

Co-Teaching Approach to Machine Learning-based Predictive Control of Nonlinear Processes ^{*}

Zhe Wu^{*} Junwei Luo^{*} David Rincon^{*}
Panagiotis D. Christofides^{**}

^{*} *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA (email: wuzhe@g.ucla.edu)*

^{**} *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 modeling of chemical processes using noisy data is practically a challenging task due to the occurrence of overfitting during learning. In this work, we propose a co-teaching learning algorithm that develops Long short-term memory (LSTM) networks to capture the ground truth (i.e., underlying process dynamics) from noisy data. We consider an industrial chemical reactor example and use Aspen Plus Dynamics to generate process operational data that is corrupted by sensor noise generated by industrial noisy measurements. An LSTM model is developed using the co-teaching method with additional noise-free data generated from simulations of the reactor first-principles model. Through open-loop and closed-loop simulations, we demonstrate that compared to the LSTM model developed from the standard training process, the co-teaching LSTM model is more accurate in predicting process dynamics, and therefore, achieves better closed-loop performance under model predictive control.

Keywords: Machine learning; Long short-term memory; Noisy data; Model predictive control; Nonlinear systems; Chemical processes

1. INTRODUCTION

Machine learning has attracted an increasing level of attention in classical engineering fields in recent years due to its ability of analyzing big data from industrial processes. Machine learning techniques such as neural networks have been successfully applied in process modeling, process monitoring, and fault detection, which fall into the categories of regression and classification problems. Among many types of neural networks, recurrent neural network (RNN), and long short-term memory (LSTM) networks become popular for modeling dynamic systems from time-series data, and have been incorporated in model predictive control (MPC) to predict evolution of process states when process first-principles models are unavailable. While many research works have studied neural network modeling of chemical processes using noise-free data, learning using noisy data is practically challenging due to the high capacity of neural network to fit noisy data (i.e., overfitting). Considering that the sensor measurements in chemical plants are commonly affected by noise and faults in real-time operation, machine learning modeling of chemical processes using industrial noisy data remains an important research topic.

One way to handle noisy measurements in linear systems is Kalman filter (Patwardhan et al. (2012)). Additionally, many other methods such as moving horizon estimation and unscented Kalman filter have been proposed to deal with data noise (Patwardhan et al. (2012)). In the state estimation methodology, to establish a correct estimation, a model representation is generally needed and the covariance matrices need to be tuned as well (Lima and Rawlings (2011)). Recently, the effect of learning with raw vibration signals from a laboratory-scale water flow system was studied using machine learning methods (i.e., LSTM and a feed-forward deep neural network) and a linear statistical learning approach (i.e., projection to latent structure, PLS) (Shah et al. (2020)). From their findings, it is shown poor performance from both machine learning methods and PLS when using raw vibration data and that further treatment is needed on the data for better model prediction. When exposing machine learning models to Gaussian noise, Ref Yeo (2019) has shown that machine learning models can efficiently learn the true process dynamics due to the dominant role of the internal states during the prediction step. Regarding the integration of machine learning models in model predictive controllers, the noiseless situation has been explored in recent works (Wu et al. (2019b,c); Hassanpour et al. (2020)). However, at this point, machine learning modeling of nonlinear processes using noisy data has not been addressed yet.

^{*} Financial support from the National Science Foundation and the Department of Energy is gratefully acknowledged.

Therefore, to handle industrial data noise following a non-Gaussian distribution, co-teaching method that was utilized in Han et al. (2018) to solve image classification problems with misclassified labels by training two machine learning models simultaneously is adapted in this work to solve process modeling problems using noisy data.

In this work, we consider a chemical reactor example simulated in Aspen Plus Dynamics, with noisy data generated from Aspen dynamic simulations. To implement the co-teaching method, the reactor first-principles model is first developed to generate noise-free data. Subsequently, LSTM models are trained using both noisy and noise-free data under co-teaching framework, and incorporated in the Lyapunov-based model predictive controller that optimizes process performance while maintaining system stability. Finally, we compare the co-teaching LSTM model with the LSTM model trained using the standard learning algorithm and demonstrate its superiority in both open-loop and closed-loop operations.

2. PRELIMINARIES

2.1 Notation

The Euclidean norm of a vector is denoted by the operator $|\cdot|$ and the weighted Euclidean norm of a vector is denoted by the operator $|\cdot|_Q$ where Q is a positive definite matrix. x^T denotes the transpose of x . The notation $L_f V(x)$ denotes the standard Lie derivative $L_f V(x) := \frac{\partial V(x)}{\partial x} f(x)$. Set subtraction is denoted by " \setminus ", i.e., $A \setminus B := \{x \in \mathbf{R}^n \mid x \in A, x \notin B\}$.

2.2 Class of Systems

We consider the class of continuous-time nonlinear systems described by the following system of first-order nonlinear ordinary differential equations:

$$\begin{aligned} \dot{x} &= F(x, u) := f(x) + g(x)u, \quad x(t_0) = x_0 \\ y &= x + w \end{aligned} \quad (1)$$

where $x \in \mathbf{R}^n$ is the state vector, $u \in \mathbf{R}^m$ is the manipulated input vector, $y \in \mathbf{R}^n$ is the vector of state measurements that are sampled continuously, and $w \in \mathbf{R}^n$ is the noise vector. The input vector is constrained by $u \in U := \{u_i^{\min} \leq u_i \leq u_i^{\max}, i = 1, \dots, m\} \subset \mathbf{R}^m$. $f(\cdot)$ and $g(\cdot)$ are sufficiently smooth vector and matrix functions of dimensions $n \times 1$ and $n \times m$, respectively with $f(0)$ assumed to be zero such that the origin is a steady-state of the nominal (i.e., $w(t) \equiv 0$) system of Eq. 1 (i.e., $(x_s^*, u_s^*) = (0, 0)$, where x_s^* and u_s^* represent the steady-state state and input vectors, respectively). Throughout the manuscript, we assume that the full state measurements are continuously available at all times, and the initial time t_0 is taken to be zero ($t_0 = 0$).

2.3 Long Short Term Memory (LSTM) Model

Long short-term memory (LSTM) networks are a type of recurrent neural network capable of modeling long-term dependencies in sequence prediction problems due to the design of three gates, i.e., the input gate, the forget gate, and the output gate, in the network structure. A schematic

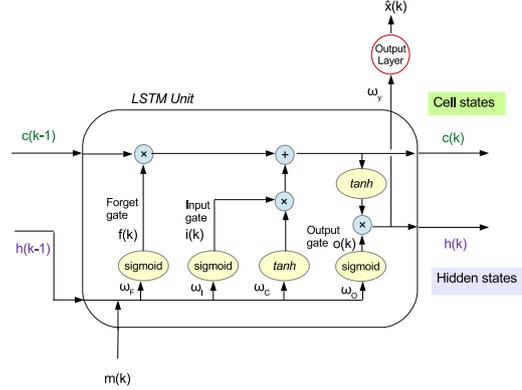


Fig. 1. Schematic of LSTM units (Chen et al. (2020)).

of LSTM network structure is shown in Fig. 1. In this work, the LSTM model is developed to predict the states of Eq. 1 given the control actions and the past noisy state measurements. Specifically, given the input sequence $m(k)$, $k = 1, \dots, T$, where T is the number of measured states of the sampled-data system of Eq. 1, the following equations are used to calculate the predicted output sequence $\hat{x}(k)$:

$$i(k) = \sigma(\omega_i^m m(k) + \omega_i^h h(k-1) + b_i) \quad (2a)$$

$$f(k) = \sigma(\omega_f^m m(k) + \omega_f^h h(k-1) + b_f) \quad (2b)$$

$$c(k) = i(k) \tanh(\omega_c^m m(k) + \omega_c^h h(k-1) + b_c) + f(k)c(k-1) \quad (2c)$$

$$o(k) = \sigma(\omega_o^m m(k) + \omega_o^h h(k-1) + b_o) \quad (2d)$$

$$h(k) = o(k) \tanh(c(k)) \quad (2e)$$

$$\hat{x}(k) = \omega_y h(k) + b_y \quad (2f)$$

where $m(k)$, $c(k)$, $h(k)$, $i(k)$, $f(k)$, and $o(k)$ are the input sequence, the cell state, the internal state, the outputs from the input gate, the forget gate, and the output gate, respectively. $\hat{x} \in \mathbf{R}^{n \times T}$ represent the LSTM network output sequences. The weight matrices for the LSTM input vector m , and the hidden state vector in the input gate are represented by ω_i^m and ω_i^h , respectively. Similarly, the weight matrices for the input vector m and hidden state vector h in calculating the cell state c , the forget gate f , and the output gate o are represented by ω_c^m , ω_c^h , ω_f^m , ω_f^h , ω_o^m , ω_o^h , respectively, with b_i , b_f , b_o , b_c representing the bias terms. Finally, the LSTM predicted state is calculated using Eq. 2f where ω_y and b_y denote the weight matrix and bias vector for the output, respectively. Since the LSTM model uses control actions and past state measurements to predict future states, the input sequence $m \in \mathbf{R}^{(n+m) \times T}$ contains the manipulated inputs $u \in \mathbf{R}^m$ and the past measured states $x \in \mathbf{R}^n$ within a certain period of time (i.e., T). The LSTM model uses the sigmoid activation function $\sigma(\cdot)$ and the hyperbolic tangent function $\tanh(\cdot)$ as the nonlinear activation functions. Additionally, as LSTM networks are a type of recurrent neural network, we can also present the LSTM model in a continuous-time nonlinear system as follows:

$$\dot{\hat{x}} = F_{nn}(\hat{x}, u) := A\hat{x} + \Theta^T z \quad (3)$$

where $\hat{x} \in \mathbf{R}^n$ is the LSTM state vector, $u \in \mathbf{R}^m$ is the manipulated input vector, and $z = [z_1 \dots z_{n+m+1}]^T = [\sigma(\hat{x}_1) \dots \sigma(\hat{x}_n) \quad u_1 \dots u_m \quad 1]^T \in \mathbf{R}^{n+m+1}$ is a vector of both the network states \hat{x} and the inputs u . $\sigma(\cdot)$ represents nonlinear activation functions in each LSTM unit, and "1"

represents the bias term. The diagonal matrix $A \in \mathbf{R}^{n \times n}$ and the matrix $\Theta \in \mathbf{R}^{(n+m+1) \times n}$ consist of the LSTM weights that will be optimized.

Training data is generated from extensive open-loop simulations of the nonlinear system of Eq. 1 under various initial conditions and control actions. The system inputs 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, and the explicit Euler method is utilized with a sufficiently small integration time step $h_c < \Delta$ to integrate the continuous-time nonlinear system of Eq. 1 in simulations. Then, the LSTM model can be trained following the learning process as discussed in Wu et al. (2019b).

Remark 1. It should be mentioned that in this work, the LSTM model is trained using noisy data (i.e., the state measurements are corrupted by industrial data noise), which makes it challenging to obtain a well-conditioned LSTM model that can capture the ground truth (i.e., the underlying process dynamics of Eq. 1) using standard learning algorithm. Therefore, to handle the noisy training data, we propose a co-teaching method that improves model prediction accuracy by taking advantage of noise-free data generated from computer simulations.

2.4 Model Predictive Control Using LSTM models

The LSTM model is incorporated in Lyapunov-based model predictive controller (LMPC) to provide state predictions in solving the MPC optimization problem. The formulation of LSTM-based MPC is presented as follows:

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), u(t)) dt \quad (4a)$$

$$\text{s.t. } \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), u(t)) \quad (4b)$$

$$u(t) \in U, \forall t \in [t_k, t_{k+N}) \quad (4c)$$

$$\dot{\hat{V}}(x(t_k), u) \leq \dot{\hat{V}}(x(t_k), \Phi_{nn}(x(t_k))), \quad (4d)$$

$$\text{if } x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_{nn}} \quad (4e)$$

where \tilde{x} , N and $S(\Delta)$ are the predicted state trajectory, the number of sampling periods in the prediction horizon, and the set of piecewise constant functions with period Δ . $\dot{\hat{V}}(x, u)$ in Eq. 4d denotes the time-derivative of \hat{V} , i.e., $\frac{\partial \hat{V}(x)}{\partial x}(F_{nn}(x, u))$. The LMPC is implemented in a receding horizon manner, where the first control action $u^*(t_k)$ in the optimal input sequence $u^*(t)$, $\forall t \in [t_k, t_{k+N})$ is applied to the system for the next sampling period. Specifically, the LMPC minimizes the time-integral of the cost function $L(\tilde{x}(t), u(t))$ that achieves its minimum value at the steady-state $(x_s^*, u_s^*) = (0, 0)$ accounting for the constraints of Eqs. 4b-4e. The control objective of LMPC is to maintain the closed-loop state in the stability region $\Omega_{\hat{\rho}}$ for all times, and ultimately bound the state in the target region $\Omega_{\rho_{nn}}$, which is a small level set of V around the origin. $\Phi_{nn}(x)$ in Eq. 4d is the a pre-determined control law that renders the origin of the LSTM system of Eq. 3 exponentially stable. When a well-conditioned LSTM model with a sufficiently high model accuracy can be obtained using noise-free training data, the LMPC of

```

for  $i = 0$  to  $I_{max}$  do
  Select a mini-batch  $D_m$  from  $D$ 
  Obtain the small-loss data sequences from model A:
   $D_A = \{x \in D_m \mid loss(A, x) \leq loss_T\}$ 
  Obtain the small-loss data sequences from model B:
   $D_B = \{x \in D_m \mid loss(B, x) \leq loss_T\}$ 
  Update the weight matrix of model A:  $\mathbf{W}_A = \mathbf{W}_A - \eta \nabla loss(A, D_B)$ 
  Update the weight matrix of model B:  $\mathbf{W}_B = \mathbf{W}_B - \eta \nabla loss(B, D_A)$ 
end

```

Algorithm 1. Co-teaching Algorithm

Eq. 4 guarantees closed-loop stability of the nonlinear system of Eq. 1. Theoretical results on closed-loop stability can be found in Wu et al. (2019b).

3. CO-TEACHING METHOD

Co-teaching method was originally proposed to improve model accuracy in image classification problems, for which the dataset is corrupted by noise (Han et al. (2018)). Specifically, data noise in classification problems could cause mislabeled data (for example, an object “A” is mislabeled as object “B”), while in regression problems, data noise could cause a deviation from its ground truth value. In either case, it is challenging for machine learning model to achieve a desired model accuracy with a noisy dataset following the standard learning algorithm. Therefore, co-teaching method provides an alternative way to train machine learning models under noisy labels by taking advantage of noise-free data and training two models at the same time (Han et al. (2018); Yang et al. (2020)). The intuition of co-teaching stems from the observations that neural networks will use a simple pattern to fit training data at the early stage of training process (Han et al. (2018)). As a result, when assessing loss function value under a simple pattern that approximates the relationship between neural network inputs and outputs, the noisy data generally has a large loss function value, while noise-free data has a small value.

Fig. 2 shows two types of co-teaching structures (i.e., symmetric and asymmetric frameworks) that train two networks: A and B , simultaneously. The symmetric co-teaching training method is implemented following Algorithm 1, which is stated as follows: 1) at each training epoch, a mini-batch D_m is selected from the original mixed dataset D . Then, each model checks its data sequences (i.e., each pair of data labeled as input and output), and generates a small dataset (i.e., D_A and D_B) with all the data that has a low loss function value (e.g., $loss(A, x) \leq loss_T$), where $loss_T$ is the threshold for identifying small-loss data sequences; 2) this new small dataset is then sent to the peer network, and the neural network weights \mathbf{W}_A , \mathbf{W}_B are updated with a learning rate η ; 3) finally, the training is resumed, and the above process is repeated until the end of training epochs I_{max} . The asymmetric co-teaching method is implemented in a similar way to train two models simultaneously. However, under asymmetric co-teaching framework, noise-free data is used by model A only, and noisy data is used by model B only. At each training epoch, model A injects a subset of noise-free data

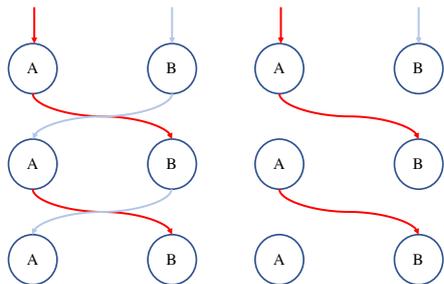


Fig. 2. The symmetric (left) and asymmetric (right) co-teaching frameworks training two networks (A and B) simultaneously.

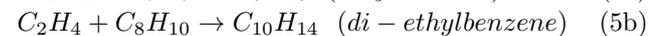
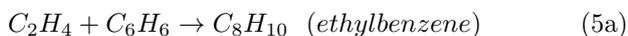
sequences into model B . Note that the information flows in one direction in asymmetric co-teaching framework, i.e., from model A to model B only.

Additionally, when using co-teaching method to solve the regression problem of LSTM modeling, the neural network structure needs to be carefully chosen. For example, the number of units in each network plays a role in learning the underlying process dynamics from a mixed dataset of both noisy and noise-free data. If a deep neural network with a large number of layers and neurons is used, the neural network may well fit the noisy data at early stage (i.e., over-fitting) before effectively learning the ground truth from noise-free data. Additionally, the mixed dataset should be constructed with an appropriate ratio of noise-free data to noisy data. If noise-free data is insufficient, the neural networks will not be able to learn the ground truth, and may overfit the noisy data as training evolves.

In the following sections, we use a chemical process example simulated in Aspen Plus Dynamics to illustrate the application of co-teaching LSTM modeling approach. Specifically, we will discuss the following steps in this case study: 1) data collection using Aspen simulation and first-principles solutions, 2) LSTM training process, and 3) development of LSTM-based MPC that drives reactor temperature to its desired set-point. Through open-loop and closed-loop simulations, we demonstrate that the proposed LSTM model using co-teaching method outperforms the standard LSTM model in terms of more accurate predictions and better control performance.

3.1 Development of Aspen Plus Reactor Model

We consider an irreversible, second-order, exothermic reaction using Ethylene(A) and Benzene(B) to produce Ethyl benzene (EB) in a well-mixed, non-isothermal continuous stirred tank reactors (CSTR) Kamal and Malah (2017). The CSTR reactor is fed with two Hexane solutions in the feeding flow F_1 and F_2 . The two flows have the same inlet temperature T_0 , but different volumetric flowrate $F_{vj,in}$, $j = 1, 2$, where $j = 1, 2$ denotes the feeding flow F_1 and F_2 . The reactants A and B are contained in each feeding flow separately with inlet molar concentration C_{A0} and C_{B0} . The reactions taking place in the CSTR are:



In this study, the reactor model is developed in Aspen Plus and Aspen Plus Dynamics V11. The model is constructed and the steady-state simulations are first performed in Aspen Plus. Then, a dynamic simulation of the reactor process is carried out in Aspen Plus Dynamics to analyze its real-time performance. In Aspen Plus, a main flow sheet is designed with three valves and one CSTR as shown in Fig. 3. The valves play a role as a connector of fluid flow

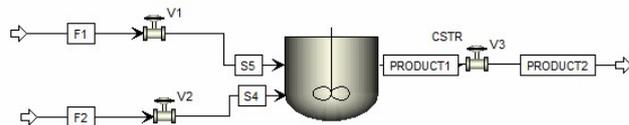


Fig. 3. Aspen flow sheet of steady-state model.

and parts by defining the pressure drop in the specific location, which is critical for generating a logical dynamic model. Without reasonable pressure drop in the process provided, Aspen Plus Dynamics can not identify the source making the fluid flow through the system and may result in failure of dynamic simulation. In this model, the pressure drop at V_1 and V_2 are both 5 bar, and the pressure drop at V_3 is 2 bar.

Hexane is chosen as the solvent in the feeding flow F_1 and F_2 to ensure that the flow is in the liquid phase under the inlet temperature. Therefore, with a constant inlet volumetric flow rate, the amount of feeding reactants can be manipulated by adjusting the feeding concentration. Process parameter values used in the Aspen model are listed in Table 1, where C_A , C_B , ρ_L , V , and T are the concentration of ethylene, the concentration of benzene, mass density, volume and temperature of the reacting liquid in the CSTR, respectively. C_p is the mass heat capacity of the liquid mixture and is assumed to be constant. C_{As} and C_{Bs} are the steady-state concentration of reactants A and B , and C_{A0} , C_{B0} are the inlet concentration of A and B .

Table 1. Parameter values of Aspen model

| | |
|--|--|
| $T_0 = 350.0 \text{ K}$ | $T_s = 322.2 \text{ K}$ |
| $F_{v1,in} = 50.0 \text{ m}^3/\text{hr}$ | $F_{v2,in} = 23.6 \text{ m}^3/\text{hr}$ |
| $C_{As} = 1.5454 \text{ kmol/m}^3$ | $C_{Bs} = 4.2714 \text{ kmol/m}^3$ |
| $C_{A0} = 4 \text{ kmol/m}^3$ | $C_{B0} = 5 \text{ kmol/m}^3$ |
| $Q_s = -695.1 \text{ kJ/s}$ | $C_p = 2.41 \text{ kJ/kg K}$ |
| $V = 60 \text{ m}^3/\text{s}$ | $\rho_L = 683.7 \text{ kg/m}^3$ |

A liquid-only CSTR equipped with a heating jacket that supplies/removes heat at a rate Q , is considered to carry out three reactions. The initial temperature and pressure of the CSTR are set to be 400 K and 15 bar which can be automatically adjusted by the steady-state simulation in Aspen. After incorporating the reactions of Eq. 5 in the CSTR, steady-state simulation is performed for analysis of plant behavior.

Before exporting the steady-state model to Aspen Plus Dynamics, reactor geometry and thermodynamic parameters are required to be defined in the dynamic mode of Aspen. In this study, the CSTR geometric is vertical, flat,

and 10 meters in length. Its thermodynamic parameter is reported in Table 2.

Table 2. Thermodynamic parameters of CSTR

| Heat Transfer Option | <i>Dynamic</i> |
|----------------------|----------------|
| Medium Temperature | 298 K |
| Temperature Approach | 77.33 K |
| Heat Capacity | 4200 J/kg K |
| Medium Holdup | 1000 kg |

Then, we run the steady-state simulation again to ensure that the dynamic mode is set up properly. After executing the pressure check in Aspen Plus Pressure Checker, the steady-state model is exported to Aspen Plus Dynamics. Lastly, we choose the pressure as the driven type of the dynamic model.

3.2 Dynamic Model in Aspen Plus Dynamics

The flow sheet of Aspen dynamic model is shown in Fig. 4. Specifically, a direct acting level controller, where direct means the output signal increases as the input signal increases, is added to the dynamic model to regulate the liquid level at 50 percent occupation in this study. Note that the controller can be designed following the default setting before exporting the steady-state model or can be manually developed in Aspen Plus Dynamics.

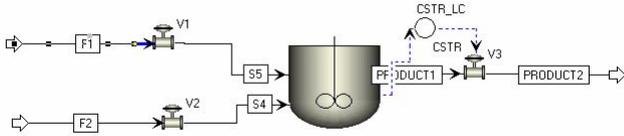


Fig. 4. Aspen flow sheet of dynamic model.

After controller configuration, we run a steady-state simulation in Aspen Plus Dynamics to obtain the steady-state for the dynamic model. The steady-state value of Q is $-695097.0 W$. Then, the heating type of CSTR is changed to constant duty to allow the outside controllers to manipulate Q during the dynamic simulation. The volumetric flow rates of F_1 and F_2 are fixed, and a steady-state simulation is performed to ensure that the dynamic model reaches the steady-state before collecting data.

Since Aspen dynamic model can be considered a high-fidelity process model for CSTR, we use Aspen dynamic simulation to generate datasets for neural network training. Industrial noise is introduced on state measurements to represent common sensor variability in chemical plants. Fig. 5 shows the normalized data noise obtained from Aspen public domain data. Specifically, open-loop simulation is carried out in Aspen Plus Dynamics using the pseudorandom input signals generated in Matlab. A local Message Passing Interface (MPI) is constructed to link Aspen with Matlab so that the Aspen dynamic model can automatically read the input signals from Matlab and apply them in the dynamic simulations.

In the open-loop simulation, the manipulated input variable Q varies within the range of $[-1.0 \times 10^6 W, -4.0 \times 10^5 W]$, and is implemented in a sample-and-hold fashion

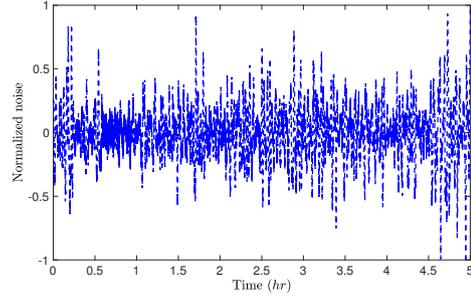


Fig. 5. Normalized industrial noise from Aspen public domain data.

with the value updated every five minutes of the simulation time. We run the open-loop simulation for 15,000 minutes (simulation time) under pseudorandom input signals of Q with the industrial noise of Fig. 5 added on the temperature measurements. All the state measurements (e.g., C_A , C_B , and T) and input value of Q are continuously recorded to build the dataset for neural network training.

3.3 First-Principles Model

Since Aspen Plus models are typically not used in controller design due to its high computational cost, to reduce the computational time of solving the process model, first-principles models can be adopted in the design of model-based controllers. Additionally, extensive computer simulations using first-principle model is one of the most efficient data generation methods in machine learning.

In this study, we take advantage of the first-principles model of CSTR to generate noise-free datasets for LSTM training using co-teaching method. Although the first-principles model may not fully capture the Aspen model dynamics under the same operating conditions, we will demonstrate that the co-teaching method using noisy data from Aspen model, and noise-free data from first-principles solutions is still able to improve prediction accuracy of LSTM model. While in practice noisy data is provided by chemical plants, and noise-free data is unavailable, the implementation of co-teaching method in this case study implies that the co-teaching LSTM modeling approach can improve prediction accuracy by using noise-free data generated from first-principles models, which broadens its application in many process modeling problems in industry. By applying mass and energy balances, the dynamic model of CSTR is described by the following nonlinear ODEs:

$$\frac{dC_A}{dt} = \frac{F_{v1,in}}{V} (C_{A0} - C_A) - r_1 - r_2 \quad (6a)$$

$$\frac{dC_B}{dt} = \frac{F_{v2,in}}{V} (C_{B0} - C_B) - r_1 - r_3 \quad (6b)$$

$$\frac{dT}{dt} = \frac{F_{v1,in} + F_{v2,in}}{V} (T_0 - T) + \frac{-\Delta H_1}{\rho_L C_p} r_1 \quad (6c)$$

$$+ \frac{-\Delta H_2}{\rho_L C_p} r_2 + \frac{-\Delta H_3}{\rho_L C_p} r_3 + \frac{Q}{\rho_L C_p V} \quad (6d)$$

$$r_1 = k_1 e^{\frac{E_1}{RT}} C_A C_B \quad (6e)$$

$$r_2 = k_2 e^{\frac{E_2}{RT}} C_A C_E B \quad (6f)$$

$$r_3 = k_3 e^{\frac{E_3}{RT}} C_B C_D E B \quad (6g)$$

where r_j , $j = 1, 2, 3$ denote the rate of each reaction in Eq. 5 based on the rate law equation, and C_{EB} , C_{DEB} represent the concentration of C_8H_{10} , and of $C_{10}H_{14}$, respectively. The kinetic parameters for reactions are given in Table. 3, where R , k_j , ΔH_j , and E_j , $j = 1, 2, 3$ represent ideal gas constant, pre-exponential constant, enthalpy of reaction, and activation energy of each reaction, respectively.

Table 3. Parameter values of the first-principles model of CSTR

| | |
|---|--|
| $k_1 = 1.528 \times 10^6 \text{ m}^3/\text{kmol s}$ | $\Delta H_1 = -1.04 \times 10^5 \text{ kJ/kmol}$ |
| $k_2 = 2.778 \times 10^5 \text{ m}^3/\text{kmol s}$ | $\Delta H_2 = -1.02 \times 10^5 \text{ kJ/kmol}$ |
| $k_3 = 0.4167 \text{ m}^3/\text{kmol s}$ | $\Delta H_3 = -5.50 \times 10^2 \text{ kJ/kmol}$ |
| $E_1 = 71160 \text{ kJ/kmol}$ | $R = 8.314 \text{ kJ/kmol K}$ |
| $E_2 = 83680 \text{ kJ/kmol}$ | $E_3 = 62760 \text{ kJ/kmol}$ |
| $V = 60 \text{ m}^3/\text{s}$ | $\rho_L = 683.7 \text{ kg/m}^3$ |
| $C_p = 2.41 \text{ kJ/kg K}$ | |

The manipulated input is the heat input rate Q represented in deviation variable form, i.e., $u^T = [Q - Q_s]$. Similarly, the process states are represented by $x^T = [C_A - C_{As} \ C_B - C_{Bs} \ T - T_s]$ where C_{As} , C_{Bs} , T_s are the steady-state values of C_A , C_B and T . By representing all the variables in deviation forms, the equilibrium of Eq. 6 is at the origin of state-space. The same pseudorandom signals of Q applied in Aspen simulations are applied to the open-loop simulation of the first-principles model of Eq. 6, where explicit Euler method is used to integrate the nonlinear ODEs with a sufficiently integration time step $h_c = 0.05 \text{ min}$. The input signals are applied in a sample-and-hold fashion with the sampling period $\Delta = 5 \text{ min}$. In open-loop simulations, process variables are measured every integration time step.

Fig. 6 compares the open-loop state profiles from Aspen simulation and first-principles solutions under the same input sequences. Although the state profiles are close to each other, small deviations in the evolution of states can be noticed between the two models, which implies the existence of a model mismatch between the Aspen model and the first-principles model of Eq. 6. Therefore, with noisy data from Aspen simulation and noise-free data from first-principles solutions, the simulation study in the next section provides an insight on the applicability of co-teaching method in handling real industrial noisy data, for which the corresponding noise-free data is generally unavailable, but can be obtained from computer simulations using first-principles or empirical models.

3.4 Co-teaching LSTM Model

To reduce the impact of measurement noise in predicting future states, the LSTM models in this work rely on the state measurements over a past period of time to make predictions. The LSTM model is developed with C_A , C_B , T , and Q as inputs to predict the temperature T in the future time. Specifically, when using LSTM models in MPC to predict future states, the LSTM input vector at the current time step $t = t_k$ consists of the state measurements of $C_A(t)$, $C_B(t)$ and $T(t)$ over past five sampling periods, i.e., $\forall t \in [t_{k-5}, t_k]$ and the heat input

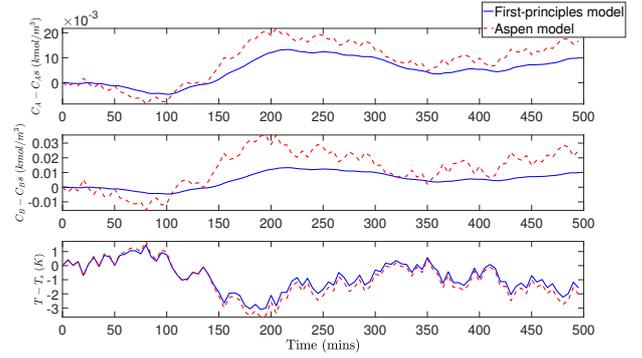


Fig. 6. State profiles ($C_A - C_{As}$, $C_B - C_{Bs}$, $T - T_s$) from open-loop simulations of Aspen model and of first-principle model, respectively, under the same input sequences of Q .

rate $Q(t)$, $\forall t \in [t_{k-4}, t_{k+1}]$ implemented in a sample-and-hold fashion. Note that the heat input rate within the last sampling period, i.e., $\forall t \in [t_k, t_{k+1}]$ is unknown at the current time step t_k as it is the variable that will be optimized by MPC to meet the control objective. The LSTM output vector at the current time step $t = t_k$ is the predicted temperature $T(t)$ over $t \in [t_{k-4}, t_{k+1}]$. Since the temperature measurements before the current time step are known, only the prediction of $T(t)$ in the last sampling period, i.e., $\forall t \in [t_k, t_{k+1}]$ will be used in MPC to solve the optimization problem.

After running Aspen dynamic simulations and open-loop simulations of the first-principles model of Eq. 6, we obtain a dataset with LSTM inputs and outputs and reshape it to the following tensor dimensions: $[2467, 500, 4]$ for inputs and $[2467, 500, 1]$ for outputs, where the first element represents the total number of data sequences, the second element represents the length of each data sequence (i.e., 500 data points correspond to five sampling periods 25 min , in which the data point is collected every integration time step $h_c = 0.05 \text{ min}$), and the last element represents the dimension of inputs and outputs, respectively (i.e., the LSTM has four inputs: C_A , C_B , T , and Q , and one output: T). Among 2467 data sequences, 494 sets of data are used for validation, and 100 sets of data are saved for testing. In the case of the co-teaching method, noiseless datasets are obtained from first-principles solutions under the same operating conditions as performed in Aspen simulations. The high-level application programming interface, Keras, is used to develop the standard and co-teaching LSTM networks under the optimization algorithm *Adam* Kingma and Ba (2014).

Remark 2. While in this study, the LSTM model is built using data only, in Wu et al. (2020), we have demonstrated that by incorporating physical knowledge of chemical processes (such as the first-principles model of Eq. 6) into neural network modeling, the model performance can be enhanced as compared to brutal-force neural network models. Therefore, in addition to the simulating noise-free data for co-teaching method, the process model of Eq. 6 can be further utilized to guide neural network structure designs following the approaches in Wu et al. (2020).

3.5 Open-loop Simulation Results

We first carry out open-loop simulation study to demonstrate the improvement of LSTM model accuracy using co-teaching method. Table 4 shows the mean squared errors (MSE) between the predicted temperature under different LSTM models, and the ground truth (i.e., actual temperature value of the nominal system) from testing dataset (the unit of MSE is K^2). The results for standard LSTMs under different datasets are shown in item 1, and those for co-teaching LSTMs are shown in item 2.

Table 4. Open-loop prediction results under industrial noise

| Methods | MSE T |
|--|---------|
| 1a) LSTM : only using noisy data | 1.8217 |
| 1b) LSTM : mixed data (noise-free data from fp) | 3.0357 |
| 1c) LSTM : mixed data (noise-free data from Aspen) | 1.5386 |
| 2a) Co-teaching LSTM (noise-free data from fp) | 0.8596 |
| 2b) Co-teaching LSTM (noise-free data Aspen) | 0.7140 |

Note that all the LSTM models in Table 4 are developed with the same structure in terms of the number of neurons, layers, epochs, and the type of activation functions and of optimization algorithms. We first consider three types of datasets for standard LSTM models, and the results are reported in item 1 of Table 4. Specifically, in item 1a, the LSTM model is trained following the standard training process with noisy data only (i.e., noisy data from Aspen simulations in Section 3.2); in item 1b, the LSTM model is trained using a mixed dataset consisting of both noisy data from Aspen simulations and noise-free data from simulations of the first-principles model in Section 3.3 (“fp” in Table 4 represents the first-principles model); in item 1c, we consider a scenario where noise-free data is also available from Aspen simulations, thereby the LSTM model is trained using both noisy and noise-free data from Aspen simulations. However, it should be noted that the last scenario is considered only for comparison purposes since the noise-free data from chemical plants (here the Aspen model can be considered as a real chemical process) are generally unavailable. It can be seen from Table 4 that introducing noise-free data into brute force learning of LSTMs (i.e., standard LSTM models) may or may not improve their prediction accuracy. Specifically, when noise-free data from the same process (i.e., from Aspen model) is provided with noisy data, standard LSTM achieves a lower MSE in item 1c than the standard LSTM using noisy data only in item 1a; however, the standard LSTM using a mixed dataset with noise-free data from first-principles model has a larger MSE due to the mismatch between the Aspen model (i.e., source of noisy data) and the first-principles model (i.e., source of noise-free data). This mismatch, if not handled appropriately, may misguide LSTM training and leads to worse prediction performance.

Subsequently, we train LSTM models using co-teaching method with the same two types of mixed datasets (i.e., the noise-free data from Aspen model, and from first-principles model, respectively). The co-teaching LSTM training is initially equipped with a noisy dataset, and as the training evolves, noise-free data sequences are introduced into the learning process as discussed in the

co-teaching algorithm. As shown in Table 4, the two co-teaching LSTM models have lower MSEs than the corresponding standard LSTM models using the same type of mixed dataset. Additionally, the co-teaching LSTM using noise-free data from Aspen simulations has the lowest MSE results among all the LSTM models in this study.

3.6 Closed-loop Simulation Results

Finally, we incorporate the LSTM models in the LMPC of Eq. 4, and carry out closed-loop simulation study to demonstrate the improved closed-loop performance under co-teaching LSTM models. The control objective of LMPC is to stabilize the reactor temperature at its steady-state T_s by manipulating the heat input rate ΔQ . The LMPC objective function of Eq. 4a is designed with the following form that has its minimum value at the steady-state:

$$L(x, u) = |x_3|_{Q_1}^2 + |u|_{Q_2}^2 \quad (7)$$

where Q_1 and Q_2 are the coefficient matrices that represent the contributions of temperature and of control actions (both are in deviation forms) in the MPC objective function. In this example, we choose $Q_1 = 1$ and $Q_2 = 5 \times 10^{-9}$. The nonlinear optimization problem of LMPC is solved using the python module of the IPOPT software package Wächter and Biegler (2006), named PyIpopt with the sampling period $\Delta = 5 \text{ min}$.

Fig. 7 and Fig. 8 show the closed-loop state profiles (with noisy measurements) under LMPC using standard LSTM and co-teaching LSTM models for two different initial conditions. Specifically, Fig. 7 shows that starting from an initial temperature $T = 340 \text{ K}$ higher than the steady-state value, both standard LSTM and co-teaching LSTM models drive the temperature to its steady-state within 100 minutes. However, in Fig. 8, with an initial temperature $T = 300 \text{ K}$ lower than the steady-state value, the standard LSTM model takes much longer time than the co-teaching LSTM model to stabilize the temperature at the steady-state.

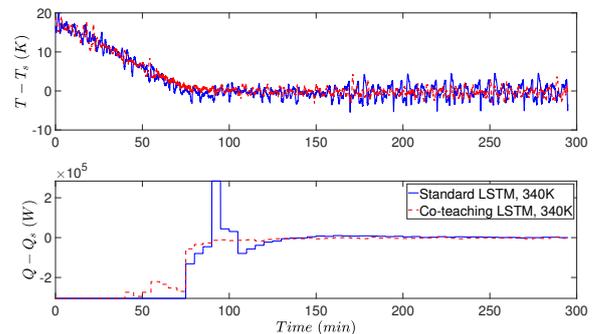


Fig. 7. Closed-loop state profile ($x_3 = T - T_s$) and manipulated input profile ($u = Q - Q_s$) for the initial condition $T = 340 \text{ K}$ under the MPC using standard LSTM, and co-teaching LSTM, respectively.

To further analyze their closed-loop performances in terms of state convergence speed and energy consumption, we use the MPC objective function as an indicator of closed-loop

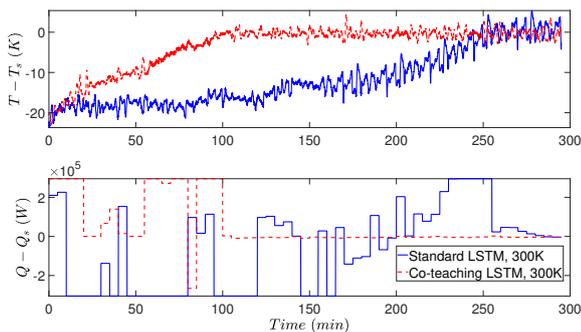


Fig. 8. Closed-loop state profile ($x_3 = T - T_s$) and manipulated input profile ($u = Q - Q_s$) for the initial condition $T = 300$ K under the MPC using standard LSTM, and co-teaching LSTM, respectively.

performance as it accounts for both state and input information. It can be seen from Eq. 7 that a lower objective function value implies a faster convergence to the steady-state and less consumption of Q during operation. Therefore, we integrate the objective function value over the closed-loop simulation period t_s , i.e., $L_s = \int_{t=0}^{t=t_s} L(x, u) dt$, for each LSTM model. For the initial condition $T = 340$ K, L_s is calculated to be 44963.07 for standard LSTM and 37843.28 for co-teaching LSTM; for the initial condition $T = 300$ K, L_s are calculated to be 120697.7 for standard LSTM and 41083.07 for co-teaching LSTM. In both cases, co-teaching LSTM model has a lower L_s value than standard LSTM model, which indicates an improvement in closed-loop performance. Additionally, we test more initial conditions of temperature within $[300, 340]$ K under LMPC. It is demonstrated that for $T_{initial} > T_s$, both standard and co-teaching LSTM models can stabilize the temperature at the steady-state within a short time, while for $T_{initial} < T_s$, co-teaching LSTM models significantly improves its dynamic response than standard LSTM (like the one in Fig. 8). For all the tested initial conditions, co-teaching LSTM model achieves a better closed-loop performance with a lower value of L_s .

Remark 3. In this work, the state profile oscillations in Figs. 7 and 8 are due to the data noise in sensor measurements. However, in the case that the state profiles vary due to varying process parameters, online learning of process models using most recent process data can be employed to update process models as time evolves. The interested reader is referred to Wu et al. (2019a) for discussion of event- and error-triggered online neural network update within real-time machine learning-based MPC.

4. CONCLUSION

In this work, we developed an LSTM modeling approach using co-teaching technique to predict underlying process dynamics (ground truth) from noisy data. A chemical reactor example was utilized to demonstrate the application of co-teaching method with noisy and noise-free data generated from Aspen simulation and first-principles solutions, respectively. Then, we simulated the reactor under LSTM-based MPC in Aspen Plus Dynamics, and demonstrated the superiority of co-teaching LSTM model over the stan-

ard LSTM model in both open-loop prediction accuracy and closed-loop control performance.

REFERENCES

- Chen, S., Wu, Z., Rincon, D., and Christofides, P.D. (2020). Machine learning-based distributed model predictive control of nonlinear processes. *AIChE Journal*, 66, e17013.
- Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, 8527–8537.
- Hassanpour, H., Corbett, B., and Mhaskar, P. (2020). Integrating dynamic neural network models with principal component analysis for adaptive model predictive control. *Chemical Engineering Research and Design*, 161, 26–37.
- Kamal, I. and Malah, A. (2017). *Aspen Plus Chemical Engineering Applications*. John Wiley& Sons, Inc.
- Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lima, F.V. and Rawlings, J.B. (2011). Nonlinear stochastic modeling to improve state estimation in process monitoring and control. *AIChE journal*, 57, 996–1007.
- 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.
- Shah, D., Wang, J., and He, Q.P. (2020). Feature engineering in big data analytics for iot-enabled smart manufacturing—comparison between deep learning and statistical learning. *Computers & Chemical Engineering*, 106970.
- 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., and Christofides, P.D. (2019a). Real-time adaptive machine-learning-based predictive control of nonlinear processes. *Industrial & Engineering Chemistry Research*, 59, 2275–2290.
- Wu, Z., Rincon, D., and Christofides, P.D. (2020). Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes. *Journal of Process Control*, 89, 74–84.
- Wu, Z., Tran, A., Rincon, D., and Christofides, P.D. (2019b). 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. (2019c). Machine learning-based predictive control of nonlinear processes. part II: Computational implementation. *AIChE Journal*, 65, e16734.
- Yang, F., Li, K., Zhong, Z., Luo, Z., Sun, X., Cheng, H., Guo, X., Huang, F., Ji, R., and Li, S. (2020). Asymmetric co-teaching for unsupervised cross-domain person re-identification. In *AAAI*, 12597–12604.
- Yeo, K. (2019). Short note on the behavior of recurrent neural network for noisy dynamical system. *arXiv preprint arXiv:1904.05158*.