Contents lists available at ScienceDirect

Chemical Engineering Research and Design

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

Machine-learning-based state estimation and predictive control of nonlinear processes

Mohammed S. Alhajeri[a], Zhe Wu[a], David Rincon[a], Fahad Albalawi[b,c], Panagiotis D. Christofides[a,d,*]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA
[b] Department of Electrical Engineering, College of Engineering, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
[c] National Center for Artificial Intelligence (NCAI) - Research Labs, Riyadh 11543, Saudi Arabia
[d] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA

ARTICLE INFO

Article history:
Received 5 December 2020
Received in revised form 7 January 2021
Accepted 12 January 2021
Available online 23 January 2021

Keywords:
Machine learning
Recurrent neural networks
State estimation
Model predictive control
Nonlinear systems
Chemical processes

ABSTRACT

Machine learning techniques have demonstrated their capability in capturing dynamic behavior of complex, nonlinear chemical processes from operational data. As full state measurements may be unavailable in chemical plants, this work proposes two machine-learning-based state estimation approaches. The first approach integrates recurrent neural networks (RNN) within the extended Luenberger observer framework to develop data-based state estimators. The second approach utilizes a hybrid model that integrates feed-forward neural networks with first-principles models to capture process dynamics in the state estimator. Then, an output feedback model predictive controller is designed based on the state estimates provided by the machine-learning-based estimators to stabilize the closed-loop system at the steady-state. A chemical process example is utilized to illustrate the effectiveness of the proposed machine-learning-based state estimation and control approaches.

1. Introduction

Closed-loop performance of chemical processes under model-based controllers (e.g., model predictive control (MPC)) depends on the model representation of the process, and the availability of real-time state measurements. In general, MPC uses a first-principles model or a data-driven process model to predict state evolution in the optimization problem, and adjusts its control actions with state feedback from the sensor measurements. However, measurements of key process states such as species concentration in a chemical reactor could be time-consuming and sometimes involves manual manipulation of samples during offline protocols (McKenna et al., 2000; Zambare et al., 2002). Additionally, the cost of equipment for getting the targeted measurement in real time also hinders its real-time application in chemical plants (Patwardhan et al., 2012). One way to address this issue is to combine measurable process state variables (e.g., pressure, level, and temperature measurements) with state estimation techniques to predict unmeasured states in real-time operation.

State estimation has been extensively studied in the literature, and includes methods for both deterministic and stochastic cases (Radke and Gao, 2006; Dochain, 2003; Patwardhan et al., 2012; Alexander et al., 2020). In stochastic state estimation, many methodologies have been proposed including recursive and optimization-based approaches, which can also address the constrained and unconstrained estimation problems. Extended Kalman filter

* 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.cherd.2021.01.009

(EKF) is one of the most popular recursive methods for unconstrained nonlinear systems. Moving horizon estimator is an optimization-based methodology that can account for constraints in its formulation. Additionally, other methodologies such as unscented Kalman filter, particle filter, constrained version of the EKF, and combination of the above methods, have been proposed to improve the performance of EKF (e.g., Lima and Rawlings, 2011; Patwardhan et al., 2012; Alexander et al., 2020). In deterministic state estimation, Luenberger-based observers are common estimation methods for the practitioners (Dochain, 2003; Ali et al., 2015). Additionally, extended Luenberger observer, sliding mode observer, adaptive state observer, high-gain observer, geometric observer, backstepping observer have found diverse applications in many fields (e.g., Ali et al., 2015). Similarities and differences among the above methods and their advantages and disadvantages are further discussed in Radke and Gao (2006) and Ali et al. (2015). In order to achieve a desired performance using these methodologies, a mathematical model for the targeted system is generally needed to describe process dynamics in a certain operating region. However, the development of such a process model for some complex reacting systems using first-principles knowledge could be challenging. For example for a catalytic carbon monoxide oxidation over Pt-alumina, a common Langmuir-Hinshelwood rate law is only valid in a small region of operation (Porru et al., 2000).

Machine learning has recently attracted an increasing level of attention in process modeling. Among many different machine learning methods, recurrent neural networks (RNN) and long-short-term-memory (LSTM) networks have been utilized to model dynamic systems due to their temporal dynamic behavior. Additionally, hybrid modeling that relies on both first-principles knowledge and process operational data can also be used to model nonlinear chemical processes and is one of the most interesting and challenging problems in the data science era (Venkatasubramanian, 2019). The idea of hybrid modeling is to use the best features of first-principles model (i.e., parametric models) and of data-driven models (i.e., nonparametric models) to better capture the process dynamics. There are many examples of hybrid modeling and their applications to chemical engineering problems in the literature, e.g., (Porru et al., 2000; Oliveira, 2004; Von Stosch et al., 2014; Zendehboudi et al., 2018; Bangi and Kwon, 2020; Lee et al., 2020). For example, in Bangi and Kwon (2020), a hybrid model was developed for a hydraulic fracturing process, where the process first-principles model was integrated with a deep neural network. In Porru et al. (2000), a neural network model was developed to represent the reaction kinetics and was coupled with the first-principles model to obtain a hybrid model that was successfully applied within the EKF. It is reported that hybrid models can not only augment the region of operation, but also provide a more general modeling framework that can build models faster and need no process insights (Wilson and Zorzetto, 1997).

Machine learning models can be utilized in model-based controllers to predict future states. Recently, in Wu et al. (2019a, 2020), machine-learning-based MPC schemes have been proposed to optimize process performance and ensure system stability with feedback measurements of process state variables assumed to be available. However, the assumption of availability of full state measurements for feedback control may not hold for the chemical processes with state variables difficult to measure in real time. In this work, we propose two machine learning approaches: (a) recurrent neural networks, and (b) hybrid models using feed-forward neural networks and first-principles models, to model nonlinear processes. Then, we integrate the RNN model and the hybrid model within the extended Luenberger observer framework and develop Lyapunov-based MPC using state estimates from machine-learning-based state estimators. Specifically, Section 2 introduces the preliminaries, including the class of systems, and the formulation of extended Luenberger observer. Section 3 presents the formulation of RNN models and of the RNN-based Luenberger observer. Section 4 presents the formulation of hybrid models and of the hybrid-model-based state estimator. Section 5 presents the formulation of output feedback model predictive controller that uses state estimates from the aforementioned machine-learning-based state estimators. Finally, in Section 6, a chemical reactor example is used to illustrate the effectiveness of the proposed estimation approaches.

2. Preliminaries

2.1. Notations

The Euclidean norm of a vector is represented by $|\cdot|$. The standard Lie derivative is represented as $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$. The notation $\backslash$ stands for set subsection, i.e., $A \backslash B = \{x \in \mathbf{R}^n | x \in A, x \notin B\}$. The function $f(\cdot)$ is said to be of class C^1 if it is continuously differentiable.

2.2. Class of systems

We consider the following class of continuous-time nonlinear systems in state-space form:

$$\dot{x} = F(x,u) := f(x) + g(x)u \tag{1a}$$

$$y = h(x) \tag{1b}$$

where the state vector is $x = [x_1, \ldots, x_n]^T \in \mathbf{R}^n$, the output vector is $y = [y_1, \ldots, y_q]^T \in \mathbf{R}^q$, and the input vector is $u = [u_1, \ldots, u_m]^T \in \mathbf{R}^m$. $F(x,u)$ is a nonlinear function with respect to x and u. The constraints on control inputs are given by $u \in U := \{u_i^{\min} \le u_i \le u_i^{\max}\}$. The function $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are matrices of dimension $n \times 1$, $n \times m$, and $q \times 1$ respectively.

2.3. Extended Luenberger Observer

Extended Luenberger observer (ELO) has been proposed for nonlinear processes as natural extension of Luenberger observer based on a linear approximation of the process (Zeitz, 1987; Dochain, 2003). The practical goal of the state observer is to provide an estimation of the unmeasured internal states of a given system by utilizing measured states from the process along with the implemented inputs. The extended Luenberger observer is presented in the following form for the nonlinear system of Eq. (1):

$$\dot{\hat{\mathbf{x}}} = F(\hat{x},u) + K(y - h(\hat{x})) \tag{2}$$

where $\hat{x}$ represents the estimated state vector, and the observer gain is denoted by K. The observer gain is also associated with desired properties from the state estimator and will be discussed in detail later. It is observed from Eq. (2) that the

first term is the process model, and the last term $K(y - h(\hat{x}))$ is known as the output prediction error, which is also considered as a correction term.

The goal of the ELO is to minimize the estimation error (i.e., $e = x - \hat{x}$) in which the dynamic of the error is determined by the following equation (Dochain, 2003; Mesbah et al., 2011):

$$\dot{e} = F(\hat{x} + e, u) - F(\hat{x}, u) - K(h(\hat{x} + e) - h(\hat{x})) \tag{3}$$

As shown in Eq. (3), the problem now is to determine under which conditions e can decay to zero. Therefore, it is important to design K to achieve this goal. In order to design K, Eq. (3) can be simplified to the following equation by linearizing the process model at a fixed point:

$$\dot{e} = (A - KL)e \tag{4}$$

where $A = \partial[F(x,u)/\partial x]_{x=\hat{x}}$ and $L = [\partial h(x,u)/\partial x]_{x=\hat{x}}$ are the linearized terms of the nonlinear system evaluated at a fixed-point (typically the operating steady-state). Finally, K is selected such that the eigenvalues of the matrix $A - KL$ have strictly negative real parts.

2.4. Stabilization via control Lyapunov function

We assume that there exists an observer and a feedback control law $u = \Phi(\hat{x}) \in U$ using the estimated states $\hat{x}$ from the observer to form an output feedback controller that can render the origin of the nonlinear system of Eq. (1) exponentially stable. This stabilizability assumption implies that there exists a $\mathcal{C}^1$ Control Lyapunov function $V(x)$ such that the following inequalities hold for all $x, \hat{x}$ in an open neighborhood D around the origin:

$$c_1|x|^2 \le V(x) \le c_2|x|^2 \tag{5a}$$

$$\frac{\partial V(x)}{\partial x} F(x, \Phi(\hat{x})) \le -c_3|x|^2 \tag{5b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \le c_4|x| \tag{5c}$$

where c_1, c_2, c_3 and c_4 are positive constants. $F(x,u)$ is the nonlinear system of Eq. (1). A candidate controller for $\Phi(\hat{x})$ is provided by the universal Sontag control law (Lin and Sontag, 1991). Then, following Wu et al. (2019a), we characterize the closed-loop stability region $\Omega_{\rho'}$ as a level set of Lyapunov function in the region D where the time-derivative of V is rendered negative under the controller $\Phi(\hat{x}) \in U$, i.e., $\Omega_{\rho'} := \{x \in D | V(x) \le \rho'\}$, where $\rho' > 0$. Additionally, based on the Lipschitz property of $F(x,u)$ and the boundedness of u, there exist positive constants M, L_x, L'_x such that the following inequalities hold for all $x, x' \in D$ and $u \in U$:

$$|F(x,u)| \le M \tag{6a}$$

$$|F(x,u) - F(x',u)| \le L_x|x - x'| \tag{6b}$$

$$\left|\frac{\partial V(x)}{\partial x}F(x,u) - \frac{\partial V(x')}{\partial x}F(x',u)\right| \le L'_x|x - x'| \tag{6c}$$

Remark 1. The assumption of the existence of an output feedback controller satisfying Eq. (5) requires that the observer states are bounded in $\Omega_{\rho'}$ and the estimate error, $e = x - \hat{x}$, converges to zero within finite time. In this work, the ELO of Eq. (2) and the Sontag control law (Sontag, 1989) are used as the observer and state feedback controller, respectively.

3. RNN-based state estimator

3.1. Recurrent neural network (RNN)

As a process model is needed in the extended Luenberger observer of Eq. (2). The following RNN model is developed to approximate the nonlinear system of Eq. (1) using process operational data when a first-principles model is not available:

$$\dot{\bar{x}} = F_{rnn}(\bar{x}, u) := A\bar{x} + \Theta^T y \tag{7}$$

where $\bar{x} = [\bar{x}_1, \ldots, \bar{x}_n]$ is the RNN state vector, and $u = [u_1, \ldots, u_m]$ is the manipulated input vector. $y = [y_1, \ldots, y_n, y_{n+1}, \ldots, y_{m+n}] = [\sigma(\bar{x}_1), \ldots, \sigma(\bar{x}_n), u_1, \ldots, u_m] \in \mathbf{R}^{n+m}$ is a vector of both $\bar{x}$ and u, where $\sigma(\cdot)$ is the nonlinear activation function. A is a diagonal coefficient matrix and $\Theta = [\theta_1, \ldots, \theta_n] \in \mathbf{R}^{(m+n)\times n}$ is a matrix with neural network weights to be optimized. The structures of unfolded and folded RNNs are shown in Fig. 1. The coefficient matrices $A = diag[-a_1, \ldots, -a_n]$, $a_i > 0, i = 1, \ldots, n$ and Θ consist of neural network weights that will be optimized during training. In practical implementation, the neural network weights are optimized to minimize the error between predicted outputs and the actual outputs in the training dataset.

After designing the RNN structure in terms of the number of layers and neurons and other hyper-parameters, the RNN is trained following the standard learning process as discussed in Wu et al. (2019a,b). Specifically, training, validation and testing datasets are generated from open-loop simulations of Eq. (1) under different initial conditions and control actions for a finite period of time. The continuous-time system of Eq. (1) is integrated using explicit Euler method with a sufficiently small integration time step h_c, and the control actions u are applied in a sample-and-hold fashion, i.e., $u(t) = u(t_k)$, $\forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$ and Δ is the sampling period. Since RNN models are able to capture process dynamic behavior from time-series data, the RNN model in this work is developed using all the integration time step data (i.e., data at each h_c step) within each sampling period to predict the state evolution for one sampling period. Additionally, as discussed in Wu et al. (2019a), the RNN models needs to satisfy a sufficiently small modeling error, i.e., $|\nu| = |F(x,u) - F_{rnn}(\bar{x},u)| \le \gamma|x| \le \nu_m$, between the RNN model and the nonlinear model of Eq. (1) during the training process such that it can well represent process dynamics in the operating region considered. Note that the modeling error ν is defined as the error between $\dot{x}$ from the first-principles model and the $\dot{x}$ predicted by the RNN model under the same input and states. The modeling error ν is not a constant for different states and input. However, since we limit the operation in the stability region Ω_ρ (i.e., both states and inputs are bounded), and the RNN model is trained using the datasets generated in Ω_ρ to achieve a very high accuracy, the modeling error can be made sufficiently small for all the states within Ω_ρ.

Remark 2. The RNN models are chosen in this work due to the ability of modeling a general class of nonlinear dynamical systems through the feedback loop in the hidden layer that

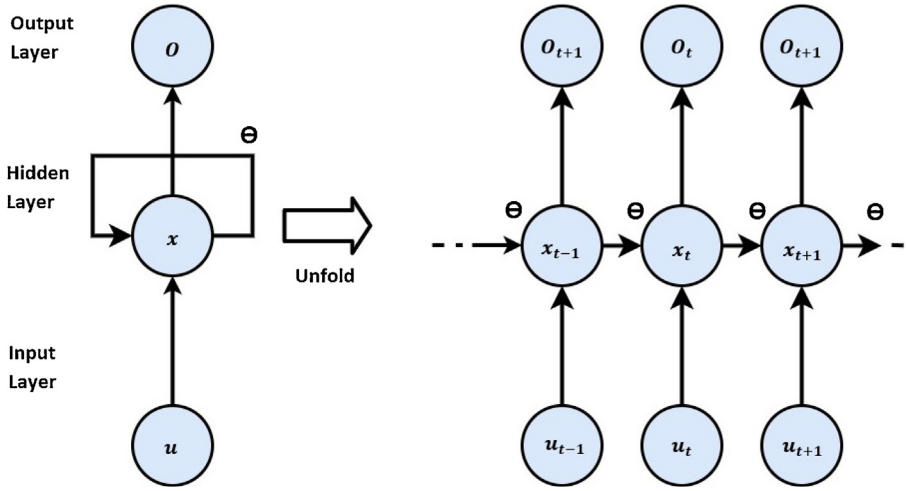

Fig. 1 – Structure of recurrent neural network.

introduces the past information to the current network (similar to the evolution of dynamical systems where past states values influence current values). The proposed approach can be extended to other machine learning approaches such as long short-term memory networks which are also widely used in modeling nonlinear dynamical systems (Ellis and Chinde, 2020; Chen et al., 2020). In addition to neural network modeling approaches, sparsity promoting algorithms, (extended) dynamic mode decompositions, and Koopman system identification have also been used to approximate nonlinear systems in recent works (Kutz et al., 2016; Narasingam and Kwon, 2019, 2020).

Remark 3. The RNN model in this work is developed using noise-free data from extensive open-loop simulations of Eq. (1) to capture process dynamics in the operating region $\Omega_{\rho'}$. In addition to computer simulations, datasets can also be generated using industrial measurements and experimental data. In the case that real industrial measurements are corrupted by noise from sensors variability and common plant variance, co-teaching training algorithm and dropout technique can be utilized in machine learning modeling approaches to improve the approximation performance by reducing the impact of noise. The interested reader is referred to Wu et al. (2021) for a detailed development of co-teaching and dropout methods.

Remark 4. Overfitting is one of the most common issues in the development of neural network models (Hassanpour et al., 2020; Wu et al., 2021). To reduce overfitting in this study, we start with a simple neural network with one layer and a few neurons, and keep increasing the number of layers and of neurons until no further improvement is noticed. We also use a large amount of data for validation such that the model performs well with respect to training and validation datasets.

3.2. *RNN-based state estimator*

The RNN model is then used in the extended Luenberger observer of Eq. (2) as follows:

$$\dot{\hat{x}} = F_{rnn}(\hat{x}, u) + K(y - h(\hat{x})) \tag{8}$$

Specifically, the state estimation based on the RNN model of Eq. (7) is obtained from the following steps. (1) Given an initial state estimate $\hat{x}(t_k)$ at time $t = t_k$ along with the manipulated input vector $u(t_k)$, the RNN model predicts the state at the next integration time step at $t = t_k + h_c$, then the state estimate at $t = t_k + h_c$ is obtained following Eq. (8) by adding the second term $h_c \times K(y - h(\hat{x}))$. (2) After the state estimate at $t = t_k + h_c$ is obtained, the above process is repeated with the same input u (because u remains constant within one sampling period). (3) Finally, the state estimate at the next sampling period $t = t_{k+1} := t_k + \Delta$ is obtained through $\frac{\Delta}{h_c}$ iterations of the above process.

Similarly, we assume that the RNN-based observer together with the state feedback control law $u = \Phi(\hat{x}) \in U$ form an output feedback controller that can render the origin of the RNN system of Eq. (7) exponentially stable. This implies that there exists a $\mathcal{C}^1$ Lyapunov function $V(x)$ such that the following inequalities hold for all x, $\hat{x}$ in an open neighborhood D around the origin:

$$\hat{c}_1|x|^2 \leq V(x) \leq \hat{c}_2|x|^2 \tag{9a}$$

$$\frac{\partial V(x)}{\partial x} F_{rnn}(x, \Phi(\hat{x})) \leq -\hat{c}_3|x|^2 \tag{9b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \leq \hat{c}_4|x| \tag{9c}$$

where $\hat{c}_1$, $\hat{c}_2$, $\hat{c}_3$ and $\hat{c}_4$ are positive constants. Note that the control law in this section is designed based on the RNN model of Eq. (7), while the control law in Section 2.4 is designed based on the first-principles model of Eq. (1). Subsequently, a new closed-loop stability region Ω_ρ can be characterized within D, where Eq. (9) is satisfied under $u = \Phi(\hat{x}) \in U$. Since the RNN model is trained with a sufficiently small modeling error, the state estimation through RNN-based state estimator of Eq. (8) is sufficiently close to the estimated value provided by Eq. (2) when the process model of Eq. (1) is known. The following proposition demonstrates that the controller $u = \Phi(\hat{x}) \in U$ designed based on the estimated state from RNN-based estimator is able to stabilize the system of Eq. (1) if the modeling error $|\nu| = |F(x, u) - F_{rnn}(\bar{x}, u)|$ is sufficiently small.

Proposition 1. *Consider the nonlinear system of Eq. (1) with an initial state $x_0 \in \Omega_\rho$ and a stabilizing control law $u = \Phi(\hat{x}) \in U$ based on the estimated states from Eq. (8), if the modeling error can be*

bounded, i.e., $|\nu| = |F(x,u) - F_{rnn}(x,u)| \le \gamma|x|$, *for all* $x \in \Omega_\rho$ *and* $u \in U$, *where* γ *is a positive real number satisfying* $\gamma < \hat{c}_3/\hat{c}_4$, *then the origin of the closed-loop system of Eq. (1) is rendered exponentially stable under* $u = \Phi(\hat{x}) \in U$ *for all* $x, \hat{x} \in \Omega_\rho$.

Proof. Based on Eq. (9), the time-derivative of V for the nonlinear system of Eq. (1) is derived as follows:

$$\begin{aligned}\dot{V} &= \frac{\partial V(x)}{\partial x} F(x, \Phi(\hat{x})) \\ &= \frac{\partial V(x)}{\partial x}(F_{rnn}(x, \Phi(\hat{x})) + F(x, \Phi(\hat{x})) - F_{rnn}(x, \Phi(\hat{x}))) \\ &\le -\hat{c}_3|x|^2 + \hat{c}_4|x|(F(x, \Phi(\hat{x})) - F_{rnn}(x, \Phi(\hat{x}))) \\ &\le -\hat{c}_3|x|^2 + \hat{c}_4\gamma|x|^2\end{aligned} \tag{10}$$

Therefore, $\dot{V} \le -(\hat{c}_3 - \hat{c}_4\gamma)|x|^2 \le 0$ holds if γ satisfies $\gamma < \hat{c}_3/\hat{c}_4$. This implies that for all $x_0 \in \Omega_\rho$, the origin of the nonlinear system of Eq. (1) is rendered exponentially stable under the controller $u = \Phi(\hat{x}) \in U$ with state estimates from RNN-based estimator.

4. Hybrid-model-based state estimator

In this section, we introduce a state estimator designed based on a hybrid model that integrates feed-forward neural network (FNN) with first-principles model. In this case, the FNN model is only used to approximate the nonlinear terms in Eq. (1), while the first-principles model of Eq. (1) can be derived from physical laws such as mass and energy balances.

4.1. *Feed-forward neural network (FNN)*

We develop a feed-forward neural network $F_{NN}(x)$ with input vector $x = [x_1, \ldots, x_n]$ and output vector $y = F_{NN}(x) = [y_1, \ldots, y_m]$ to approximate the nonlinear terms in Eq. (1). Fig. 2 shows the structure of a feed-forward neural network with three layers, i.e., input layer, hidden layer, and output layer. The hidden neurons h_j, $j = 1, \ldots, p$, and the outputs y_k are obtained using the following equations:

$$h_j = \sigma_1\left(\sum_{i=1}^{n} w_{ji}^{[1]}\ x_i + w_{j0}^{[1]}\right), \quad j = 1, 2, \ldots, p \tag{11}$$

$$y_k = \sigma_2\left(\sum_{i=1}^{p} w_{ki}^{[2]}\ h_i + w_{k0}^{[2]}\right), \quad k = 1, 2, \ldots, m \tag{12}$$

The weights in the first two layers are denoted by $w_{ji}^{[1]}$, and $w_{ki}^{[2]}$ respectively, with $w_{j0}^{[1]}$ and $w_{k0}^{[2]}$ representing biases. σ_1 and σ_2 are nonlinear activation functions such as hyperbolic tangent function, $\tanh(x) = \frac{2}{(1+\ e^{-2x})} - 1$, and logistic sigmoid function $S(x) = \frac{1}{(1+e^{-x})}$. The activation function σ_1 is utilized with a linear combination of input variables x_i in the calculation of hidden neurons h_j, while the output variable y_k is calculated through σ_2 with a linear combination of hidden neurons.

The development of FNN models also requires datasets for training and validating. Hence, a set of input vectors $\{x^n\}$ with the corresponding output vectors $\{\hat{y}^n\}$ are used in constructing datasets. The data can be collected in a variety of ways, including but not limited to experimental data and extensive computer simulations. The FNN model is trained to minimize the following loss function:

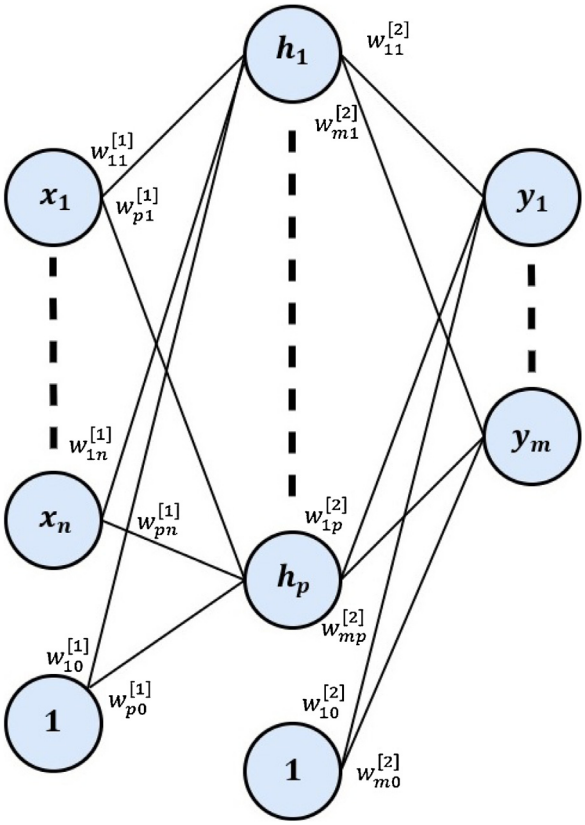

Fig. 2 – Three-layer feed-forward neural network structure with biases represented by neuron '1'.

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}|y(x^n, w) - \hat{y}^n|^2 \tag{13}$$

where N is the number of data points in training. The loss function is the sum of squared error between the predicted output by FNN model and the actual output in datasets. The weight vectors w are obtained by minimizing Eq. (13) via the gradient descent optimization method $w^{\iota+1} = w^\iota - \eta\nabla E(w^\iota)$, where the iteration is denoted by ι, and $\eta > 0$ is the learning rate.

4.2. *Hybrid-model-based estimator*

The hybrid model is developed by integrating the FNN model with the first-principles model of Eq. (1). The FNN model is used to represent the nonlinear terms, while the first-principles model is developed from the physical knowledge (Wu et al., 2020). The hybrid model for the system of Eq. (1) is in the following form:

$$\dot{x} = F_h(x, F_{NN}(x), u) \tag{14}$$

where F_h denotes the hybrid model and F_{NN} is the FNN model used to capture the static nonlinear relationship between the inputs and outputs of the nonlinear terms. The hybrid-model-based state estimator utilizes the hybrid model of Eq. (14) as the process model and includes the estimation correction term as follows:

$$\dot{\hat{x}} = F_h(\hat{x}, F_{NN}(\hat{x}), u) + K(y - h(\hat{x})) \tag{15}$$

F_h represents the hybrid model that is developed to capture the dynamics of the whole nonlinear system of Eq. (1). It should be noted that the hybrid model is used when a full first-principles model is unavailable. In that case, some of the nonlinearities like reaction rates are unknown and can be approximated by feedforward neural networks from data. Similar to the RNN-based estimator of Eq. (8), the hybrid-model-based state observer takes the state estimate $\hat{x}(t_{k-1})$ and the manipulated variable $u(t_{k-1})$ along with the last measurement $y(t_k)$ as inputs, and integrates Eq. (15) for one sampling period to estimate the states at the next sampling time. The explicit Euler method is used to integrate Eq. (15) with a small integration step h_c. Both estimators are initialized with initial conditions within Ω_ρ and if they calculate an estimate outside of Ω_ρ, then this estimate is discarded and is replaced with an estimate within Ω_ρ which ensures via the conditions of Eq. (5) that maintains closed-loop stability.

5. Output feedback model predictive control

In this section, an output feedback model predictive control (MPC) is designed based on state estimates provided by the RNN-based estimator to stabilize the nonlinear system of Eq. (1) at the steady-state. Specifically, the Lyapunov-based MPC is used in this work and the formulation is presented as the following optimization problem:

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), u(t))dt \tag{16a}$$

$$\text{s.t.} \quad \tilde{x}(t) = F_{rnn}(\tilde{x}(t), u(t)) \tag{16b}$$

$$u(t) \in U, \quad \forall t \in [t_k, t_{k+N}) \tag{16c}$$

$$\begin{aligned} &\tilde{x}(t_k) = \hat{x}(t_k) \\ &\dot{V}(\hat{x}(t_k), u) \leq \dot{V}(\hat{x}(t_k), \Phi(\hat{x}(t_k))), \end{aligned} \tag{16d}$$

$$if \hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{nn}} \tag{16e}$$

$$V(\tilde{x}(t)) \leq \rho_{nn}, \quad \forall t \in [t_k, t_{k+N}), \text{if} \hat{x}(t_k) \in \Omega_{\rho_{nn}} \tag{16f}$$

where $\tilde{x}$ is the predicted state trajectory, $S(\Delta)$ is the set of piecewise constant functions with period Δ, and N is the number of sampling periods in the prediction horizon. $\dot{V}(x, u)$ represents the time-derivative of V, i.e., $\frac{\partial V(x)}{\partial x}(F_{rnn}(x, u))$. The LMPC calculates the optimal input sequence $u^*(t)$ over the prediction horizon $t \in [t_k, t_{k+N})$, and sends the first control action $u^*(t_k)$ to the system to be applied for the next sampling period. Then the LMPC receives new measurements and is resolved with new state estimates at the next sampling time.

In the optimization problem of Eq. (16), Eq. (16a) is the objective function of LMPC that minimizes the time-integral of $L(\tilde{x}(t), u(t))$ over the prediction horizon subject to the following constraints. The constraint of Eq. (16b) is the RNN model of Eq. (7) for predicting state evolution given control actions and an initial state. Eq. (16c) is the input constraint. Eq. (16d) defines the initial condition $\tilde{x}(t_k)$ of Eq. (16b), which is the state estimates provided by the RNN-based state estimator of Eq. (8) at $t = t_k$. Specifically, given the state estimates at the previous time step, and the control actions, the estimation for the current state at $t = t_k$ is obtained following the steps as discussed in Section 3.2. Then, the state estimates $\hat{x}(t_k)$ is used as the initial state for the prediction model of Eq. (16b), and also in the constraints of Eq. (16e). If $\hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{nn}}$, the constraint of Eq. (16e) is activated, under which the state is forced to move towards the origin since $\Phi(\hat{x})$ is a stabilizing feedback control law. If the estimated state $\hat{x}(t_k)$ enters a small neighborhood around the origin, $\Omega_{\rho_{nn}}$, then the constraint of Eq. (16f) requires the states to remain inside $\Omega_{\rho_{nn}}$ for the entire prediction horizon.

The following theorem is established to demonstrate guaranteed closed-loop stability for the nonlinear system of Eq. (1) under the LMPC of Eq. (16) using state estimates from RNN-based estimator.

Theorem 1. *Consider the closed-loop system of Eq. (1) with an initial state $x_0 \in \Omega_\rho$ under the LMPC of Eq. (16). Let $\Delta > 0$, $\epsilon_w > 0$ and $\rho > \rho_{\min} > \rho_{nn} > \rho_s$ satisfy*

$$-\frac{\hat{c}_3 - \hat{c}_4\gamma}{\hat{c}_2}\rho_s + L'_x M\Delta \leq -\epsilon_w \tag{17a}$$

$$\rho_{nn} = \max\{V(\bar{x}(t_k + \Delta))|\hat{x}(t_k) \in \Omega_{\rho_s}, u \in U\} \tag{17b}$$

$$\rho_{\min} = \max\{V(x(t_k))|\bar{x}(t_k) \in \Omega_{\rho_{nn}}\}. \tag{17c}$$

Then, for any initial state $x_0 \in \Omega_\rho$, it is guaranteed that the state is bounded in the stability region for all times, i.e., $x(t) \in \Omega_\rho, \forall t \geq 0$, and $x(t)$ ultimately converges to $\Omega_{\rho_{\min}}$ for the closed-loop system of Eq. (1) under the LMPC of Eq. (16).

Proof. We first consider the estimated state $\hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{nn}}$ at $t = t_k$. In this case, the LMPC uses the constraint of Eq. (16e) to render the time-derivative of V under u less than that under the stabilizing controller $u = \Phi(\hat{x})$. We show that under the constraint of Eq. (16e), the state is able to move towards the origin over the next sampling period. Specifically, we derive the time-derivative of $V(x)$ under $u = \Phi(\hat{x})$ for the nonlinear system of Eq. (1) as follows:

$$\begin{aligned} \dot{V}(x(t)) &= \frac{\partial V(x(t))}{\partial x}F(x(t), \Phi(\hat{x}(t_k))) \\ &= \frac{\partial V(x(t_k))}{\partial x}F(x(t_k), \Phi(\hat{x}(t_k))) \\ &\quad + \frac{\partial V(x(t))}{\partial x}F(x(t), \Phi(\hat{x}(t_k))) \\ &\quad - \frac{\partial V(x(t_k))}{\partial x}F(x(t_k), \Phi(\hat{x}(t_k))) \end{aligned} \tag{18}$$

Since Eq. (10) shows that the controller $u = \Phi(\hat{x})$ can render the time-derivative of $V(x(t_k))$ negative, i.e., $\frac{\partial V(x(t_k))}{\partial x}F(x(t_k), \Phi(x(t_k))) \leq -\hat{c}_3|x(t_k)|^2 + \hat{c}_4\gamma|x(t_k)|^2 < 0, \ \forall x(t_k) \neq 0$, we can further derive the following inequality for $t \in [t_k, t_{k+1})$:

$$\begin{aligned} \dot{V}(x(t)) &\leq -\frac{\hat{c}_3 - \hat{c}_4\gamma}{\hat{c}_2}\rho_s + \frac{\partial V(x(t))}{\partial x}F(x(t), \Phi(\hat{x}(t_k))) \\ &\quad - \frac{\partial V(x(t_k))}{\partial x}F(x(t_k), \Phi(\hat{x}(t_k))) \\ &\leq -\frac{\hat{c}_3 - \hat{c}_4\gamma}{\hat{c}_2}\rho_s + L'_x|x(t) - x(t_k)| \\ &\leq -\frac{\hat{c}_3 - \hat{c}_4\gamma}{\hat{c}_2}\rho_s + L'_x M\Delta \end{aligned} \tag{19}$$

Therefore, if Eq. (17a) is satisfied, the time-derivative of $V(x)$ under $u = \Phi(\hat{x})$ for the nonlinear system of Eq. (1) is rendered negative for the next sampling time, which implies the state of the system of Eq. (1) will move towards the origin under the constraint of Eq. (16e). It should be noted that the state estimate $\hat{x}$ is assumed to be bounded in Ω_ρ. If the estimate is outside of Ω_ρ, we discard it and replace with an estimate inside the Ω_ρ. This new estimate could be a state inside Ω_ρ that is closest to the original estimate. Since the state estimate is bounded in Ω_ρ for all times, and closed-loop stability is ensured under the output feedback controller assumed in Section 3.2 (i.e., the state feedback control law $u = \Phi(\hat{x})$ designed based on the estimated states from RNN observer), the true state can be driven into a small neighborhood around the origin within finite sampling periods. If the estimated state enters $\Omega_{\rho_{nn}}$, the LMPC activates the constraint of Eq. (16f) to maintain the predicted states of the RNN model within $\Omega_{\rho_{nn}}$ over the prediction horizon. However, since there exists a model mismatch between the nonlinear system of Eq. (1) and the prediction model of Eq. (16b) (i.e., the RNN model), we need to show that the actual state of the nonlinear system of Eq. (1) is bounded in a small neighborhood around the origin under LMPC. To that end, we characterize the set $\Omega_{\rho_{\min}}$ of Eq. (17c) to account for the sufficiently small modeling error between the RNN model and the nonlinear system of Eq. (1). Eq. (17c) shows that if the RNN predicted state $\overline{x}(t_k)$ is inside $\Omega_{\rho_{nn}}$, then the actual state of the nonlinear system of Eq. (1) is bounded in $\Omega_{\rho_{\min}}$. This completes the proof of closed-loop stability for the system of Eq. (1) under LMPC. □

Remark 5. The formulation of the LMPC using hybrid-model-based estimator is very similar to that of Eq. (16) using RNN-based estimator, and therefore, is omitted here. The only difference in the LMPC formulation would be the prediction model of Eq. (16b), for which the hybrid model of Eq. (14) will be used to replace the RNN model. Additionally, the hybrid model will also be used to provide state estimates at each sampling time in LMPC. The closed-loop stability analysis for hybrid-model-based estimator is also similar to Theorem 1 based on the fact that a well-conditioned feed-forward neural network is obtained to represent the nonlinear terms with a sufficiently high accuracy.

6. Application to a chemical reactor example

In this section, a nonlinear chemical process is used to illustrate the application of the proposed RNN-based and hybrid-model-based estimators in the LMPC controller. A non-isothermal, a well mixed continuous stirred tank reactor (CSTR) is considered, with the following reversible first-order exothermic reaction (Zhang et al., 2019):

$$A \leftrightarrow B$$

The nonlinear dynamical model that describes the process dynamics is given by the following mass and energy balance equations:

$$\frac{dC_A}{dt} = \frac{1}{\tau}(C_{A0} - C_A) - r_A + r_B \tag{20a}$$

$$\frac{dC_B}{dt} = \frac{-1}{\tau}C_B + r_A - r_B \tag{20b}$$

Table 1 – Parameter and steady-state values for the CSTR.

$T_o = 400\,K$	$T_s = 426.743\,K$
$k_A = 5000/s$	$E_A = 1 \times 10^4$ cal/mol
$k_B = 10^6/s$	$E_B = 1.5 \times 10^4$ cal/mol
$R = 1.987$ cal/(mol K)	$\Delta H = -5000$ cal/mol
$\rho = 1$ kg/L	$C_p = 1000$ cal/(kg K)
$C_{A0} = 1$ mol/L	$V = 100$ L
$C_{A_s} = 0.4977$ mol/L	$\tau = 60$ s
$C_{B_s} = 0.5023$ mol/L	$Q_s = 40,386$ cal/s

$$\frac{dT}{dt} = \frac{1}{\tau}(T_0 - T) + \frac{-\Delta H}{\rho C_p}(r_A - r_B) + \frac{Q}{\rho C_p V} \tag{20c}$$

$$r_A = k_A e^{\frac{-E_A}{RT}} C_A \tag{20d}$$

$$r_B = k_B e^{\frac{-E_B}{RT}} C_B \tag{20e}$$

The concentration of A and B in the CSTR are given by C_A and C_B respectively, and T represents the reactor temperature. The feed concentration is denoted by C_{A0} and the feed temperature is denoted by T_0. As for the reaction kinetics, k_A and E_A represent the pre-exponential constant and the activation energy for the forward reaction, while k_B and E_B are for the reverse reaction. The reactor residence time is denoted by τ. V represents the reactor volume, ΔH is the reaction enthalpy, and the heat capacity of the mixture liquid is denoted by C_p. The CSTR is equipped with a heating/cooling jacket to provide/remove required heat at rate Q to/from the reactor. Zhang et al. (2019) has provided the optimal steady-states for the process described in Eq. (20). The optimal steady-state values and process parameter values are listed in Table 1.

6.1. Simulation settings

The control objective is to drive C_A, C_B, and T to the steady-state by manipulating the heat input rate Q. The manipulated variable is considered in the deviation form as $u = Q - Q_s$. The control action u is bounded with upper bound $u^{UB} = 40,000$ cal/s and a lower bound $u^{LB} = -40,000$ cal/s. The process states are all represented in the deviation form. The optimal steady-state is at $x^T = [x_1 \;\; x_2 \;\; x_3] = [C_A - C_{A_s} \;\; C_B - C_{B_s} \;\; T - T_s]$ such that the origin is the equilibrium point of this system. Since in practice not all process states are measurable (Kurtz and Henson, 1998), unmeasurable states needs to be estimated based on measurable states. In this case study, we assume that the only measured state is $x_3 = T - T_s$. Therefore, $x_1 = C_A - C_{A_s}$ and $x_2 = C_B - C_{B_s}$ can be estimated using the proposed machine-learning-based state estimators. Based on the measurement y of the state variable x_3, the machine-learning-based observer first utilizes the RNN model (or hybrid model) to predict x_1 and x_2, and then add the estimation error part ($K(y - \hat{x}_3)$) to obtain the state estimates at the current time step. Subsequently, the estimated states $\hat{x}^T = [\hat{x}_1 \;\; \hat{x}_2 \;\; \hat{x}_3]$ are sent to the MPC for solving the optimal control action for the next sampling period. Additionally, the nonlinear system of Eq. (20) is observable for the given output (i.e., temperature T) in the sense there exists an estimator-based output feedback controller that exponentially stabilizes the closed-loop system.

The nonlinear optimization problem of LMPC is solved using the IPOPT software package (Wächer and Biegler, 2006), and its python version, PyIpopt, with the sampling period $\Delta = 10$ s. The objective function of LMPC is of the form:

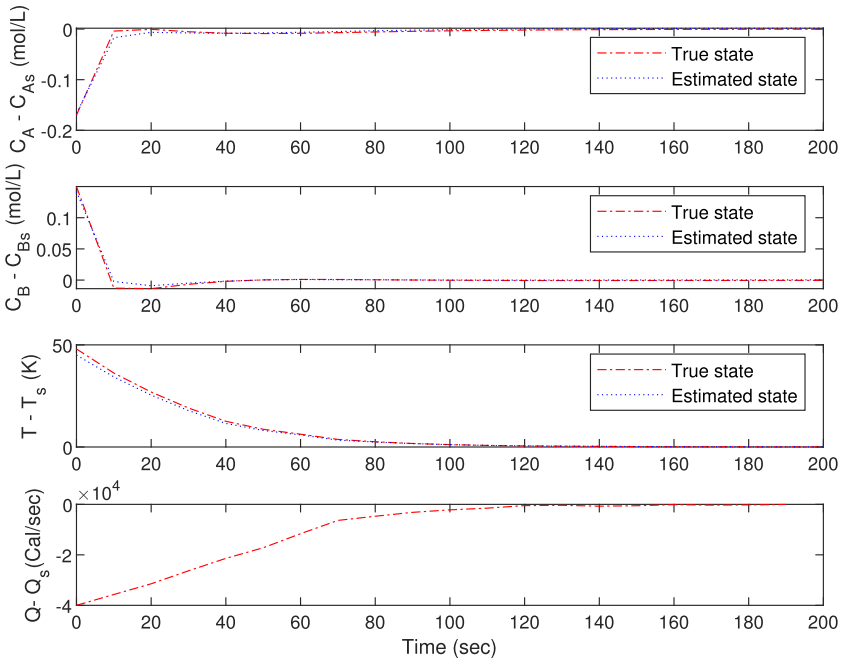

Fig. 3 – True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC using RNN-based estimator with the initial condition IC_1 (top three plots). The bottom plot displays the manipulated input profile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

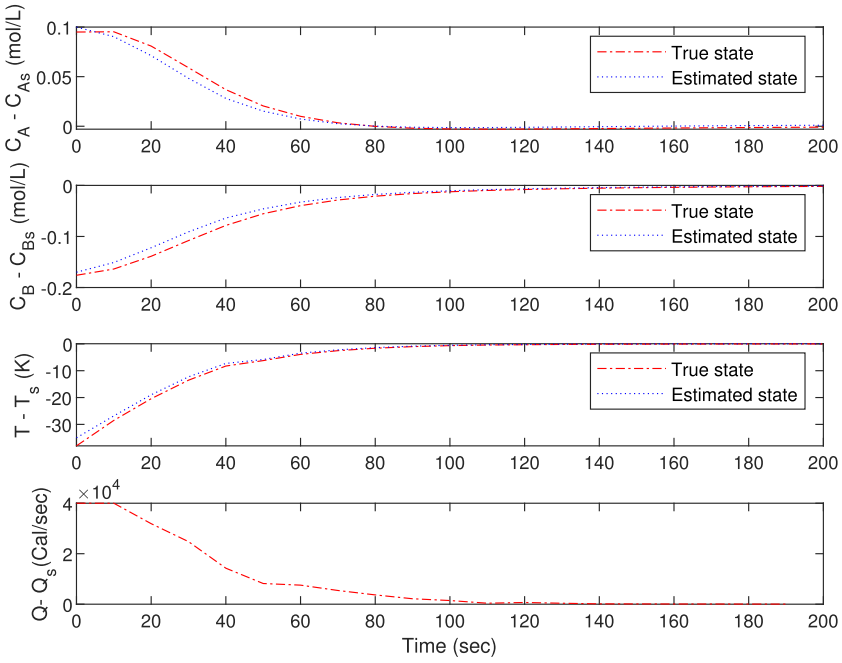

Fig. 4 – True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC using RNN-based estimator with the initial condition IC_2 (top three plots). The bottom plot displays the manipulated input profile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$L(x,u) = x^T Q x + u^T R u$, where $Q = diag[5 \times 10^4 \quad 5 \times 10^4 \quad 1]$, and $R = [10^{-7}]$ are penalty matrices that should be tuned properly to achieve a better MPC performance (Alhajeri and Soroush, 2020). The observer gains used in this work are $K^T = [0.0005, 0.0005, 0.5]$. The Lyapunov function is given by $V(x) = x^T P x$, with the following positive definite P matrix:

$$P = \begin{bmatrix} 625 & 0 & 0 \\ 0 & 625 & 0 \\ 100 & 100 & 10^5 \end{bmatrix}$$

6.2. *Neural networks model training*

The data generation, neural network training and validation process for the RNN model are carried out as follows. To generate the dataset for RNN model, the system of Eq. (20) was numerically integrated for one sampling period under different initial conditions. The explicit Euler method with an integration time step of $h_c = 0.5$ s is utilized. Specifically, a data set of size 1.6×10^6 was built using MATLAB. The data base was then divided into an input matrix with u, x_1, x_2, x_3 at $t = t_k$ and an output matrix with x_1, x_2, and x_3 as outputs at $t = t_{k+1}$, from

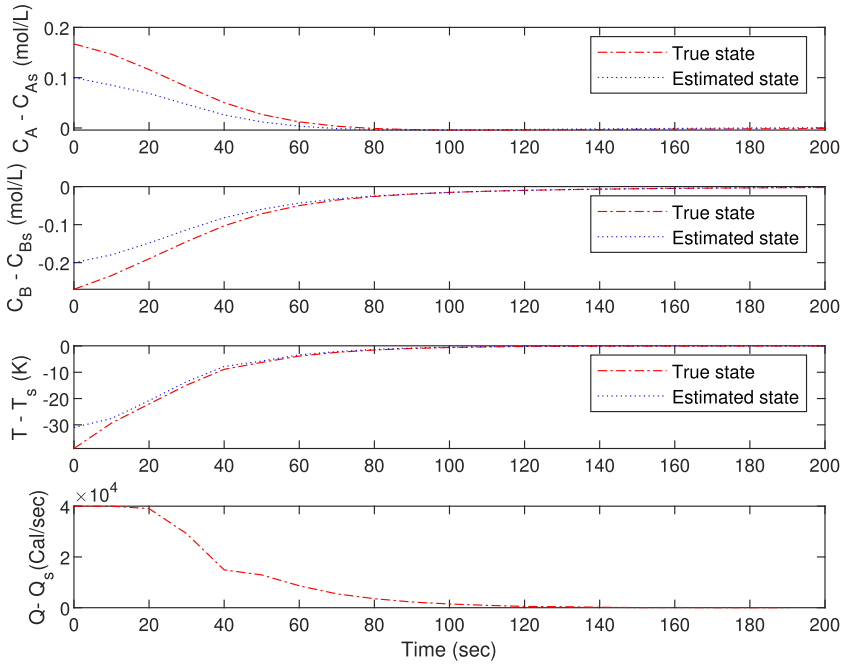

Fig. 5 – True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC using RNN-based estimator with the initial condition IC_3 (top three plots). The bottom plot displays the manipulated input profile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

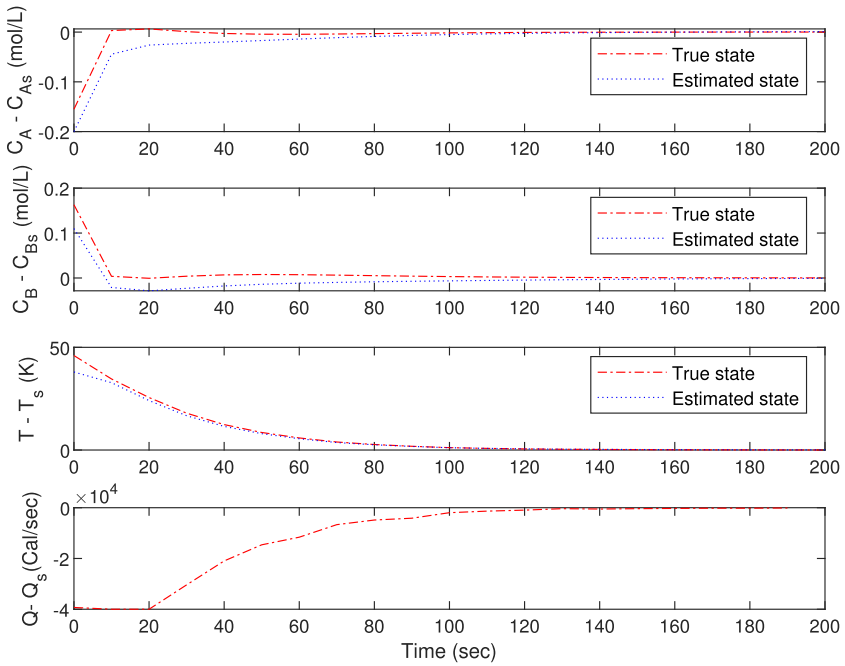

Fig. 6 – True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC using RNN-based estimator with the initial condition IC_4 (top three plots). The bottom plot displays the manipulated input profile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which 70% of the data was utilized for model training, and 30% was for validation. Note that the full state measurements are available in the training stage as the data can be obtained offline, while in real-time operation of CSTR, only the temperature can be measured every sampling time. The RNN model was developed using Keras library with two hidden layers of 50 units in each layer and tanh activation function, and an output layer with 3 neurons and linear activation function. 274 epochs were used for the training process.

Similarly, the data was generated for feed-forward neural networks using MATLAB simulations of different values of x_1, x_2, and x_3 in the reaction kinetics model of Eq. (20). A dataset consisting of 1.25×10^5 data points was generated with x_1, x_2, and x_3 as inputs, and reaction rate as the output. Keras library was used for the FNN model training with three inputs and one output (i.e., the output is $r_A - r_B$). Three layers were used with 6 neurons, 12 neurons, and 1 neuron in each layer, respectively. *Relu* activation function was utilized in the first two layers and sigmoid activation function was used for the last layer.

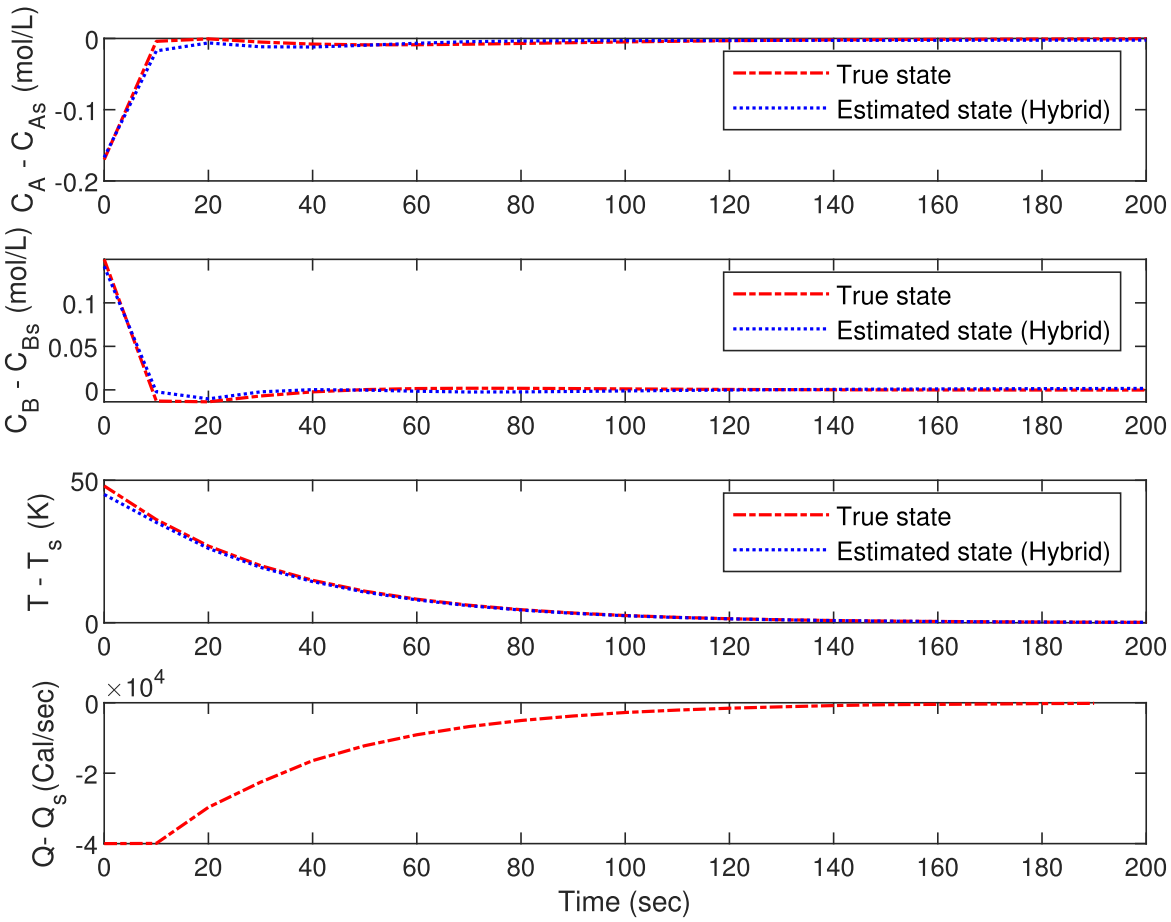

Fig. 7 – True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC using hybrid-model-based estimator with the initial condition IC_1 (top three plots). The bottom plot displays the manipulated input profile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

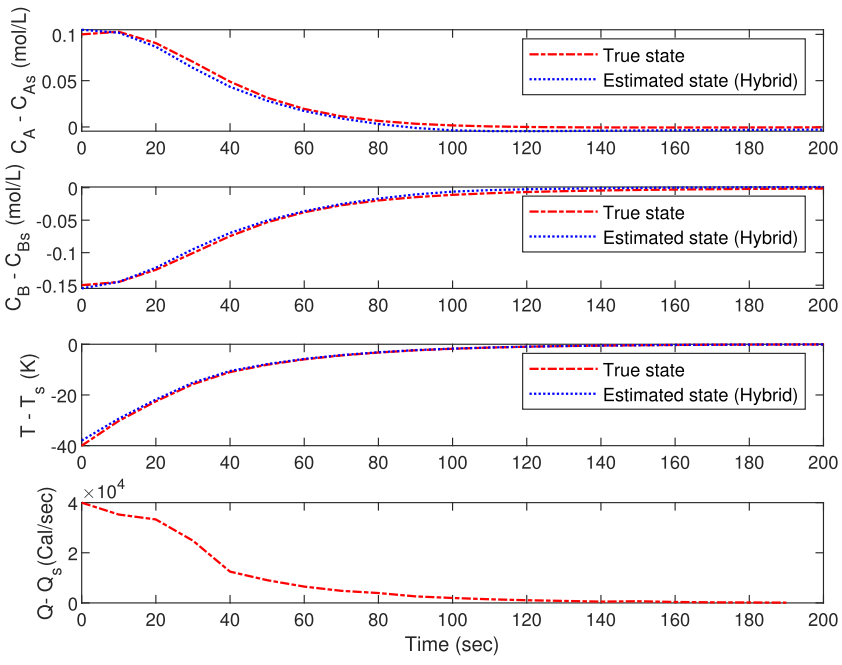

Fig. 8 – True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC using hybrid-model-based estimator with the initial condition IC_2 (top three plots). The bottom plot displays the manipulated input profile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

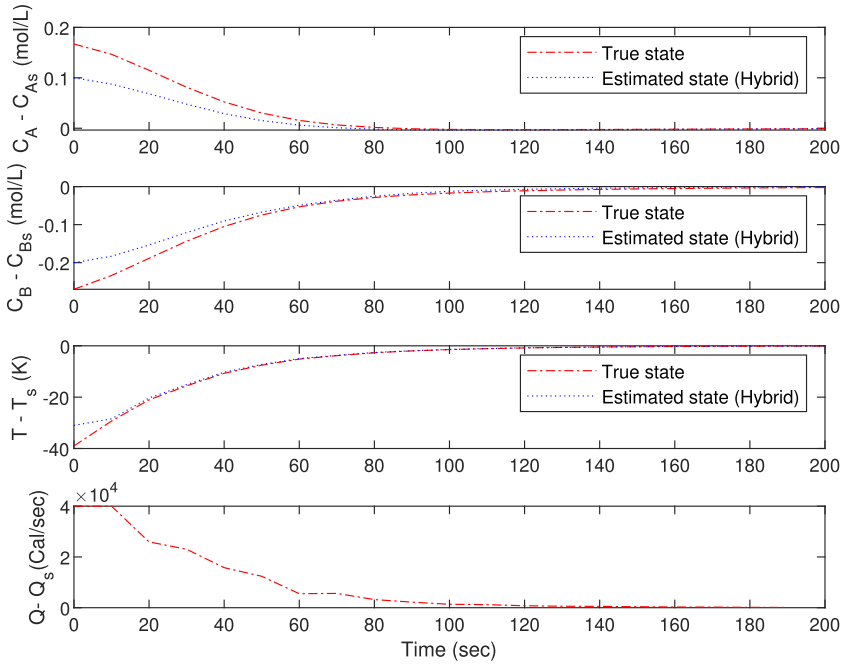

Fig. 9 – True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC using hybrid-model-based estimator with the initial condition IC_3 (top three plots). The bottom plot displays the manipulated input profile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Remark 6. The RNN is developed to approximate the process model using all the states including temperature and species concentrations. The RNN outputs are the process states in the first-principles model. To build the state estimator, the RNN model is used to replace the process model $F(\hat{x}, u)$ in the extended Luenberger observer of Eq. (2). The training dataset is generated from simulations of the first-principles model, and the RNN model is trained with a sufficiently high accuracy, which guarantees that the RNN model predictions are sufficiently close to the estimates of the first-principles model. As the RNN model is developed to approximate the process model, the datasets include all the process states.

Table 2 – Estimation mean squared error of the closed-loop CSTR under LMPC using RNN-based and hybrid-model-based state estimators.

Model	Simulation no.	MSE of x_1	MSE of x_2
RNN model	1	1.3699×10^{-5}	9.9753×10^{-6}
	2	1.9458×10^{-5}	5.606×10^{-5}
	3	6.0499×10^{-4}	5.3197×10^{-4}
	4	3.24×10^{-4}	7.41×10^{-4}
Hybrid model	1	1.6198×10^{-5}	1.409×10^{-5}
	2	1.5112×10^{-5}	1.266×10^{-5}
	3	5.8035×10^{-4}	4.5938×10^{-4}
	4	3.6534×10^{-4}	3.9027×10^{-4}

Remark 7. Hybrid modeling approaches require a careful selection of process parameter/variable in first-principles models which will be estimated through data based approaches. Local/global sensitivity analysis is one of the most common methods for selecting such parameters during model identification for hybrid representations (Haaker and Verheijen, 2004; Madar et al., 2005; Von Stosch et al., 2014). In this example, the nonlinear terms in a nonlinear process (i.e., the reaction rates in CSTR first-principles equations) are chosen to be represented by neural networks to better capture the nonlinearities in a wide operating region.

6.3. *Closed-loop simulation results*

Closed-loop simulation study is carried out to demonstrate the performance of the two proposed estimation approaches in the CSTR of Eq. (20). The closed-loop simulations results using the RNN-based estimator with four different sets of initial conditions, IC_1, IC_2, IC_3, and IC_4 are shown in Figs. 3–6, and the closed-loop simulation results using hybrid-model-based estimator with the same four initial conditions are shown in Figs. 7–10.

It can be seen from Figs. 3–6 that starting from different initial conditions and different initial estimates, the closed-loop states are stabilized at the steady-state under LMPC using RNN-based state estimator. Specifically, in Figs. 3 and 4, we consider two initial estimates that are very close to the true state values, where the true states are obtained from the first-principles model of Eq. (20). It is demonstrated that the state estimates provided by the RNN-based estimator converge to the true state value quickly, and after that, the closed-loop states are driven the steady-state smoothly. In Figs. 5 and Fig. 6, we consider two initial estimates that are not close to the true state values at the beginning. It is demonstrated that the state estimates still converge to the true states but takes longer time than those in Figs. 3 and 4 . In all cases, closed-loop stability is achieved for the system under LMPC.

Subsequently, the mean squared errors (MSE) between true state profiles and estimate state profiles are used to evaluate the performance of the estimator. Table 2 summarized the MSE of state estimation using the RNN-based state estimator in the four closed-loop simulations. It is shown that all the closed-loop simulations achieve sufficiently small MSEs, and the simulations with IC_1 and IC_2 achieve better results due to better initial estimates. This is consistent with the closed-

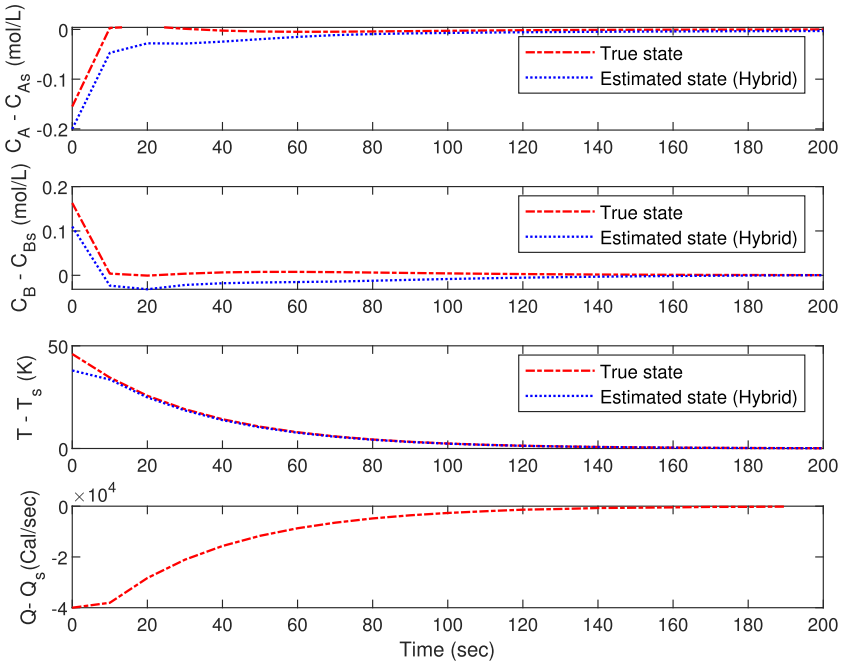

Fig. 10 – True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC using hybrid-model-based estimator with the initial condition IC_4 (top three plots). The bottom plot displays the manipulated input profile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

loop simulation results as shown in Figs. 3–6 . Therefore, from this simulation study of CSTR example, it is demonstrated that the RNN-based estimator can estimate true state values with a sufficiently high accuracy.

The closed-loop simulation results using hybrid-model-based estimator with the same four initial conditions are shown in Figs. 7–10, and the MSE results are also summarized in Table 2. The closed-loop stability analysis and the MSE results are similar to those using RNN-based estimator, and are omitted here.

7. Conclusion

In this work, we proposed machine-learning-based state estimation approaches for nonlinear processes. The RNN model was first developed to represent process dynamics in the operating region, and incorporated in extended Luenberger observer. Then, the RNN-based estimator was used to provide state estimates for the optimization problem of LMPC. Subsequently, a hybrid model was develop to represent process dynamics and used in the state estimator. From closed-loop simulations, it was demonstrated that both the RNN-based estimator and the hybrid-model-based estimator achieved a desired accuracy in state estimation, and all the state trajectories initiating from different initial conditions converged to the steady-state under the LMPC using machine-learning-based estimators.

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. Mohammed Alhajeri would like to express his sincere appreciation to Kuwait University for its support through the KU-scholarship program. Fahad Albalawi acknowledges Taif University for their support via Taif University Researchers Supporting Project (TURSP-2020/97).

References

Alexander, R., Campani, G., Dinh, S., Lima, F., 2020. Challenges and opportunities on nonlinear state estimation of chemical and biochemical processes. Processes 8, 1462.

Alhajeri, M., Soroush, M., 2020. Tuning guidelines for model-predictive control. Ind. Eng. Chem. Res. 59, 4177–4191.

Ali, J.M., Hoang, N.H., Hussain, M.A., Dochain, D., 2015. Review and classification of recent observers applied in chemical process systems. Comput. Chem. Eng. 76, 27–41.

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

Chen, S., Wu, Z., Rincon, D., Christofides, P.D., 2020. Machine learning-based distributed model predictive control of nonlinear processes. AIChE J. 66, e17013.

Dochain, D., 2003. State and parameter estimation in chemical and biochemical processes: a tutorial. J. Process Control 13, 801–818.

Ellis, M.J., Chinde, V., 2020. An encoder-decoder LSTM-based EMPC framework applied to a building HVAC system. Chem. Eng. Res. Des. 160, 508–520.

Haaker, M.P.R., Verheijen, P.J.T., 2004. Local and global sensitivity analysis for a reactor design with parameter uncertainty. Chem. Eng. Res. Des. 82, 591–598.

Hassanpour, H., Corbett, B., Mhaskar, P., 2020. Integrating dynamic neural network models with principal component analysis for adaptive model predictive control. Chem. Eng. Res. Des. 161, 26–37.

Kurtz, M.J., Henson, M.A., 1998. State and disturbance estimation for nonlinear systems affine in the unmeasured variables. Comput. Chem. Eng. 22, 1441–1459.

Kutz, J.N., Brunton, S.L., Brunton, B.W., Proctor, J.L., 2016. Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems. SIAM.
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, e1008472.
Lima, F.V., Rawlings, J.B., 2011. Nonlinear stochastic modeling to improve state estimation in process monitoring and control. AIChE J. 57, 996–1007.
Lin, Y., Sontag, E.D., 1991. A universal formula for stabilization with bounded controls. Syst. Control Lett. 16, 393–397.
Madar, J., Abonyi, J., Szeifert, F., 2005. Feedback linearizing control using hybrid neural networks identified by sensitivity approach. Eng. Appl. Artif. Intell. 18, 343–351.
McKenna, T., Othman, S., Fevotte, G., Santos, A., Hammouri, H., 2000. An integrated approach to polymer reaction engineering: a review of calorimetry and state estimation. Polym. React. Eng. 8, 1–38.
Mesbah, A., Huesman, A.E., Kramer, H.J., Van den Hof, P.M., 2011. A comparison of nonlinear observers for output feedback model-based control of seeded batch crystallization processes. J. Process Control 21, 652–666.
Narasingam, A., Kwon, J.S., 2019. Koopman Lyapunov-based model predictive control of nonlinear chemical process systems. AIChE J. 65, e16743.
Narasingam, A., Kwon, J.S., 2020. Data-Driven Feedback Stabilization of Nonlinear Systems: Koopman-Based Model Predictive Control. arXiv:2005.09741.
Oliveira, R., 2004. Combining first principles modelling and artificial neural networks: a general framework. Comput. Chem. Eng. 28, 755–766.
Patwardhan, S.C., Narasimhan, S., Jagadeesan, P., Gopaluni, B., Shah, S.L., 2012. Nonlinear bayesian state estimation: a review of recent developments. Control Eng. Pract. 20, 933–953.
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, 331–338.
Radke, A., Gao, Z., 2006. A survey of state and disturbance observers for practitioners. Proceedings of the American Control Conference, Minneapolis, Minnesota, 5183–5188.
Sontag, E.D., 1989. A 'universal' construction of Artstein's theorem on nonlinear stabilization. Syst. Control Lett. 13, 117–123.
Venkatasubramanian, V., 2019. The promise of artificial intelligence in chemical engineering: is it here, finally? AIChE J. 65, 466–478.
Von Stosch, M., Oliveira, R., Peres, J., de Azevedo, S.F., 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, J., Zorzetto, L., 1997. A generalised approach to process state estimation using hybrid artificial neural network/mechanistic models. Comput. Chem. Eng. 21, 951–963.
Wu, Z., Rincon, D., Christofides, P.D., 2020. Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes. J. Process Control 89, 74–84.
Wu, Z., Rincon, D., Luo, J., Christofides, P.D., 2021. Machine learning modeling and predictive control of nonlinear processes using noisy data. AIChE J. 67, 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.
Zambare, N., Soroush, M., Grady, M.C., 2002. Real-time multirate state estimation in a pilot-scale polymerization reactor. AIChE J. 48, 1022–1033.
Zeitz, M., 1987. The extended Luenberger observer for nonlinear systems. Syst. Control Lett. 9, 149–156.
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, Z., Wu, Z., Rincon, D., Christofides, P.D., 2019. Real-time optimization and control of nonlinear processes using machine learning. Mathematics 7, 890.