2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Phased LSTM-Based MPC for Modeling and Control of Nonlinear Systems Using Asynchronous and Delayed Measurement Data

Wanlu Wu, Yujia Wang, Ming-Qing Zhang, Min-Sen Chiu, and Zhe Wu

Abstract— **In this work, we develop novel machine learning modeling and predictive control techniques for nonlinear chemical systems with asynchronous and delayed measurements in both offline and online data collection. Specifically, Phased Long Short-Term Memory (PLSTM) network is used to learn the process dynamics amidst the irregularities in the data, during the offline training process. The generalization performance of PLSTM is theoretically studied on the basis of statistical machine learning theory to better understand the capabilities of PLSTM models. The PLSTM model is employed to forecast the evolution of states for a Lyapunov-based Model Predictive Controller (LMPC) that is designed to account for data loss and delays in real-time implementation. Finally, an application to a benchmark chemical process is adopted to show the effectiveness of PLSTM modeling and predictive control methods.**

I. Introduction

Delays and data loss are common problems faced during information exchange in Networked Control Systems (NCS), which can compromise the system stability [1]. Extensive research has been carried out to devise a control technique that addresses data loss and delays in NCS. Among the various control strategies such as optimal stochastic control [2] and robust control (H_∞) ([3]), Model Predictive Control (MPC) stood out for its predictive capabilities, which actively account for time delays and data loss. Specifically, a Lyapunov-based MPC (LMPC) was proposed to handle the independent occurences of delays and data loss, by modeling them as time-varying measurement delays ([4]) and asynchronous measurements ([5]), respectively.

The MPC prediction model can originate from either theoretical knowledge or empirical data. Although first-principles models can provide valuable insights into process behavior, they are often time-consuming and expensive to construct, especially for complex nonlinear systems. With the increasing availability of industrial data, there has been a noticeable shift from the sole reliance on first-principles models to the adoption of data-driven models. Machine learning models, specifically Recurrent Neural Networks (RNNs), have demonstrated success in capturing the dynamics of nonlinear processes [6], [7].

As a result of delays and data loss, process state measurements may appear to be irregular, with missing data at certain sampling times. These irregularities pose challenges for RNNs in learning the system's dynamics. Thus, variations have been proposed for standard RNNs, capable of handling irregular data with minimal preprocessing. These include time-aware Long Short-Term Memory (LSTM) [8] and Gated Recurrent Units (GRU) with bistable cells [9]. However, these models have limited transferability to process modeling domain, as they have yet to be applied to either missing data or data with real-valued input variables [10]. Phased Long Short-Term Memory (PLSTM), a modification of the standard LSTM unit, is a promising method to process irregularly sampled data. In [11], PLSTM was shown to retain high accuracy in a frequency discrimination task to differentiate two classes of sine waves for three different sampling conditions: standard, high resolution, and asynchronous. The asynchronous sampling process is random, which resembles the asynchronous modeling of data loss [5].

In process modeling, it is necessary for the model to capture the dynamics of the target system accurately. However, given that the training dataset is finite and insufficient to cover all possible state trajectories under different operating conditions, the generalization error is a useful tool to quantify the model's ability to predict unseen data. Although many studies have attempted to evaluate the generalizability of conventional machine learning models [12], [13], [14], the generalization performance of PLSTM has not yet been studied. Motivated by the aforementioned considerations, we derive an upper bound for the generalization error of PLSTMs and propose a PLSTM-based MPC with stability guarantees for the closed-loop system. The proposed modeling and control methods are capable of handling data loss and delays in state measurements of nonlinear systems, in both offline training and online MPC implementation.

II. Preliminaries

A. Notations

For a given matrix $A \in \mathbb{R}^{m\times n}$, its Frobenius norm is denoted as $\|A\|_F$. For a given vector $b \in \mathbb{R}^d$, $\|b\|$ denotes its Euclidean norm. The superscript T is used to indicate the transpose of a vector / matrix. The term $\mathbb{R}_+$ denotes non-negative real numbers. Set subtraction is represented using "$\backslash$", i.e., $P\backslash Q := \{z \in \mathbb{R}^n \mid z \in P, z \notin Q\}$. Given a function $g : \mathbb{R}^n \to \mathbb{R}^m$, for all $a, b \in \mathbb{R}^n$, if $\|g(x)-g(y)\| \leq M\|x-y\|$ where $M > 0$, then g is M-Lipschitz. If a function $g(\cdot)$ is continuously differentiable, then g belongs to the class $\mathcal{C}^1$. Let $g : [0, g) \to [0, \infty)$ be a continuous function, g is said to be in class $\mathcal{K}$ if g is strictly increasing and $g(x) = 0$, if and only if $x = 0$. The expected value of a random variable X

Financial support from NRF-CRP 27-2021-0001 is gratefully acknowledged.

Wanlu Wu, Yujia Wang, Min-Sen Chiu, and Zhe Wu are with the Department of Chemical and Biomolecular Engineering, National University of Singapore. (e-mail: wuzhe@nus.edu.sg). Ming-Qing Zhang is with the College of Information Science & Technology, Beijing University of Chemical Technology.

is represented using the notation $\mathbb{E}[X]$. The notation $\mathbb{P}(M)$ is the probability of the event M occurring.

B. Class of Systems

Continuous-time nonlinear systems of the state-space representation below are commonly used to model chemical processes:

$$\dot{x} = F(x, u),\ x(t_0) = x_0 \tag{1}$$

The manipulated input vector is $u \in \mathbb{R}^k$. The state vector is $x \in \mathbb{R}^n$. The following constraint is imposed on the control actions $u \in U$, $U := [u_{min}, u_{max}]$, where the minimum and maximum values of the allowed inputs are denoted by u_{min} and u_{max}, respectively. t_0 is defined as the initial time, which is assumed to be $t_0 = 0$. In addition, the Lipschitz property of $F(x, u)$ assumes that there exists a positive constant L_x such that $\|F(x, u) - F(x', u)\| \leq L_x\|x - x'\|$, for all $x, x', \in O$, where O is an open set around $x = 0$, and $u \in U$. It is also assumed that the origin of Eq. (1) is a steady state and can be rendered exponentially stable under a feedback control law $u = \Phi(x) \in U$.

III. Phased LSTM

In this study, we consider a PLSTM model that is able to approximate the nonlinear system dynamics of Eq. (1), using m data samples ($\mathbf{x}_{i,t}$, $\mathbf{y}_{i,t}$, η_t) where $\mathbf{x}_{i,t} \in \mathbb{R}^{d_x}$, $\mathbf{y}_{i,t} \in \mathbb{R}^{d_y}$, $\eta_t \in \mathbb{R}$, and $i = 1, ..., m$ and $t = 1, ..., T$. The PLSTM input is represented by the pair $< \mathbf{x}_{i,t}, \eta_t >$ and the output is given by $\mathbf{y}_{i,t}$. The term η_t is a timestamp that indicates the time at which the state measurement is collected. To avoid confusion, boldface will be used for all vectors associated with the model. The update equations of PLSTM are listed below [11]:

$$\mathbf{g}_{i,t} = \sigma(W_{xg}\mathbf{x}_{i,t} + W_{hg}\mathbf{h}_{i,t-1}) \tag{2a}$$

$$\mathbf{r}_{i,t} = \sigma(W_{xr}\mathbf{x}_{i,t} + W_{hr}\mathbf{h}_{i,t-1})) \tag{2b}$$

$$\mathbf{o}_{i,t} = \sigma(W_{xo}\mathbf{x}_{i,t} + W_{ho}\mathbf{h}_{i,t-1})) \tag{2c}$$

$$\mathbf{k}_{i,t} = [\kappa_{i,t}, .., \kappa_{i,t}] \tag{2d}$$

$$\kappa_{i,t} = \begin{cases} \frac{2\phi_{i,t}}{r_{on}} & \text{if } \phi_{i,t} < \frac{1}{2}r_{on} \\ 2 - \frac{2\phi_{i,t}}{r_{on}} & \text{if } \frac{1}{2}r_{on} < \phi_{i,t} < r_{on} \\ \alpha\phi_{i,t} & \text{otherwise} \end{cases} \tag{2e}$$

$$\tilde{\mathbf{c}}_{i,t} = \tanh(W_{xc}\mathbf{x}_{i,t} + W_{hc}\mathbf{h}_{i,t-1}) \tag{2f}$$

$$\hat{\mathbf{c}}_{i,t} = \mathbf{g}_{i,t} \odot \mathbf{c}_{i,t-1} + \mathbf{r}_{i,t} \odot \tilde{\mathbf{c}}_{i,t} \tag{2g}$$

$$\mathbf{c}_{i,t} = (1 - \mathbf{k}_{i,t}) \odot \mathbf{c}_{i,t-1} + \mathbf{k}_{i,t} \odot \hat{\mathbf{c}}_{i,t} \tag{2h}$$

$$\hat{\mathbf{h}}_{i,t} = \mathbf{o}_{i,t} \odot \tanh(\hat{\mathbf{c}}_{i,t}) \tag{2i}$$

$$\mathbf{h}_{i,t} = (1 - \mathbf{k}_{i,t}) \odot \mathbf{h}_{i,t-1} + \mathbf{k}_{i,t} \odot \hat{\mathbf{h}}_{i,t} \tag{2j}$$

where $\mathbf{h}_{i,t}, \mathbf{c}_{i,} \in \mathbb{R}^{d_h}$ represent the hidden state and cell state vectors, respectively, with initial values $\mathbf{h}_{i,0} = \mathbf{c}_{i,0} = 0$. The Hadamard product is denoted by $\odot$ and Eqs. (2a) - (2c) define the three standard LSTM gate functions at time t, namely, the forget $\mathbf{g}_{i,t}$, input $\mathbf{r}_{i,t}$, and output $\mathbf{o}_{i,t}$ gates, where $\mathbf{g}_{i,t}, \mathbf{r}_{i,t}, \mathbf{o}_{i,t} \in \mathbb{R}^{d_h}$. The gates

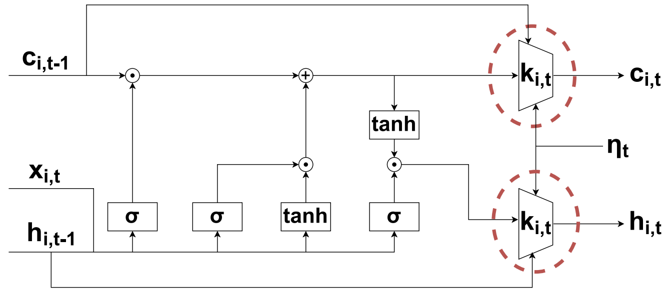

Fig. 1. Schematic of PLSTM.

use element-wise nonlinear activation functions (e.g., $\tanh(\cdot)$ and $\sigma(\cdot)$). Weight matrices $W_{xg}, W_{xr}, W_{xo}, W_{xc}, \in \mathbb{R}^{d_h \times d_x}$ and $W_{hg}, W_{hr}, W_{ho}, W_{hc} \in \mathbb{R}^{d_h \times d_h}$ are used to connect the input layer and the hidden states to the different gates, respectively. The output $\mathbf{y}_{i,t} \in \mathbb{R}^{d_y}$ is given by $\mathbf{y}_{i,t} = \sigma_y(V\mathbf{h}_{i,t})$, with element-wise activation function σ_y (usually linear unit for regression), and weight matrix $V \in \mathbb{R}^{d_y \times d_h}$.

A schematic of PLSTM is shown in Fig. 1. From Eq. (2) and Fig. 1, it can be seen that PLSTM has an additional oscillatory time gate, $\mathbf{k}_{i,t} \in \mathbb{R}^{d_h}$, as compared to a standard LSTM. This additional time gate $\mathbf{k}_{i,t}$ determines the update of the cell and hidden states by opening and closing its gates rhythmically. It should be noted that $\mathbf{k}_{i,t}$ is a constant vector with constant $\kappa_{i,t}$ that depends on three learnable parameters: τ, r_{on}, and s. τ denotes the period of oscillation, r_{on} indicates the ratio of the open phase to the full period, and s represents the phase shift of the oscillation with respect to each PLSTM cell. A linearized formulation of $\kappa_{i,t}$, is given in Eq. (2e), where $\phi_{i,t} = \frac{(\eta_t - s) \mod \tau}{\tau}$. The term $\phi_{i,t}$ which represents the phase inside the oscillation cycle, is dependent on time; thus, information on time, e.g. timestamp η_t, must be passed into PLSTM for time gate calculation. From Eq. (2e), it can be seen that the time gate $\mathbf{k}_{i,t}$ has three phases: the opening phase (i.e., $\kappa_{i,t}$ increases from 0 to 1), the closing phase (i.e., $\kappa_{i,t}$ decreases from 1 to 0), and the closed phase. When the gate is closed, the cell and hidden states are not updated, and the previous states are maintained. Similar to the leaky ReLu function, the time gate is designed with a leak of rate α, that is active during its closed phase.

Since PLSTM only decays during the open phase of the time gate, the addition of a time gate allows PLSTM to have a slower rate of memory decay and a longer memory length compared to standard LSTM [11]. This enables PLSTM to learn longer and irregular time series better than LSTM. Details on how the time gate helps to process irregular time series will be discussed in Section VI.

IV. Generalization Performance of PLSTM

A. Preliminaries

The assumptions made on the PLSTM model and the datasets are:

1) The validation, testing, and training datasets are sampled using the same distribution.

2) The Frobenius norm of V, the weight matrix of the output layer is bounded i.e., $\|V\|_F \leq B_V$.
3) In the output layer, the activation function $\sigma_y(\cdot)$ is 1-Lipschitz continuous, and is positive homogenous. Specifically, for all $\alpha \geq 0$, $\sigma_y(\alpha z) = \alpha\sigma_y(z)$ (linear activation function is an example of $\sigma_y(z)$ for regression problems).

To simplify the discussion, we will use the augmented vector $\check{\mathbf{x}}_t$ to denote the input layer of PLSTM, $\check{\mathbf{x}}_t = [\mathbf{x}_t \; \eta_t] \in \mathbb{R}^{d_{\check{x}}}$, where $t = 1, ..., T$ and $d_{\check{x}} = d_x + 1$. Let $h(\cdot)$ represent the PLSTM functions in the set of hypotheses $\mathcal{H}$, which maps input $\check{\mathbf{x}} \in \mathbb{R}^{d_{\check{x}}}$ to output $\mathbf{y} \in \mathbb{R}^{d_y}$. The loss function is denoted as $L(\check{\mathbf{y}}, \mathbf{y})$, where $\check{\mathbf{y}}$ is the predicted PLSTM output and and $\mathbf{y}$ is the labeled / true output. In this work, the loss function adopts the Mean Squared Error (MSE). It was proved in [13] that the MSE loss function exhibits local Lipschitz continuity, provided that the true and predicted outputs $\mathbf{y}, \check{\mathbf{y}} \in \mathbb{R}^{d_y}$ are bounded. This assumption of bounded output is valid, as only a finite class of neural network models that satisfies Assumptions 1 - 3 are considered in this work.

Next, we present a formal definition of the generalization error. Let h denote a function that maps each input value a to an output value b, with an unknown distribution Z. The **generalization error** of h is defined as:

$$L_Z(h) \triangleq \mathbb{E}[L(h(a), b] = \int_{A\times B} L(h(a), b)\beta(a, b)\, da\, db \quad (3)$$

where A and B, respectively, denote the input and output vector spaces. The joint probability distribution for a and b is denoted by $\beta(a, b)$.

As the underlying probability distribution Z is unknown in most cases, the empirical error, which is calculated using data sampled from the same probability distribution Z, acts as an estimate of the generalization error. The **empirical error** of a given dataset S with size m, i.e., $S = \{s_i, i = 1, ..., m\}$, is defined as:

$$\hat{\mathbb{E}}_s[L(h(a), b)] = \frac{1}{m}\sum_{i=1}^{m} L(h(a_i), b_i) \quad (4)$$

With the assumption that the empirical error is sufficiently small and bounded, the objective of the next segment is to determine an upper bound for the generalization error. This claim on the empirical error is achievable, since the PLSTM model aims to optimize the empirical error of Eq. (4) during the training phase.

B. Rademacher Complexity Bound

The Rademacher complexity measures the capacity of a function class to fit random noise and provides a tool to analyze the generalization performance.

Definition 1: Let $\mathcal{J}$ represent a hypothesis class comprising functions of real-valued outputs. We denote S as a set of size m, $S = \{s_1, ..., s_m\}$. The definition of the empirical Rademacher complexity is given as follows for $\mathcal{J}$ associated with the dataset S:

$$\mathcal{R}_S(\mathcal{J}) = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{j\in\mathcal{J}} \frac{1}{m}\sum_{i=1}^{m} \epsilon_i\, j(s_i)\right] \quad (5)$$

where $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_m)^T$ with ϵ_i being independent and identically distributed (i.i.d.) random variables of Rademacher distribution, i.e., $\mathbb{P}(\epsilon_i = 1) = \mathbb{P}(\epsilon_i = -1) = 0.5$.

Next, we define $\mathcal{F}_t$ as the collection of loss functions associated with $\mathcal{H}$, which maps PLSTM input $\check{\mathbf{x}} \in \mathbb{R}^{d_{\check{x}}}$ to output $\mathbf{y} \in \mathbb{R}^{d_y}$ for $h \in \mathcal{H}$.

$$\mathcal{F}_t = \{f_t : (\check{\mathbf{x}}, \mathbf{y}) \to L(\mathbf{y}, h(\check{\mathbf{x}})),\ \} \quad (6)$$

where vectors $\check{\mathbf{x}}$ and $\mathbf{y}$ represent the PLSTM input and true output, respectively. The following lemma proves that the generalization error associated with the hypothesis class $\mathcal{F}_t$ can be bounded by $\mathcal{R}_S(\mathcal{F}_t)$.

Lemma 1 (c.f. Theorem 3.3. in [15]): Given an i.i.d. dataset S of size m, i.e., $S = \{s_1, ..., s_m\}$ where $s_i = (\check{\mathbf{x}}_{i,t}, \mathbf{y}_{i,t})_{t=1}^{T}$. For any $\delta > 0$, with a probability of at least $1-\delta$ over samples S, we have the following inequality for all $f_t \in \mathcal{F}_t$:

$$\mathbb{E}[f_t(\check{\mathbf{x}}, \mathbf{y})] \leq 3\sqrt{\frac{\log(\frac{2}{\delta})}{2m}} + 2\mathcal{R}_S(\mathcal{F}_t) + \frac{1}{m}\sum_{i=1}^{m} f_t(\check{\mathbf{x}}_i, \mathbf{y}_i) \quad (7)$$

The next lemma will explore the upper bound for $\|\mathbf{h}_i\|$ in PLSTM, which is essential for deriving the generalization error bound.

Lemma 2: Let $\mathcal{V}$ be a set of functions with vector-valued outputs that map PLSTM input $\check{\mathbf{x}} \in \mathbb{R}^{d_{\check{x}}}$ to $\mathbf{h} \in \mathbb{R}^{d_h}$ (i.e., hidden states). For the PLSTM model of Eq. (2), we have the following:

$$\|\mathbf{h}_{i,t}\| \leq \left(\sqrt{d_h}\right)^3 \left(\frac{\left(\sqrt{d_h}\right)^t - 1}{\sqrt{d_h} - 1}\right) = M \quad (8)$$

Proof: First, we give a short proof of the property of the Hadamard product of two vectors, $\|\mathbf{u} \odot \mathbf{v}\| \leq \|\mathbf{v}\|\|\mathbf{u}\|$. Let $\mathbf{v} = [v_1, ..., v_n]$, $\mathbf{u} = [u_1, ..., u_n]$, where $v_i, u_i \in \mathbb{R}, i = 1, 2, ..., n$. Then, it follows that

$$\begin{aligned}\|\mathbf{u} \odot \mathbf{v}\|^2 &= u_1^2 v_1^2 + u_2^2 v_2^2 + ... + u_n^2 v_n^2 \\ &\leq (u_1^2 + u_2^2 + ... + u_n^2)(v_1^2 + v_2^2 + ... + v_n^2) \\ &= \|\mathbf{u}\|^2\|\mathbf{v}\|^2\end{aligned} \quad (9)$$

Since the Euclidean norm of a vector is non-negative, we have $\|\mathbf{u} \odot \mathbf{v}\| \leq \|\mathbf{u}\|\|\mathbf{v}\|$. Next, from Eq. (2j), we can decompose the hidden state in the following way:

$$\begin{aligned}\|\mathbf{h}_{i,t}\| &\leq \|\mathbf{k}_{i,t}\|\|\hat{\mathbf{h}}_{i,t}\| + \|(1 - \mathbf{k}_{i,t})\|\|\mathbf{h}_{i,t-1}\| \\ &\leq \sqrt{d_h}(\|\hat{\mathbf{h}}_{i,t}\| + \|\mathbf{h}_{i,t-1}\|) \\ &\leq \sqrt{d_h}(\|\mathbf{o}_{i,t}\|\|\tanh(\hat{\mathbf{c}}_{i,t})\| + \|\mathbf{h}_{i,t-1}\|) \\ &\leq \left(\sqrt{d_h}\right)^3 + \sqrt{d_h}\|\mathbf{h}_{i,t-1}\| \\ &\leq \left(\sqrt{d_h}\right)^3 \left(\frac{\left(\sqrt{d_h}\right)^t - 1}{\sqrt{d_h} - 1}\right) = M\end{aligned} \quad (10)$$

The first line is based on the expansion of $\mathbf{h}_{i,t}$ using the hidden state definition in Eq. (2j) and the property of the Hadamard product, where $\|\mathbf{u} \odot \mathbf{v}\| \leq \|\mathbf{u}\|\|\mathbf{v}\|$. The second line is obtained via the fact that $\kappa_{i,t} \leq 1$; hence, $\|\mathbf{k}_{i,t}\| = \sqrt{\sum_{j=1}^{d_h} |\kappa_{i,t}|^2} \leq \sqrt{d_h}$, where j represents the j-th neuron in the PLSTM layer. Similarly, $\|(1-\mathbf{k}_{i,t})\| \leq \sqrt{d_h}$. The third line is obtained by expanding on the term $\hat{\mathbf{h}}_{i,t}$ using Eq. (2i). The fourth line is derived based on the fact that $\|\tanh(\cdot)\|, \|\sigma(\cdot)\| \leq 1$ (see Eq. (2c) for the definition of $\mathbf{o}_{i,t}$). With the assumption that the initial hidden state $\mathbf{h}_{i,0} = 0$, the last line is obtained by applying the fifth line recursively to get a geometric series. ■

The next lemma shows that the Rademacher complexity of the PLSTM function class has an upper bound.

Lemma 3: Given an i.i.d. dataset S of size m. Consider a hypothesis class of functions $\mathcal{H}_{k,t}$, $k = 1, ..., d_y$, with real-valued output that corresponds to the k-th element of the PLSTM output at the t-th time instance. If the assumptions we made earlier for PLSTMs are satisfied, we have

$$\mathcal{R}_S(\mathcal{H}_{k,t}) \leq MB_V \tag{11}$$

where $M = \left(\frac{(\sqrt{d_h})^t - 1}{\sqrt{d_h} - 1}\right)\left(\sqrt{d_h}\right)^3$.

Proof: In the weight matrix V, the k-th row is denoted by $\mathbf{v}_k$. Using the idea of "peeling" off the output layer, the scaled Rademacher complexity can be bounded as follows:

$$\begin{aligned} & m\mathcal{R}_S(\mathcal{H}_{k,t}) \\ & = \mathbb{E}\left[\sup_{h\in\mathcal{H}_{k,t}, \|V\|_F \leq B_V} \sum_{i=1}^{m} \epsilon_i \sigma_y(\mathbf{v}_k \mathbf{h}_{i,t})\right] \\ & \leq \frac{1}{\lambda}\log \mathbb{E}\left[\sup_{h\in\mathcal{H}_{k,t}} \exp\left(B_V \lambda \left\|\sum_{i=1}^{m} \epsilon_i \mathbf{h}_{i,t}\right\|\right)\right] \\ & \leq \frac{1}{\lambda}\log \mathbb{E}\left[\exp\left(B_V \lambda M \sum_{i=1}^{m} |\epsilon_i|\right)\right] \\ & \leq \frac{1}{\lambda}\log \mathbb{E}\left[\exp\left(B_V \lambda M m\right)\right] \end{aligned} \tag{12}$$

where $M = \left(\sqrt{d_h}\right)^3\left(\frac{(\sqrt{d_h})^t - 1}{\sqrt{d_h} - 1}\right)$, $\exp(\cdot)$ corresponds to the monotonically increasing function $g(\cdot)$ in Lemma 1 of [14], and $\lambda > 0$ is an arbitrary parameter. The first inequality is a result of Lemma 4 in [13] and Lemma 1 in [14]. The second inequality is a consequence of the triangle inequality and the upper bound of the hidden state $\mathbf{h}_{i,t}$ in Lemma 2. The final line is based on the fact that $|\epsilon_i| = 1$, since $\mathbb{P}(\epsilon_i = -1) = \mathbb{P}(\epsilon_i = 1) = \frac{1}{2}$.

Let $q = B_V M m$. Note that q is a constant and $\mathbb{E}[a] = a$, where $a \in \mathbb{R}$ is a constant. Thus, we have

$$\begin{aligned} m\mathcal{R}_S(\mathcal{H}_{k,t}) & \leq \frac{1}{\lambda}\log \mathbb{E}[\exp(\lambda q)] \\ & \leq \frac{1}{\lambda}\log[\exp(\lambda q)] \\ & = B_V M m \end{aligned} \tag{13}$$

Simplifying the above equation, we will get

$$\mathcal{R}_S(\mathcal{H}_{k,t}) \leq MB_V \tag{14}$$

With this, we have established Lemma 3. ■

Using Lemmas 1 – 3, the following theorem will establish the generalization error bound for the PLSTM model.

Theorem 1: Given an i.i.d. dataset S of size m. Let $\mathcal{F}_t$ be the collection of loss functions calculated using the vector-valued function set $\mathcal{H}_t$ that maps the PLSTM inputs to the outputs at the t-th time step. If Assumptions 1 – 3 are satisfied, for any $\delta > 0$, with a probability of at least $1-\delta$ over S, the following inequality holds:

$$\mathbb{E}[f_t(\check{\mathbf{x}}, \mathbf{y})] \leq 3\sqrt{\frac{\log(\frac{2}{\delta})}{2m}} + \mathcal{O}(L_r d_y M B_V) + \frac{1}{m}\sum_{i=1}^{m} f_t(\check{\mathbf{x}}_i, \mathbf{y}_i) \tag{15}$$

where $M = \left(\sqrt{d_h}\right)^3\left(\frac{(\sqrt{d_h})^t - 1}{\sqrt{d_h} - 1}\right)$.

Proof: Using Lemma 3, we can bound the Radamacher complexity of the loss function as follows:

$$\begin{aligned} \mathcal{R}_S(\mathcal{F}_t) & = \mathbb{E}\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \epsilon_i L(\mathbf{y}_i, h(\check{\mathbf{x}}_i))\right] \\ & \leq \sqrt{2}L_r \sum_{k=1}^{d_y} \mathbb{E}\left[\sup_{h\in\mathcal{H}_k} \frac{1}{m}\sum_{i=1}^{m} \epsilon_i h(\check{\mathbf{x}}_i)\right] \\ & \leq \sqrt{2}L_r \sum_{k=1}^{d_y} \mathcal{R}_S(\mathcal{H}_k) \\ & \leq \sqrt{2}L_r d_y M B_V \end{aligned} \tag{16}$$

The first inequality is the result of a contraction inequality presented in Corollary 4 of [12], where the vector-valued function class $\mathcal{H}$ consisting of $h \in \mathbb{R}^{d_y}$ can be further bounded by its scalar-valued functions. Substituting the upper bound of the loss function's Rademacher complexity into Eq. (16) into Eq. (7) will give us the upper bound to the generalization error, shown in Eq. (15). ■

V. PLSTM-BASED MPC WITH CLOSED-LOOP STABILITY ANALYSIS

This section discusses the design and closed-loop stability properties of a Lyapunov-based MPC (LMPC) using PLSTM as the prediction model. The generalization error bound derived in the previous section will be used to quantify the model-plant mismatch in the design of MPC.

A. Lyapunov-Based Control Using PLSTM Model

The PLSTM model can be described as follows:

$$\dot{\hat{x}} = F_{nn}(\hat{x}, u) \tag{17}$$

where $\hat{}$ is used to differentiate the variables / functions associated with the PLSTM model. The PLSTM state and the manipulated input vectors are denoted by $\hat{x} \in \mathbb{R}^n$ and $u \in \mathbb{R}^k$, respectively, and u is constrained by $u \in U$.

It is assumed that there is a controller for stabilization $u = \Phi_{nn}(x) \in U$ by which the PLSTM model of Eq. (17) is rendered exponentially stable in an open set $\hat{O}$ around the origin. The assumption suggests the existence of a $\mathcal{C}^1$ Lyapunov function $\hat{V}(x)$ which meets the constraints below:

$$\theta_1\|x\|^2 \leq \hat{V}(x) \leq \theta_2\|x\|^2 \tag{18}$$

$$\frac{\partial \hat{V}(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x)) \leq -\theta_3\|x\|^2 \tag{19}$$

$$\left\|\frac{\partial \hat{V}(x)}{\partial x}\right\| \leq \theta_4\|x\| \tag{20}$$

where $\theta_1, \theta_2, \theta_3, \theta_4$ are positive constants. For simplicity, the Lyapunov function $\hat{V}(x)$ associated with the PLSTM model will be expressed as $V(x)$ in the subsequent discussion.

The boundedness of u and the Lipschitz property of $F_{nn}(x,u)$ also indicates the existence of positive constants M_{nn}, L_{nn}, such that for all $x', x \in \hat{O}$ and $u \in U$, we have the inequalities below:

$$\|F_{nn}(x,u)\| \leq M_{nn} \tag{21}$$

$$\left\| \frac{\partial V(x')}{\partial x} F_{nn}(x',u) - \frac{\partial V(x)}{\partial x} F_{nn}(x,u) \right\| \leq L_{nn}\|x - x'\| \tag{22}$$

The $\hat{\rho}$-level set of the $\hat{V}$ characterizes the closed-loop stability region $\Omega_{\hat{\rho}}$ of the PLSTM model of Eq. (17), where $\Omega_{\hat{\rho}} := \{x \in \hat{O} \mid V(x) \leq \hat{\rho}\}$, and $\hat{\rho}$ is a positive real number.

The modeling error $\|F_{nn}(x,u) - F(x,u)\|$ quantifies the plant-model mismatch between the PLSTM model of Eq. (17) and the system of Eq. (1). If the modeling error is sufficiently small, the following proposition will prove that the closed-loop stability of Eq. (1) is ensured under the feedback control law $u = \Phi_{nn}(x) \in U$, with significant probability.

Proposition 1 (c.f. Proposition 1 of [13]): Consider a PLSTM model trained using an i.i.d. dataset S, satisfying Assumptions 1-3. Assuming that the feedback controller $u = \Phi_{nn}(x) \in U$ guarantees that the origin of the PLSTM model of Eq. (17) is exponentially stable, for all $x \in \Omega_{\hat{\rho}}$ and $u \in U$. If there exists a positive real number ζ that satisfies $\zeta < \theta_3/\theta_4$, such that the modeling error is bounded by $\|F_{nn}(x,u) - F(x,u)\| \leq \zeta\|x\|$. Then, for all $x \in \Omega_{\hat{\rho}}$, the PLSTM-based controller $u = \Phi_{nn}(x) \in U$ guarantees the exponential stability of $x = 0$ of Eq. (1), with a probability of no less than $1 - \delta$.

Proof: The full proof can be found in Proposition 1 of [13]. It should be noted that the generalization error bound in [13] was derived for a standard RNN model. Replacing the PLSTM generalization error bound in Eq. (15) with its RNN counterpart in [13], will give the following:

$$\|F_{nn}(x, \Phi_{nn}(x)) - F(x, \Phi_{nn}(x))\| \leq B_P \tag{23}$$

where B_P is a function of the generalization error bound, $B_P = \mathcal{P}\left(3\sqrt{\frac{\log(\frac{2}{\delta})}{2m}} + \mathcal{O}(L_r d_y M B_V) + \frac{1}{m}\sum_{i=1}^{m} f_t(\check{\mathbf{x}}_i, \mathbf{y}_i)\right)$. Note that B_P is dependent on the number of samples m and the confidence interval δ. To bound B_P, we can select a considerable amount of samples such that $m \geq m_N(\delta, \|x\|)$, where the minimum requirement for the sample size is denoted by $m_N(\delta, \|x\|)$, to satisfy the condition $B_P \leq \zeta\|x\|$, $\zeta < \theta_3/\theta_4$. The remaining proof follows that in [13]. ■

B. Missing Real-time Data

One consequence of having data loss or delayed measurements is the possible absence of current state readings, which is necessary for MPC calculation. Fortunately, the current state can be estimated using the machine learning model recursively. For instance, in the case of data loss, data is not available at the current sampling time t_k, but data from the past sampling time t_{k-j} is available, where $t_k - t_{k-j} = j\Delta$ and Δ represents one sampling period. Real-time data at t_k can be estimated recursively using the model, based on data from the previous sampling time t_{k-j}. Similarly, for delayed data, if the current reading at t_k is unavailable but the data from t_{k-d} is received at t_k, i.e., there is a delay of $d\Delta$, the same method can be employed to estimate the current state at t_k. Due to the similarity between the handling of the independent occurrences of data loss and delayed measurements, in the subsequent discussions, we will focus on the worst-case scenario, where both data loss and delayed measurements occur at the same time. An upper bound H is defined on the overall 'missingness' of the data, $j + d \leq H$, where j and d represent the number of sampling points where the system is operating without real-time data due to data loss and delays in data collection, respectively. This is to limit the system from operating in an open-loop for a prolonged time, which could lead to instability.

Assuming that the measurements are time-labeled, the following Lyapunov-based MPC (LMPC) design is formulated using the PLSTM model to handle missing data.

$$\mathcal{J} = \min_{u_k \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), u_k(t))\, dt \tag{24a}$$

$$\text{s.t. } \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), u_k(t)) \tag{24b}$$

$$u_k(t) = u_{k-1}^*(t),\ \forall t \in [t_{k-h}, t_k) \tag{24c}$$

$$\tilde{x}(t_{k-h}) = x(t_{k-h}) \tag{24d}$$

$$\dot{\hat{x}}(t) = F_{nn}(\hat{x}, \Phi_{nn}(x(t_k))),\ t \in [t_k, t_{k+N}) \tag{24e}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{24f}$$

$$V(\tilde{x}(t)) \leq V(\hat{x}(t)),\ \forall t \in [t_k, t_{k+N}) \tag{24g}$$

$$u_k(t) \in U,\ \forall t \in [t_k, t_{k+N}) \tag{24h}$$

$$\text{with } u_k^*(t) = \arg\min_{u_k \in S(\Delta)} \mathcal{J} \tag{24i}$$

where N represents the number of predictions, such that $N \geq H + 1$ and H is the maximum number of sampling steps where the system operates in open-loop. $S(\Delta)$ denotes the set of decision variables which are piece-wise constant functions corresponding to sample-and-hold implementation. $u_{k-1}^*(t)$ is the optimal solution obtained from LMPC at the last time step t_{k-1}. $\tilde{x}(t)$ is the predicted state trajectory of the nonlinear system of Eq. (1). $\hat{x}(t)$ is the predicted trajectory of the estimated state $\tilde{x}(t_k)$ under the controller $\Phi_{nn}(x(t_k))$.

The LMPC formulation consists of two main segments: estimation and prediction. First, we estimate the current state $\tilde{x}(t_k)$ from past measurement $x(t_{k-h})$. Next, we predict the state trajectory $\tilde{x}(t)$ using the estimated current state $\tilde{x}(t_k)$. Note that if the measurement at t_k is available real-time, we can skip the estimation process.

In detail, if the current reading at time t_k is not available but a past measurement at t_{k-h} is available, the PLSTM model will recursively estimate the current state $\tilde{x}(t_k)$ using the past control inputs applied to the system from t_{k-h} to t_k, with the initial condition being $\tilde{x}(t_{k-h}) = x(t_{k-h})$ (see Eqs. (24b), (24c) and (24d)). The estimated current state $\tilde{x}(t_k)$ is subsequently employed to solve the LMPC problem in Eq. (24a). In other words, the PLSTM model of Eq. (24b) will be called again, but this time to predict the state trajectory $\tilde{x}(t)$ over the prediction horizon N, from the estimated current state $\tilde{x}(t_k)$. Then, LMPC will try to minimize the objective function in Eq. (24a), by searching for an optimal $u_k^*(t)$, subjected to control input constraints as shown in Eq. (24h). The contractive constraint of Eq. (24g) ensures that the closed-loop state moves towards the origin, where $\hat{x}(t)$ is the predicted state trajectory subjected to controller $\Phi_{nn}(x(t_k))$, with $\tilde{x}(t_k)$ as its initial condition, see Eqs. (24e) and (24f). In the case where the available measurement at t_k is a delayed measurement from t_{k-h}, we can still apply the above-mentioned method.

In the following segment, we will prove that the LMPC of Eq. (24) is able to ensure the stability of the nonlinear system of Eq. (1) in the presence of data loss and delayed measurements, under sample-and-hold implementation.

Theorem 2: Consider the closed-loop system of the nonlinear system of Eq. (1) under the PLSTM-MPC of Eq. (24) with $u = \Phi_{nn}(x) \in U$ that meets Eqs. (18)-(20). Let $\Delta, \epsilon_s, \epsilon_h > 0$, and $\hat{\rho} > \rho_h > \rho_{nn} > \rho_s$ satisfy the following:

$$-\frac{\theta_3}{\theta_2}\rho_s + L_{nn}M_{nn}\Delta \leq -\epsilon_s \tag{25}$$

$$\rho_{nn} := \max\{V(\hat{x}(t_{i+1}) \mid \ u \in U, \ \hat{x}(t_i) \in \Omega_{\rho_s}\} \tag{26}$$

and

$$-\Delta\epsilon_s + f_v(f_w(H\Delta)) + f_v(f_w(H+1)\Delta) < -\epsilon_h \tag{27}$$

where $f_w(\cdot)$ is a class $\mathcal{K}$ function and $f_v(\cdot)$ is quadratic function with $\xi > 0$, given by:

$$f_w(\tau) := \frac{B_P}{L_x}(e^{L_x\tau} - 1) \qquad f_v(\lambda) := \frac{\theta_4\sqrt{\hat{\rho}}}{\sqrt{\theta_1}}\lambda + \xi\lambda^2$$

If $H+1 \leq N$, $x_0 \in \Omega_{\hat{\rho}}$ and the initial measurement x_0 is not subjected to delay or asynchronicity, then $x(t)$ is bounded in Ω_{ρ_h}, with a probability of at least $1 - \delta$, where:

$$\rho_h \geq \rho_{nn} + f_v(f_w(H\Delta)) + f_v(f_w((H+1)\Delta)) \tag{28}$$

Proof: In this proof, we will show that $V(x)$ is a decreasing function of time. First, we assume that the current measurement is not available at t_k but past measurement at t_{k-h} is available. We also assume that a new reading is not available until t_{k+g} where $g \leq N$. The LMPC optimization problem is solved from t_k to t_{k+g} using PLSTM model for estimation.

Let $\hat{x}$ be the predicted trajectory of the PLSTM model of Eq. (17) in closed-loop with controller $u = \Phi_{nn}(x(t_k)) \in U$, under sample-and-hold implementation, with initial state $\tilde{x}(t_k)$ (see Eqs. (24e) and (24f)). The derivation of Eqs. (25) and (26) can be found in the proof of Proposition 4 in [16]. It is important to emphasize that if the condition in Eq. (25) is met, then for all $\hat{x}(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_s}$ and $t \in [t_k, t_{k+1})$, we have:

$$\dot{V}(\hat{x}(t)) \leq -\epsilon_s \tag{29}$$

By integrating Eq. (29) over $t \in [t_k, t_{k+g})$, we have $V(\hat{x}(t_{k+g})) \leq V(\hat{x}(t_k)) - g\Delta\epsilon_s$. If Eq. (26) is satisfied, the closed-loop state $\hat{x}$ of the PLSTM model of Eq. (17) is always bounded in $\Omega_{\rho_{nn}}$. Using this fact, we have the following:

$$V(\hat{x}(t_{k+g})) \leq \max\{V(\hat{x}(t_k) - g\Delta\epsilon_s, \rho_{nn}\} \tag{30}$$

Using the contractive constraint $V(\tilde{x}(t)) \leq V(\hat{x}(t))$ in Eq. (24g) and the initial condition of $\hat{x}$ in Eq. (24f), we have

$$\begin{aligned} V(\tilde{x}(t_{k+g})) &\leq V(\hat{x}(t_{k+g})) \\ &\leq \max\{\rho_{nn}, V(\tilde{x}(t_k) - g\Delta\epsilon_s\} \end{aligned} \tag{31}$$

Assuming that $x(t), \tilde{x}(t) \in \Omega_{\hat{\rho}}$ for all $t \in [t_{k-h}, t_{k+g})$, we can derive the following using Proposition 2 in [13]:

$$V(\tilde{x}(t_k)) \leq f_v(\|\tilde{x}(t_k) - x(t_k)\|) + V(x(t_k)) \tag{32}$$

$$V(x(t_{k+g})) \leq f_v(\|\tilde{x}(t_{k+g}) - x(t_{k+g})\|) + V(\tilde{x}(t_{k+g})) \tag{33}$$

Using Proposition 2 in [13] again, we can bound the terms below.

$$\|\tilde{x}(t_k) - x(t_k)\| \leq f_w(h\Delta) \tag{34}$$

$$\|\tilde{x}(t_{k+g}) - x(t_{k+g})\| \leq f_w((h+g)\Delta) \tag{35}$$

Combining Eqs. (31)-(35), we can bound $V(x(t_{k+g}))$ in the following manner:

$$\begin{aligned} V(x(t_{k+g})) \leq\ & f_v(f_w((h+g)\Delta)) + f_v(f_w(h\Delta)) \\ & + \max\{\rho_{nn}, V(x(t_k) - g\Delta\epsilon_s\} \end{aligned} \tag{36}$$

It is noted that the missing data interval, $h + g - 1$, from t_{k-h} to t_{k+g} can be any natural number (including 0, which indicates no missing data) smaller than its upper bound H. To demonstrate that $V(x)$ is decreasing over time, the worst-case scenario is considered, where the missing interval is H, i.e., $g = H + 1 - h$. To simplify the discussion, we let $g = 1$. This implies that the data were missing for the H intervals before t_k and a new measurement will be available at the next sampling time t_{k+1}. Therefore, we have $h = H$ and the system is operating in an open loop for $H + 1$ sampling periods. If the following constraint holds, then $V(x)$ is guaranteed to decrease with time.

$$-\Delta\epsilon_s + f_v(f_w(H\Delta)) + f_v(f_w((H+1)\Delta)) < 0 \tag{37}$$

If Eq. (27) is satisfied, a negative real number $-\epsilon_h$ can be found to bound $V(x(t))$.

$$V(x(t_{k+1})) \leq \max\{\rho_h, V(x(t_k) - \epsilon_h\} \tag{38}$$

Note that f_w is a function containing B_P. As mentioned in Proposition 1, by selecting the sample size m, such that $B_P \leq \zeta\|x\|$, the probability of $V(x(t_{k+1})) \leq V(x(t_k)$ is greater than $1 - \delta$. This shows that $u = \Phi_{nn}(x) \in U$ will drive the state of the actual system of Eq. (1) to Ω_{ρ_h}, with a probability of no less than $1 - \delta$, when the state $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_h}$. If the state $x(t_k)$ is in Ω_{ρ_h}, it will remain bounded in Ω_{ρ_h}. Given the upper bound on the missing interval $h + g - 1 \leq H$, we select the prediction horizon $N \geq H + 1$ such that the assumption $g \leq N$ is valid. Since $V(x)$ is guaranteed to decrease with time, the previous assumption that $\tilde{x}(t), x(t), \in \Omega_{\hat{\rho}}$ for all $t \in [t_{k-h}, t_{k+g})$ is valid. ■

VI. Case Study of a Chemical Reactor

This section demonstrates the implementation of PLSTM-based MPC to a Continuous Stirred Tank Reactor (CSTR). The process that occurs within the CSTR is an irreversible exothermic reaction of second order. It is assumed that the reactor is perfectly mixed and non-isothermal. A heating jacket is used to maintain the reactor temperature T, with a heat input rate of Q. The CSTR dynamics can be generally captured by its mass and energy balances:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} {C_A}^2 \tag{39}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{Q}{\rho_L C_p V} + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} {C_A}^2 \tag{40}$$

where the concentration of species A and reactor temperature are C_A and T. The terms Q and C_{A0} denote the rate of heat supply and feed concentration of A, respectively. Detailed descriptions of the remaining notation and the process parameters can be found in [7].

The input variables in this process are C_{A0} and Q, and the process variables to be controlled (i.e., state variables) are C_A and T. The manipulated input u and the state variables x are represented in terms of deviation variables, denoted by $u^T = [C'_{A0}, Q']$ and $x^T = [C'_A, T']$, where the origin represents the steady-state of the process. The objective of PLSTM-MPC is to stabilize and manage the CSTR in its unstable steady state, $C_{As} = 1.95$ kmol/m^3 and $T_s = 402$ K by adjusting C_{A0} and Q using the PLSTM-based MPC. In addition, this control system is subjected to data loss and delayed measurements of the state variables, C_A and T. To mimic real-world systems, we impose the following constraints on the control actions: $|Q'| \leq 5 \times 10^5$ kJ/h and $|C'_{A0}| \leq 3.5 \mathrm{kmol/m}^3$.

The designed Lyapunov function is in the form of $V(x) = x^T P x$, where $P = [1.06 \times 10^3 \; 22.0; \; 22.0 \; 0.52]$, and $\Omega_{\hat{\rho}}$ denotes the operating region with $\hat{\rho} = 372$, i.e., $V(x) \leq 372$, for the CSTR. For all initial states $x_0 \in \Omega_{\hat{\rho}}$ and control inputs $u \in U$, forward Euler method was applied with $h_c = 2 \times 10^{-4}$ h as the integration time interval.

1) Model description: It is assumed that the state measurements, which are collected by sensors in real life, are affected by asynchronicity and delay. As such, we created a new dataset from the open-loop dataset, containing samples of assumed missing rates between 66.7% to 98.7%.

The PLSTM model design is provided in Fig. 2. The PLSTM layer contains 24 neurons and the LSTM layer contains 8 neurons. The input of the PLSTM model is in the form of two input vectors $< \mathbf{x}, \eta >$ where η is the timestamp vector and $\mathbf{x} = [x_p, u_p, u]$, where x_p is the past state measurements (with missing data) from the previous two sampling periods (i.e. from t_{k-2} to t_k), u_p is the corresponding past control actions and u is the current control action at t_k. The model output $\mathbf{y}$ is the state measurements for the next two sampling periods, i.e., $x(t_{k+1})$ and $x(t_{k+2})$. It is important to note that missing data are only relevant to the input data and not to the output data; we assumed that the state measurements at the next two sampling periods are always available for modeling purposes.

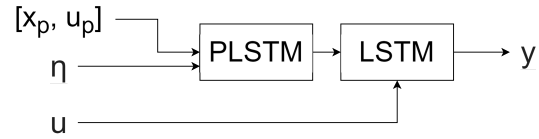

Fig. 2. A schematic of PLSTM architecture for CSTR.

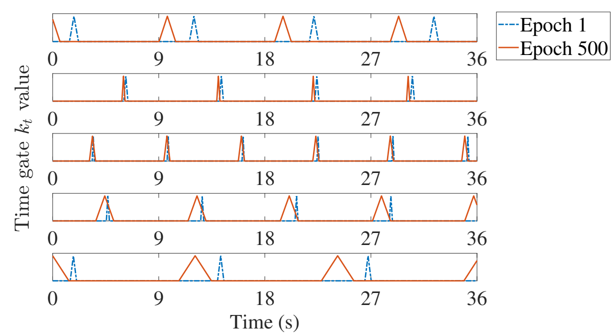

Fig. 3. Time gate evolution of the first 5 neurons from training epoch 1 (blue line) to 500 (red line).

The evolution of the time gate over the training process was monitored and presented in Fig. 3. From Fig. 3, it is observed that the opening phase of the time gate of some neurons has increased during the training process, to allow more frequent updates. It is observed that the configuration of PLSTM is unique in the sense that only a portion of neurons are active at any time. This is similar to having a dropout layer with varying dropout rates at every time step. This structure could have contributed to PLSTM's ability in processing irregularly sampled data; it is able to reduce overfitting by filtering out unnecessary information to extract the intrinsic relationships between the input and output.

To assess the performance of the PLSTM model in forecasting the future states, a testing dataset containing fully observed input data was used. This is to assess the effectiveness of the PLSTM model in capturing the underlying dynamics of CSTR, when trained with historical missing data. The MSE of PLSTM against the fully observed dataset is 2.44×10^{-2}. Additionally, an LSTM model using the same missing dataset was also developed, with the testing error 9.96×10^{-1}. Additionally, we develop a conventional LSTM model using the fully observed dataset (i.e., no missing data). It serves as a reference for model performance evaluation, since this is the best machine learning model that one can develop with state measured at every sampling step. The testing error of the reference LSTM model (i.e., best LSTM model) is 3.61×10^{-6}. All the models were developed with the same hyperparameters (i.e., number of layers, neurons, optimizers, initialization).

2) Simulation of closed-loop system under MPC: Finally, we perform closed-loop simulations under MPC using PLSTM and LSTM models. For PLSTM-based MPC and LSTM-based MPC, we consider real-time missing data

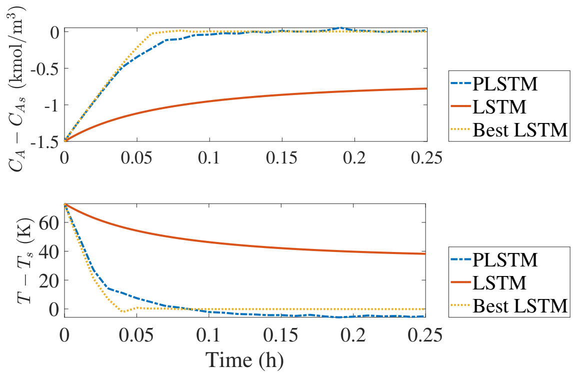

Fig. 4. State profiles under LMPC using PLSTM (blue trajectory), LSTM (red trajectory) and Best LSTM (yellow trajectory) for the initial condition (-1.5 kmol/m^3, 73 K).

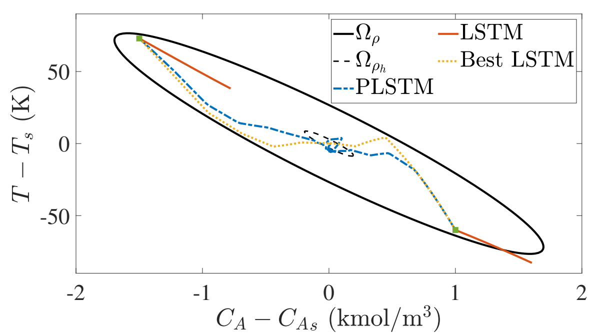

Fig. 5. State trajectories for the closed-loop simulation under LMPC with different initial conditions (marked with green squares) using PLSTM (blue trajectory), LSTM (red trajectory) and Best LSTM (yellow trajectory).

in feedback control. The maximum allowable number of sampling periods with missing real-time measurements, H, is chosen to be 1. The Best LSTM-based MPC, on the other hand, was not subjected to real-time missing data (i.e., it received full state measurements at all sampling times). This acts as the benchmark for assessing the performance of PLSTM-based MPC, as it represents the performance of MPC under ideal conditions, using the most accurate machine learning model as its predictive model. Since $N \geq H + 1$, the prediction horizon N is selected to be 2.

Figs. 4 and 5 show the state profiles and state-space trajectories under PLSTM- and LSTM-based MPCs. It can be seen that the LSTM-based MPC has the worst MPC performance and does not guarantee closed-loop stability. On the other hand, the PLSTM-based MPC showed comparable performance with the Best LSTM-based MPC. By preserving the states in the operating region Ω_ρ and ultimately driving the states to the steady state, both MPC schemes have achieved closed-loop stability. Therefore, it is concluded from the case study that PLSTM can efficiently capture process dynamics using datasets with delayed/asynchronous measurements, and stability of closed-loop system is maintained under the PLSTM-MPC scheme.

VII. Conclusion

This work developed a PLSTM model for modeling nonlinear chemical processes with data loss in state measurements. A theoretical analysis was first performed for the generalization performance of PLSTM. Subsequently, an MPC scheme was designed using the PLSTM model and accounting for the missing real-time data in feedback control. Closed-loop stability was further demonstrated using the error bound of PLSTMs. Lastly, to demonstrate the effectiveness of PLSTM-based MPC for real-time control of chemical processes with irregular state measurements, PLSTM was applied to a CSTR example.

References

[1] R. A. Gupta and M. Y. Chow, *Overview of Networked Control Systems*, pp. 1–23. London: Springer London, 2008.

[2] H. Xu, S. Jagannathan, and F. L. Lewis, "Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses," *Automatica*, vol. 48, no. 6, pp. 1017–1030, 2012.

[3] J. H. Kim and H. B. Park, "H$^\infty$ state feedback control for generalized continuous/discrete time-delay system," *Automatica*, vol. 35, no. 8, pp. 1443–1451, 1999.

[4] J. Liu, D. Muñoz de la Peña, P. D. Christofides, and J. F. Davis, "Lyapunov-based model predictive control of nonlinear systems subject to time-varying measurement delays," *International Journal of Adaptive Control and Signal Processing*, vol. 23, no. 8, pp. 788–807, 2009.

[5] D. Muñoz de la Peña and P. D. Christofides, "Lyapunov-based model predictive control of nonlinear systems subject to data losses," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2076–2089, 2008.

[6] Y. Pan and J. Wang, "Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3089–3101, 2012.

[7] Z. Wu, A. Tran, D. Rincon, and P. D. Christofides, "Machine-learning-based predictive control of nonlinear processes. part ii: Computational implementation," *AIChE Journal*, vol. 65, no. 11, p. e16734, 2019.

[8] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware lstm networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 65–74, 2017.

[9] N. Vecoven, D. Ernst, and G. Drion, "A bio-inspired bistable recurrent cell allows for long-lasting memory," *PLOS One*, vol. 16, no. 6, p. e0252676, 2021.

[10] P. B. Weerakody, K. W. Wong, G. Wang, and W. Ela, "A review of irregular time series data handling with gated recurrent neural networks," *Neurocomputing*, vol. 441, pp. 161–178, 2021.

[11] D. Neil, M. Pfeiffer, and S. C. Liu, "Phased lstm: Accelerating recurrent network training for long or event-based sequences," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[12] A. Maurer, "A vector-contraction inequality for rademacher complexities," in *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*, pp. 3–17, Springer, 2016.

[13] Z. Wu, D. Rincon, Q. Gu, and P. D. Christofides, "Statistical machine learning in model predictive control of nonlinear processes," *Mathematics*, vol. 9, no. 16, p. 1912, 2021.

[14] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," in *Conference On Learning Theory*, pp. 297–299, PMLR, 2018.

[15] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[16] Z. Wu, A. Tran, D. Rincon, and P. D. Christofides, "Machine learning-based predictive control of nonlinear processes. part i: Theory," *AIChE Journal*, vol. 65, no. 11, p. e16729, 2019.