer—which define the structure of the NN should be based on the complexity of the nonlinear relationship between the inputs and outputs of the NN. Typically, as seen in the examples considered in this work, a shallow neural network (*i.e.*, an NN with only one or two hidden layers) is sufficient for most chemical process modeling problems. However, a deep neural network can be used for a large-scale, complex system with a large number of process states when a shallow neural network cannot achieve the desired model accuracy. In practical implementation, to obtain a well-conditioned neural network model with sufficiently high model accuracy on both training and testing data sets, we will start with a one-hidden-layer neural network, and gradually increase the number of layers and of neurons until no further improvement is noticed. Following these steps, a well-conditioned NN model is obtained to predict the fast states from the slow states. Once the optimal structure of the NN is determined, the NN is trained 10 times and the most accurate model is chosen based on the test set accuracy. The structure of the NN used in NLPCA-SI is shown in Fig. 1.

For the activation functions of the two hidden layers, the first hidden layer uses a sigmoidal activation since, as mentioned, this is required for the universal approximation theorem to hold. For the second activation function, several common activation functions are tested, and the model with the minimal test set error is selected. The most important metric used in the NN training, however, was the learning rate, which typically requires significant tuning. In this method, the learning rate can significantly affect the results for some systems and is tuned until satisfactory performance is observed.

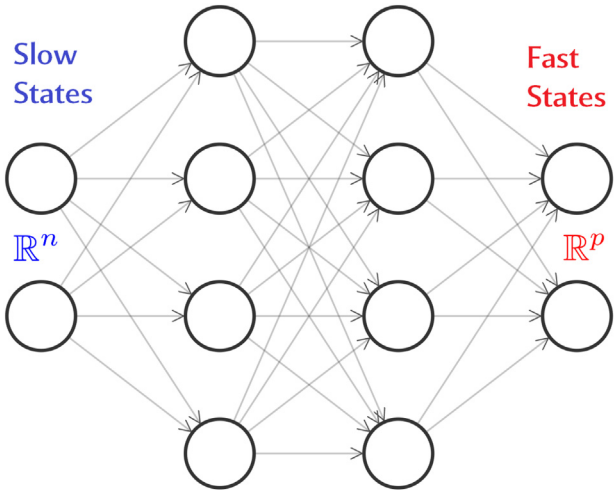

Fig. 1 – Structure of the neural network used for NLPCA-SI.

Finally, sparse identification is used on the data containing only the slow states. This identifies the slow dynamics in the form of ODEs containing only the slow states in the vector function f in Eq. (1). For predicting the states of the original system, new initial conditions are first used to integrate the sparse identified ODEs in time to yield the slow manifold. It is then used as input for the NN built during NLPCA to predict the fast states, neglecting a short transient period. In this way, the full state of the original system is recovered for any initial condition without integrating the fast dynamics.

The alternative is to simply reconstruct the entire system using only sparse identification and integrate it to predict the full state of the system at any time. The results from the NLPCA-SI method are compared with the results from this aforementioned brute-force approach using only sparse identification, abbreviated SI.

Remark 2. It is important to note that, as opposed to classical singular perturbation modeling of two-time-scale processes where the dimension of the slow state vector and of the fast state vector are fixed once the singular perturbation parameter is specified and the process model is written in the standard singularly perturbed form, the proposed approach uses process data to determine the number of fast and slow states and, therefore, leads to a more accurate representation of the dimensions of the slow and fast states. Furthermore, it directly accounts for "hidden" time-scale multiplicity in the right hand-side of the process dynamic model owing to the presence of large or small parameters there and yields a slow subsystem of the lowest order as a result of the application of NLPCA to the overall process state data set. Therefore, this approach can be viewed as a way of carrying out index reduction in high-order (higher than 1) differential-algebraic equation systems (which arise as slow subsystems of two-time-scale systems) to yield a slow subsystem of the lowest order.

4. Application of NLPCA-SI

The method explained in Section 3 is applied to two reactor systems. In the first example, a two-time-scale system due to multiple reactions with different rate constants is studied. Next, a CSTR with a single reaction but fast temperature dynamics is considered. In this section, a "proper" solution refers to a sparse identified system that can be integrated until the end of the time span without early termination due to divergence to infinity during numerical integration.

4.1. *Example 1: Isothermal batch reactor with multiple reactions*

An isothermal, constant-volume batch reactor with the following reaction scheme is considered:

$$A \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} B \xrightarrow{k_2} C \tag{15}$$

where k_i is the rate constant associated with the corresponding reaction. The forward reaction in $A \rightleftharpoons B$ is much faster than both the backward reaction and the $B \longrightarrow C$ reaction such that $k_1 \gg k_{-1} > k_2$. This causes a two-time-scale separation in this system since species A is consumed much faster than the rate of consumption of species B or production of species C.

If the reactor has volume, V, and the concentration of species i is denoted by C_i, the material balances that govern the dynamical behavior of the batch reactor take the following form:

$$\dot{C}_A = -k_1 C_A + k_{-1} C_B \tag{16}$$

$$\dot{C}_B = k_1 C_A - k_{-1} C_B - k_2 C_B \tag{17}$$

$$\dot{C}_C = k_2 C_B \tag{18}$$

$$C_A(0) + C_B(0) = \text{constant} \tag{19}$$

$$C_C(0) = 0 \tag{20}$$

The initial concentration of species C is assumed to be zero, and Eq. (19) ensures that the final, steady-state concentration of species C is constant for each initial condition chosen. Therefore, only one species' initial concentration can be arbitrarily chosen. Note that an explicit ε is not necessary in the left-hand side of the fast ODE as seen in this example. Therefore, for this system, standard singularly perturbed methods for analytically separating the slow and fast states would not be adequate.

The rate constants in this system are chosen to be $k_1 = 5.0\,h^{-1}$, $k_{-1} = 1.0h^{-1}$, $k_2 = 0.5\,h^{-1}$. The constant in Eq. (19), which is the sum of initial concentrations of species A and B, is 9.0 mol m^{-3}. Ten initial conditions are chosen with $C_A(0)$ increasing from 0 mol m^{-3} to 9.0 mol m^{-3} in increments of 1.0 mol m^{-3}, while $C_B(0)$ decreases to satisfy Eq. (19). Using each of these initial conditions, the ODE system given by Eq. (16–18) is numerically integrated from 0.0 hr to 10.0 hr with a step size of 10^{-6} hr. The resulting data is sampled every 0.1 hr and concatenated from the 10 runs into the X data matrix, which is used for NLPCA.

In NLPCA, a span of 0.25 is used, meaning 25% of the data points are used in the smoothing step of each iteration when computing the principal curve. This value yielded the optimal principal curve. The NN built to approximate this curve used a sigmoidal activation function for the first hidden layer and the rectified linear unit (ReLU) for the second as this was sufficient to accurately capture the curve. For both hidden layers, by testing 1 to 15 neurons using the simplified cross-validation scheme, 13 neurons were found to be optimal. With these values of the hyper-parameters, the learning rate did not significantly affect the results and 0.01 was used. The neural

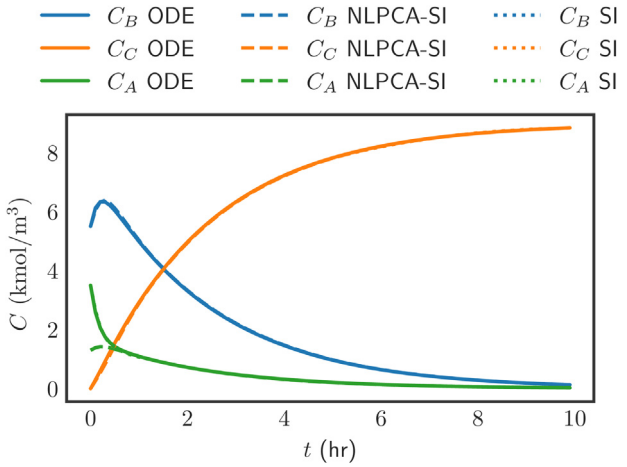

Fig. 2 – Comparison of original data (solid lines) with results from NLPCA-SI (dashed lines) and only sparse identification (dotted lines) for Example 1. Slow states: C_B (blue lines), and C_C (orange lines). Fast state: C_A (green lines)(For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

network hyperparameters used for NLPCA in this example are reported in Table 2.

For sparse identification, the augmented library of functions consisted of the constant/bias term (column of 1), polynomial terms up to fourth order, sin(X), cos(X), tan(X), tanh(X), exp(X), and exp(−X). The exponential terms are added since the reaction terms in reactor mass and energy balances often contain such terms. The optimal λ for the sparse identification algorithm was found to be 0.5 for the NLPCA-SI and 0.13 for SI.

Fig. 2 shows the results for this system. The reconstructed equations from the NLPCA-SI and SI are not included due to their length. Instead, for brevity, only the reconstructed states are plotted and compared. Both NLPCA-SI and SI reproduce the dynamics very accurately. As seen earlier, the NLPCA-SI does not model the transient regime well.

The differences between NLPCA-SI and only SI becomes apparent when the value of the sparsification knob, λ, in sparse identification and the sampling period in the data generation step, Δ, are investigated.

The sparsification knob, λ, is tuned through extensive simulations to achieve the desired balance between sparsity and approximation accuracy for nonlinear dynamic systems—a larger λ leads to more sparse but less accurate dynamics. However, the range of values of λ that yielded the optimal results shown in Fig. 2 were not similar in the two cases. For the NLPCA-SI, any value of λ below the optimal 0.5 produced identical trajectories for the slow states, C_B and C_C. This was tested until $\lambda = 10^{-6}$. However, for SI, the value of λ that produced the ideal curves for all three states was only 0.05. Several values did not yield a proper solution, while most of them resulted in significantly inferior performance than shown in Fig. 2. It could not be determined that the optimal solution has been found without extensive testing of a large number of values for λ, as there was no range of values that consistently produced optimal or near-optimal solutions.

For the sampling period, Δ, as expected, reducing the sampling period generally yields significantly more accurate sparse identified models due to the greater number of data points and more accurate gradient calculations from more closely spaced data points. However, it might not be practically feasible to sample data at such short intervals due to, for example, limitations of the measurement device, such as for concentration. Furthermore, this can cause chattering behavior and numerical instabilities in the gradient estimation for the fast states, especially if noise is present. If the sampling period is not small enough, and the fast dynamics are extremely fast (ε in Eq. (2) is extremely small), it is possible that the fast transient takes place in a time shorter than the sampling period. In this case, the sparse identification algorithm cannot, in general, capture the fast dynamics and may predict a wrong steady-state value entirely. Therefore, the sampling period, Δ, is also carefully chosen based on extensive simulations in this work. Conversely, since NLPCA-SI only uses the slow manifold to predict the fast states, it is affected by neither the sampling period nor the ε. However, the key advantage of NLPCA-SI over SI, as seen clearly in this example, is that NLPCA-SI predicts the fast state at least as accurately as SI (after the short transient) without requiring any integration of stiff ODEs with extremely small time steps.

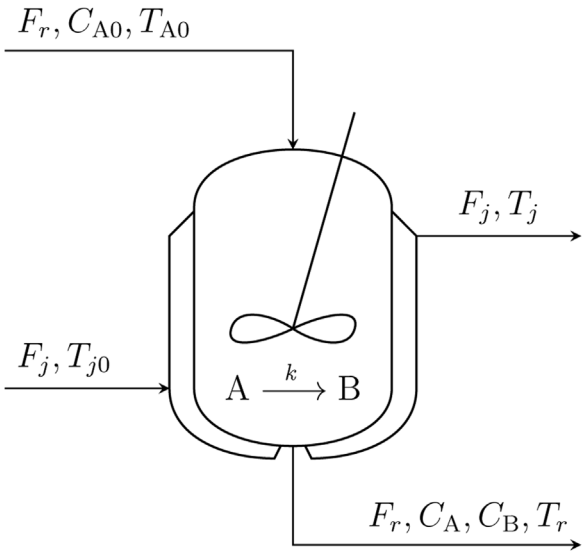

Fig. 3 – A continuous-stirred tank reactor with jacket.

Table 1 – Parameter values for example 2

$V_r = 1.0\,m^3$	$k_0 = 3.36 \times 10^6\,h^{-1}$
$V_j = 0.08\,m^3$	$E = 8.0 \times 10^3\,kcalkg^{-1}$
$A_r = 6.0\,m^3$	$T_{A0} = 310.0\,K$
$U = 1000.0\,kcalh^{-1}m^{-2}K^{-1}$	$T_{j0} = 357.5\,K$
$R = 1.987\,kcalkmol^{-1}K^{-1}$	$\rho_m = 900.0\,kgm^{-3}$
$C_{A0} = 3.75\,kmolm^{-3}$	$\rho_j = 800.0\,kgm^{-3}$
$c_{p,m} = 0.231\,kcalkg^{-1}.K^{-1}$	$F_r = 3.0\,m^3h^{-1}$
$c_{p,j} = 0.200\,kcalkg^{-1}.K^{-1}$	$F_j = 20.0\,m^3h^{-1}$
$\Delta H_r = 5.4 \times 10^4\,kcalmol^{-1}$	

4.2. *Example 2: Non-isothermal CSTR with jacket*

A single, endothermic, irreversible reaction of the form,

$$A \xrightarrow{k} B \tag{21}$$

takes place in a perfectly mixed non-isothermal CSTR as shown in Fig. 3.

In this process, C_A is the concentration of reactant A in the reactor, V_r is the volume of the reacting liquid in the reactor (assuming the vessel has constant holdup), and T_r is the tem-

perature of the reactor. The inlet stream contains pure species A at a volumetric flow rate F, concentration C_{A0}, and temperature T_{A0}. A heating jacket of volume, V_j, heats the reactor. The heat transfer fluid is added to the jacket at a volumetric flow rate F_j and an inlet temperature T_{j0}. The reacting liquid has a constant density of ρ_m and a heat capacity of $c_{p,m}$, while the heat transfer fluid has a constant density of ρ_j and a heat capacity of $c_{p,j}$. The enthalpy of the reaction is ΔH_r. The heat transfer coefficient is U, and the area of heat transfer between the jacket and the reactor is A_r. The rate of the reaction in Eq. (21) is assumed to be of the form,

$$r = -k_0 \exp\left(\frac{-E}{RT_r}\right) C_A \tag{22}$$

where k_0, R, and E represent the pre-exponential constant, ideal gas constant, and activation energy, respectively. Under these assumptions, a material balance for species A, a reactor energy balance, and a jacket energy balance can be formulated, respectively, as:

$$V_r \dot{C}_A = F_r (C_{A0} - C_A) - k_0 e^{-E/RT_r} C_A V_r \tag{23}$$

$$V_r \dot{T}_r = F_r (T_{A0} - T_r) + \frac{(-\Delta H_r)}{\rho_m c_{p,m}} k_0 e^{-E/RT_r} C_A V_r + \frac{UA_r}{\rho_m c_{p,m}} (T_j - T_r) \tag{24}$$

$$V_j \dot{T}_j = F_j T_{j0} - F_j T_j - \frac{UA_r}{\rho_j c_{p,j}} (T_j - T_r) \tag{25}$$

If ε is defined as the ratio of the jacket volume to the reactor volume ($\varepsilon = V_j/V_r$), Eq. (25) can be rewritten explicitly in ε and V_r only without a V_j term as follows:

$$\varepsilon \dot{T}_j = \frac{1}{V_r}\left(F_j T_{j0} - F_j T_j - \frac{UA_r}{\rho_j c_{p,j}} (T_j - T_r)\right)$$

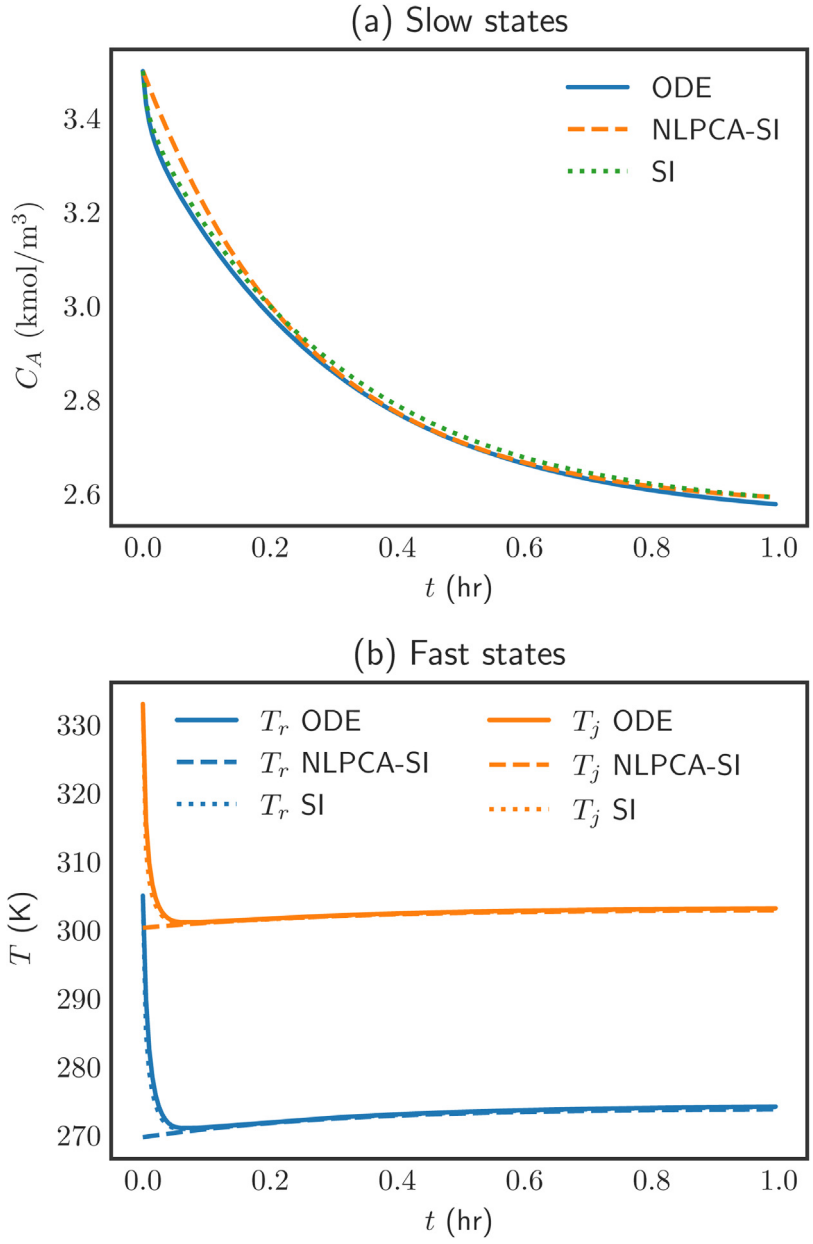

Fig. 4 – Comparison of original data (solid lines) with results from NLPCA-SI (dashed lines) and only sparse identification (dotted lines) for Example 2.

Table 2 – Neural network hyperparameters used for NLPCA in example systems of Section 4

Neural network in example	Learning rate	First hidden layer		Second hidden layer	
		Neurons	Activation	Neurons	Activation
1	0.01	13	Sigmoid	13	ReLU
2	0.01	5	Sigmoid	13	tanh

However, the right-hand sides of the equations are also relevant in determining the slow-fast dynamics of a process. Hence, writing the system in terms of ε is not crucial in our approach and can also be misleading. In this system both temperatures have fast dynamics in comparison to the concentration, which has much slower dynamics. In standard singularly perturbed methods, only T_j would be considered the fast state, while, in fact, T_r is also a fast state in this system.

The parameter values used for this system are given in Table 1. Ten initial conditions are chosen with $C_A(0)$ increasing from 0 mol m^{-3} to 9.0 mol m^{-3} in increments of 1.0 mol m^{-3}, $T_r(0)$ increasing from 280 K to 370 K in increments of 10 K, and $T_j(0)$ increasing from 300 K to 390 K in increments of 10 K. Using each initial condition, the ODE system given by Eq. (23–25) is numerically integrated from 0.0 hr to 1.0 hr with a step size of 10^{-6} hr. The resulting data is sampled every 0.005 hr and concatenated from the 10 runs into the data matrix, X. A time-series plot of X and/or gradient of X shows that only C_A is the slow state in this system, while both temperatures are fast. The NLPCA is carried out accordingly with C_A being the input to the NN and T_r and T_j being the outputs of the NN.

The optimal span for NLPCA is 0.2 *i.e.* 20% of the data points are used for the smoothing step when calculating the principal curve. The activation function for the second hidden layer in the NN required tanh in this example. For the first hidden layer, only 5 neurons are required, while 13 is optimal for the second hidden layer. With these values of the hyperparameters, the learning rate does not significantly affect the results and 0.01 is used.

For sparse identification, the augmented library of functions consisted of the constant term, polynomials up to third order, sin(X), cos(X), tan(X), and tanh(X). The exponential terms are omitted in this example because, if they are included, while NLPCA-SI results improve, SI does not yield a proper solution. The optimal λ for the sparse identification algorithm was found to be 2.0 for the NLPCA-SI and 1.0 for SI. The sampling period in this example was chosen to be 0.005 hr because any value larger than this caused SI to not produce a proper solution.

Fig. 4 shows the results for this system. The reconstructed equations are omitted for brevity. The prediction of the slow state, C_A, is similar across both methods. The differences in Fig. 4(a) appear more significant due to the scale of the figure. The fast states, both the reactor and jacket temperatures, are reconstructed very accurately in the post-transient regime using both methods.

Remark 3. It is noted that the computational time needed to solve the original two-time-scale ODEs and the reduced-order slow model obtained with the proposed NLPCA-SI approach depends on the numerical integration method and time step of integration adopted. However, the lack of stiffness of the reduced-order slow model allows the use of larger time-steps leading to a reduced solution time for the calculation of the system solutions at the expense of missing the initial sharp transient of the fast states of the two-time-scale ODE system. Therefore, a fair comparison cannot be made when using different integration time steps for the different ODE systems identified, while the advantage of the NLPCA-SI method is nullified if an identical, very small integration time step is used for both systems.

5. Conclusion

In this work, reduced-order models were developed for nonlinear two-time-scale systems using measurement data. Nonlinear principal component analysis, a technique to extract nonlinear relationships between variables, was combined with sparse identification, an algorithm to reconstruct the underlying ODEs governing the system, to build a nonlinear model that can be used to predict the full state of the original system. The effectiveness of this method was demonstrated using two reactor-based examples and comparing the results with the original system and a purely sparse identified system. In both examples, the results were in close agreement with the original system and also the purely sparse identified system. It was observed, however, that the accuracy of the sparse identification method for the full state was strongly linked to the sampling period at which the original data was sampled and also, in some cases, required a very specific degree of sparsity in the algorithm. Furthermore, the sparse identified stiff ODEs required a very short integration time step of 10^{-6}, while the NN model in NLPCA-SI could predict the fast state (after a short transient) to at least a similar degree of accuracy without any integration. In the future, we will focus on the designs of controllers for nonlinear two-time-scale systems using data-based reduced-order models. The predicted states using NLPCA-SI can be used as the process model in model-based control methods such as MPC.

Conflict of interest

None declared.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgments

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

References

Arnaldo, I., O'Reilly, U.-M., Veeramachaneni, K., 2015. Building predictive models via feature synthesis. Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15, Association for Computing Machinery, New York, NY, USA, 983–990.

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.
Brunton, S.L., Tu, J.H., Bright, I., Kutz, J.N., 2014. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. SIAM J. Appl. Dyn. Syst. 13 (4), 1716–1732.
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.
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.
Christofides, P.D., Daoutidis, P., 1996. Feedback control of two-time-scale nonlinear systems. Int. J. Control. 63 (5), 965–994.
Cleveland, W.S., 1979. Robust locally weighted regression and smoothing scatterplots. J Am. Stat. Assoc. 74 (368), 829–836.
Dong, D., McAvoy, T., 1996. Nonlinear principal component analysis-based on principal curves and neural networks. Comput. Chem. Eng. 20 (1), 65–78.
Georgakis, C., 1977. A quasi-modal approach to model reduction. Proceedings of Joint American Control Conference, Vol. 14, San Fransisco, CA, USA, 639–644.
Hastie, T., Stuetzle, W., 1989. Principal curves. J Am. Stat. Assoc. 84 (406), 502–516.
Heineken, F.G., Tsuchiya, H.M., Aris, R., 1967. On the mathematical status of the pseudo-steady state hypothesis of biochemical kinetics. Math. Biosci. 1 (1), 95–113.
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.
Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Netw. 4 (2), 251–257.
Kimaev, G., Ricardez-Sandoval, L.A., 2019. Nonlinear model predictive control of a multiscale thin film deposition process using artificial neural networks. Chem. Eng. Sci. 207, 1230–1245.
Kimaev, G., Ricardez-Sandoval, L.A., 2020a. Artificial neural network discrimination for parameter estimation and optimal product design of thin films manufactured by chemical vapor deposition. J. Phys. Chem. C. 124 (34), 18615–18627.
Kimaev, G., Ricardez-Sandoval, L.A., 2020b. Artificial neural networks for dynamic optimization of stochastic multiscale systems subject to uncertainty. Chem. Eng. Res. Des. 161, 11–25.
Kokotović, P., Khalil, H.K., O'Reilly, J., 1999. Singular Perturbation Methods in Control: Analysis and Design. Society for Industrial and Applied Mathematics.
Kramer, M.A., 1991. Nonlinear principal component analysis using autoassociative neural networks. AIChE J. 37 (2), 233–243.
Lévine, J., Rouchon, P., 1991. Quality control of binary distillation columns via nonlinear aggregated models. Automatica 27 (3), 463–480.
Mackey, A., Schaeffer, H., Osher, S., 2014. On the compressive spectral method. Multiscale Model. Simul. 12 (4), 1800–1827.
Merrill, S.J., 1978. A model of the stimulation of B-cells by replicating antigen-II. Math. Biosci. 41 (1), 143–156.
Monge, J.J., Georgakis, C., 1987. The effect of operating variables on the dynamics of catalytic cracking processes. Chem. Eng. Commun. 60 (1–6), 1–26.
Ozolinš, V., Lai, R., Caflisch, R., Osher, S., 2013. Compressed modes for variational problems in mathematics and physics. Proc. Natl. Acad. Sci. 110 (46), 18368–18373.
Paluš, M., Dvorák, I., 1992. Singular-value decomposition in attractor reconstruction: pitfalls and precautions. Physica. D. 55 (1), 221–234.
Proctor, J.L., Brunton, S.L., Brunton, B.W., Kutz, J.N., 2014. Exploiting sparsity and equation-free architectures in complex systems. Eur. Phys. J. Spec. Top. 223 (13), 2665–2684.
Schaeffer, H., Caflisch, R., Hauck, C.D., Osher, S., 2013. Sparse dynamics for partial differential equations. Proc. Natl. Acad. Sci. 110 (17), 6634–6639.
Sontag, E.D., 1992. Feedback stabilization using two-hidden-layer nets. IEEE Trans. Neural Netw. 3 (6), 981–990.
Wang, W.-X., Yang, R., Lai, Y.-C., Kovanis, V., Grebogi, C., 2011. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. Phys. Rev. Lett. 106 (15), 154101.
Xu, L., Oja, E., Suen, C.Y., 1992. Modified hebbian learning for curve and surface fitting. Neural Netw 5 (3), 441–457.