Estimation-Based Predictive Control of Nonlinear Processes Using Recurrent Neural Networks [★]

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

[*] *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA (email: malhajeri@g.ucla.edu)*
[**] *Department of Electrical Engineering, College of Engineering, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia.*
[***] *National Center for Artificial Intelligence (NCAI) - Research Labs, Riyadh 11543, Saudi Arabia*
[****] *Department of Chemical and Biomolecular Engineering and Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA (email: pdc@seas.ucla.edu)*

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 integrates recurrent neural networks (RNN) within extended Luenberger observers to develop data-based state estimators. Then, an output feedback model predictive controller is designed based on state estimates provided by the RNN-based estimator to stabilize the closed-loop system at the steady-state. A chemical process example is utilized to illustrate the effectiveness of the proposed state estimation approach.

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

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 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 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 equipments for getting the targeted measurement in real time also hinders its application in chemical plants (Patwardhan et al. (2012)). One way to address this issue is to use measurable process state variables (e.g., pressure, level, and temperature) with state estimation techniques to predict unmeasurable 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 the deterministic state estimation case, Luenberger-based observers are common source of filters 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 (Ali et al. (2015)). Similarities and difference between the above methods together with their advantages and disadvantages are further discussed in Radke and Gao (2006); Ali et al. (2015). In order to achieve a better 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)). An alternative method of process modeling is using machine learning techniques with process data. Among many different machine learning methods, recurrent neural networks

[★] 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.

(RNN) and Long-Short-Term-Memory (LSTM) networks have attracted an increasing level of attention due to their ability of modeling dynamic systems. Recently, in Wu et al. (2019a, 2020b), machine-learning-based MPC schemes have been proposed to optimize process performance and ensure system stability with feedback measurements of process state variables available. However, the assumption of full state measurements available for feedback control may not hold for the chemical processes with some state variables difficult to measure in real time.

As both MPC and state estimation need a process model for predicting future states, this work utilizes machine learning modeling techniques to develop a data-driven model that can be efficiently implemented in both state estimation and output feedback controllers. Specifically, we develop an RNN model for a general class of nonlinear systems and integrate the RNN model within the extended Luenberger observer and Lyapunov-based MPC that uses state estimates in optimization. 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 output feedback model predictive controller that uses state estimates from RNN-based state estimator. Finally, in Section 5, a chemical reactor example is used to illustrate the effectiveness of the proposed estimation approach.

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 $\setminus$ stands for set subsection, i.e., $A \setminus B = \{x \in \mathbf{R}^n | x \in A, x \notin B\}$. 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 $x = [x_1, ..., x_n]^T \in \mathbf{R}^n$, the output vector is $y = [y_1, ..., y_q]^T \in \mathbf{R}^q$, and the input vector is $u = [u_1, ..., u_m]^T \in \mathbf{R}^m$. $F(x,u)$ is a nonlinear function with respect to x and u. The constrains on control inputs is given by u $\in U := \{{u_i}^{min} \leq u_i \leq {u_i}^{max}\}$. 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.

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

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 shade some light of the choice of K, Eq. 3 can be simplified to the following equation by linearizing the process model at a fixed point, $e = 0$:

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

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 the estimated states. 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 a feedback control law $u = \Phi(\hat{x}) \in U$ based on estimated states $\hat{x}$ that can render the origin of the 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 in an open neighborhood D around the origin:

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

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

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

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 Ω_ρ as a level set of Lyapunov function in the region D where the time-derivative of V is rendered negative under the controller $\Phi(\hat{x}) \in U$ $\Omega_\rho := \{x \in D \mid V(x) \leq \rho\}$, 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 $u \in U$:

$$|F(x,u)| \leq M \tag{6a}$$
$$|F(x,u) - F(x',u)| \leq L_x|x - x'| \tag{6b}$$
$$\left|\frac{\partial V(x)}{\partial x}F(x,u) - \frac{\partial V(x')}{\partial x}F(x',u)\right| \leq L_x^{'}|x - x'| \tag{6c}$$

3. RNN-BASED STATE OBSERVER

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 $\bar{x} = [\bar{x}_1, ..., \bar{x}_n]$ is the RNN state vector, and $u = [u_1, ..., u_m]$ is the manipulated input vector. $y = [y_1, ..., y_n, y_{n+1}, ..., y_{m+n}] = [\sigma(\bar{x}_1), ..., \sigma(\bar{x}_n), u_1, ..., u_m] \in \mathbf{R}^{n+m}$ is a vector of both $\bar{x}$ and u, where $\sigma(\cdot)$ is the nonlinear activation function. A is a diagonal coefficient matrix and $\Theta = [\theta_1, ..., \theta_n] \in \mathbf{R}^{(m+n)\times n}$ is a matrix with neural network weights to be optimized. The structure of unfolded and folded RNNs are shown in Fig. 1.

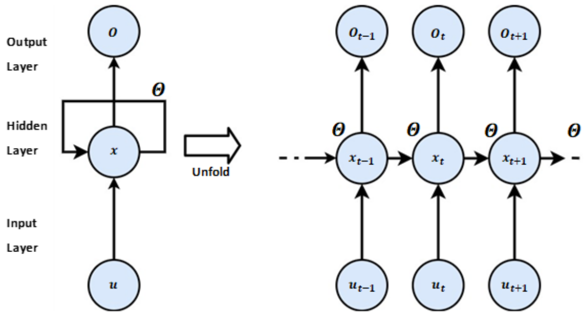

Fig. 1. Structure of recurrent neural network.

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., with a sufficiently high model accuracy) during training such that it can well represent process dynamics in the operating region we considered.

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 $h_c \times K(y - h(\bar{x}))$. 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.

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. As a result, the following condition $|\hat{x}_{rnn} - \hat{x}| \leq \epsilon$ holds for all states in the operating region, where $\epsilon > 0$ is the upper bound for the error between the state estimate $\hat{x}_{rnn}$ provided by RNN-based estimator and $\hat{x}$ from Eq. 2. The proof is omitted here due to space limitation.

Remark 1. 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. 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. (2020a) for a detailed development of co-teaching and dropout methods.

4. 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:

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), u(t))dt \tag{9a}$$
$$\text{s.t.}\quad \dot{\tilde{x}}(t) = F_{rnn}(\tilde{x}(t), u(t)) \tag{9b}$$
$$u(t) \in U,\ \forall\ t \in [t_k, t_{k+N}) \tag{9c}$$
$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{9d}$$
$$\dot{V}(\hat{x}(t_k), u) \leq \dot{V}(\hat{x}(t_k), \Phi(\hat{x}(t_k))),\quad \text{if } \hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{nn}} \tag{9e}$$
$$V(\tilde{x}(t)) \leq \rho_{nn},\ \forall\ t \in [t_k, t_{k+N}),\ \text{if } \hat{x}(t_k) \in \Omega_{\rho_{nn}} \tag{9f}$$

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. 9, Eq. 9a 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. 9b is the RNN model of Eq. 7 for predicting state evolution given control actions and an initial state. Eq. 9c is the input constraint. Eq. 9d defines the initial condition $\tilde{x}(t_k)$ of Eq. 9b, 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. 9b, and also in the constraints of Eq. 9e. If $\hat{x}(t_k) \in \Omega_\rho \backslash \Omega_{\rho_{nn}}$, the constraint of Eq. 9e 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 $\Omega_{\rho_{nn}}$ around the origin $\Omega_{\rho_{nn}}$, then the constraint of Eq. 9f requires the states to remain inside $\Omega_{\rho_{nn}}$ for the entire prediction horizon.

Closed-loop stability is guaranteed for the nonlinear system of Eq. 1 under the LMPC of Eq. 9 using state estimates from RNN-based estimator. Since the RNN is a well-conditioned model with sufficiently high model accuracy, the state estimation provided by the RNN-based estimator is of high accuracy, and converges to the true state value in a short time. Given that the error between the state estimate $\hat{x}_{rnn}$ provided by RNN-based estimator and the one $\hat{x}$ from Eq. 2 is bounded for all times, the stabilizing controller $u = \Phi(\hat{x})$ designed based on the state estimates from Eq. 2 also guarantees closed-loop stability of MPC (in sample-and-hold fashion) that uses the state estimates from RNN-based estimator.

5. APPLICATION TO A CHEMICAL REACTOR EXAMPLE

In this section, a nonlinear chemical process is used to illustrate the application of the proposed machine learning-based estimator 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:

$$\frac{dC_A}{dt} = \frac{1}{\tau}(C_{A0} - C_A) - r_A + r_B \tag{10a}$$

$$\frac{dC_B}{dt} = \frac{-1}{\tau}C_B + r_A - r_B \tag{10b}$$

$$\frac{dT}{dt} = \frac{1}{\tau}(T_0 - T) + \frac{-\Delta H}{\rho C_p}(r_A - r_B) + \frac{Q}{\rho C_p V} \tag{10c}$$

$$r_A = k_A e^{\frac{-E_A}{RT}} C_A \tag{10d}$$

$$r_B = k_B e^{\frac{-E_B}{RT}} C_B \tag{10e}$$

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. 10. The optimal steady state values and process parameter values are listed in Table 1.

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

$T_o = 400\ K$	$T_s = 426.743\ K$
$k_A = 5000\ /s$	$E_A = 1 \times 10^4\ cal/mol$
$k_B = 10^6\ /s$	$E_B = 1.5 \times 10^4\ cal/mol$
$R = 1.987\ cal/(mol\ K)$	$\Delta H = -5000\ cal/mol$
$\rho = 1\ kg/L$	$C_p = 1000\ cal/(kg\ K)$
$C_{A0} = 1\ mol/L$	$V = 100\ L$
$C_{A_s} = 0.4977\ mol/L$	$\tau = 60\ s$
$C_{B_s} = 0.5023\ mol/L$	$Q_s = 40386\ cal/s$

5.1 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 RNN-based state estimator. Based on the measurement y of the state variable x_3, the RNN-based observer first utilizes the RNN 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.

The nonlinear optimization problem of LMPC is solved using the IPOPT software package (Wächter and Biegler (2006)), and its python version, PyIpopt, with the sampling period $\Delta = 10s$. The objective function of LMPC is of the form: $L(x,u) = x^T Q x + u^T R u$, where $Q = diag[5 \times 10^4 \;\; 5 \times 10^4 \;\; 1]$, and $R = [10^{-7}]$. The observer gains used in this work are $K = [0.5, 0.0005, 0.0005]$. The Lyapunov functions is given by $V(x) = x^T \; P \; x$, with the following positive definite P matrix:

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

5.2 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. 10 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 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 unit 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.

5.3 Closed-loop Simulation Results

Closed-loop simulation study is carried out to demonstrate the performance of the proposed estimation approach in the CSTR of Eq. 10. 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. 2-5.

It can be seen from these figures that starting from different initial conditions and different initial estimates, the closed-loop states are stabilized at the steady-state under LMPC using RNN-base state estimator. Specifically, in Fig. 2 and Fig. 3, we consider two initial estimates that are very close to the true state values. 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 the steady-state smoothly. In Fig. 4 and Fig. 5, 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 Fig. 2 and Fig. 3. In all cases, closed-loop stability is achieved for the system under LMPC.

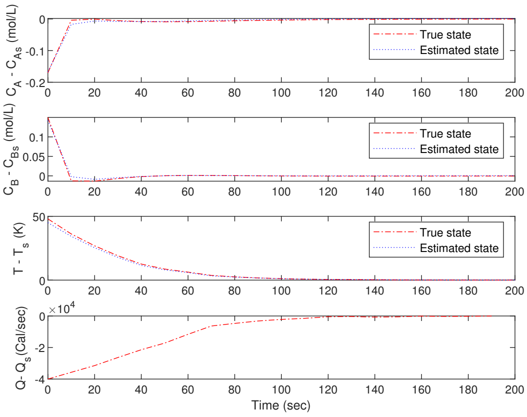

Fig. 2. True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC with the initial condition IC_1.

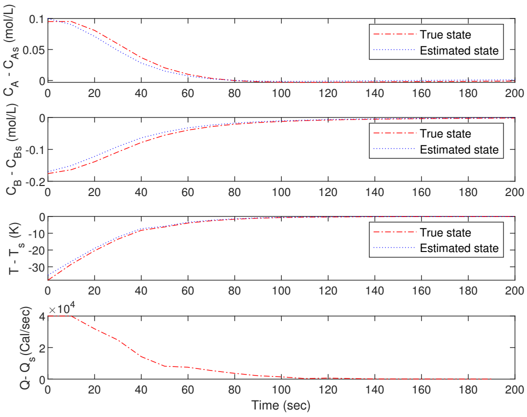

Fig. 3. True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC with the initial condition IC_2.

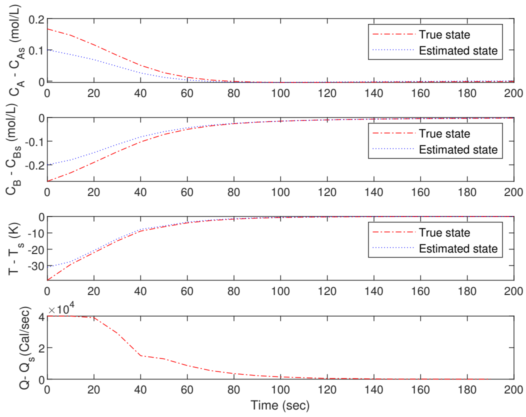

Fig. 4. True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC with the initial condition IC_3.

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-

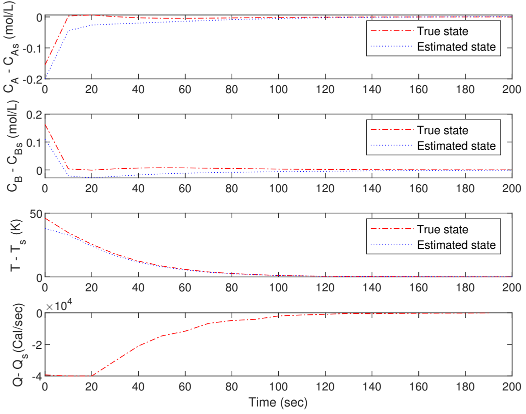

Fig. 5. True state (red line) and estimated state (blue line) trajectories for the closed-loop CSTR under LMPC with the initial condition IC_4.

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-loop simulation results as shown in Figs. 2-5. 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.

Table 2. Estimation mean squared error of the closed-loop CSTR under LMPC

Simulation No.	*MSE of* x_1	*MSE of* x_2
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}

6. CONCLUSION

In this work, we proposed a machine-learning-based state estimation approach 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. From closed-loop simulations, it was demonstrated that RNN-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 RNN-based estimator.

REFERENCES

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

Ali, J.M., Hoang, N.H., Hussain, M.A., and Dochain, D. (2015). Review and classification of recent observers applied in chemical process systems. *Computers & Chemical Engineering*, 76, 27–41.

Dochain, D. (2003). State and parameter estimation in chemical and biochemical processes: a tutorial. *Journal of Process Control*, 13, 801–818.

Kurtz, M.J. and Henson, M.A. (1998). State and disturbance estimation for nonlinear systems affine in the unmeasured variables. *Computers & Chemical Engineering*, 22, 1441–1459.

Lin, Y. and Sontag, E.D. (1991). A universal formula for stabilization with bounded controls. *Systems & Control Letters*, 16, 393–397.

McKenna, T., Othman, S., Fevotte, G., Santos, A., and Hammouri, H. (2000). An integrated approach to polymer reaction engineering: a review of calorimetry and state estimation. *Polymer Reaction Engineering*, 8, 1–38.

Mesbah, A., Huesman, A.E., Kramer, H.J., and Van den Hof, P.M. (2011). A comparison of nonlinear observers for output feedback model-based control of seeded batch crystallization processes. *Journal of Process Control*, 21, 652–666.

Patwardhan, S.C., Narasimhan, S., Jagadeesan, P., Gopaluni, B., and Shah, S.L. (2012). Nonlinear bayesian state estimation: A review of recent developments. *Control Engineering Practice*, 20, 933–953.

Porru, G., Aragonese, C., Baratti, R., and Servida, A. (2000). Monitoring of a co oxidation reactor through a grey model-based EKF observer. *Chemical Engineering Science*, 55, 331–338.

Radke, A. and Gao, Z. (2006). A survey of state and disturbance observers for practitioners. In *Proceedings of the American Control Conference*, 5183–5188. Minneapolis, Minnesota.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106, 25–57.

Wu, Z., Rincon, D., Luo, J., and Christofides, P.D. (2020a). Machine learning modeling and predictive control of nonlinear processes using noisy data. *AIChE Journal*, 67, e17164.

Wu, Z., Tran, A., Rincon, D., and Christofides, P.D. (2019a). Machine learning-based predictive control of nonlinear processes. part I: Theory. *AIChE Journal*, 65, e16729.

Wu, Z., Tran, A., Rincon, D., and Christofides, P.D. (2019b). Machine learning-based predictive control of nonlinear processes. part II: Computational implementation. *AIChE Journal*, 65, e16734.

Wu, Z., Rincon, D., and Christofides, P.D. (2020b). Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes. *Journal of Process Control*, 89, 74–84.

Zambare, N., Soroush, M., and Grady, M.C. (2002). Real-time multirate state estimation in a pilot-scale polymerization reactor. *AIChE journal*, 48, 1022–1033.

Zeitz, M. (1987). The extended luenberger observer for nonlinear systems. *Systems & Control Letters*, 9, 149–156.

Zhang, Z., Wu, Z., Rincon, D., and Christofides, P.D. (2019). Real-time optimization and control of nonlinear processes using machine learning. *Mathematics*, 7, 890.