Online Machine Learning Modeling and Predictive Control of Switched Nonlinear Systems

Cheng Hu [a] and Zhe Wu [a,1]

[a] Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore

Abstract

Online machine learning (ML) method updates ML models at each time step when data becomes available in a sequential order, and has shown its potential for improving modeling of nonlinear chemical processes using real-time data. This paper highlights our recent work on the analysis of generalization performance of online learning of recurrent neural network (RNN) models for nonlinear dynamic systems, and the development of ML-based model predictive control (MPC) using online learning models for switched nonlinear systems. We first develop the generalization error bounds for online learning RNNs using independent and identically distributed (i.i.d.) and non-i.i.d. data samples, respectively. Then, online learning models are incorporated into MPC and probabilistic closed-loop stability results are derived for switched nonlinear systems. Finally, we conclude with discussions on a practical implementation strategy of online learning within ML-based MPC via event-trigger and error-trigger mechanisms.

Keywords

Online Learning; Recurrent Neural Networks; Generalization Error; Model Predictive Control; Nonlinear Systems.

Introduction

Online learning algorithms have been widely used to develop machine learning (ML) models for large-scale problems with a tremendous amount of data, since the training process is more computationally efficient than batch algorithms. In addition to the considerations of computational efficiency, online learning has demonstrated its benefits in improving model prediction and closed-loop performance in many real-time control problems (Schwenkel et al. (2020)). However, an ongoing challenge for the practical implementation of online learning models in real-world chemical processes is their generalization performance on unseen data, for which a fundamental understanding needs to be developed. Generalization error bound is commonly used in statistical machine learning to quantitatively characterize the generalization performance of machine learning models. Online-to-batch conversion is a common technique in machine learning theory to evaluate the generalization performance of online learning algorithms, which has been studied in the setting of i.i.d. data samples (Cesa-Bianchi et al. (2004)) and of non-i.i.d. data samples (Kuznetsov and Mohri (2016)) for conventional ML methods such as kernal perceptron models. However, the generalization error bounds for online learning of recurrent neural networks (RNNs) that model nonlinear dynamic systems have not been studied.

In addition, online learning of RNN models has been utilized in MPC to achieve real-time control of nonlinear processes (Wu et al. (2019), Zheng et al. (2022)). Despite an increasing number of successful applications of ML-based MPC to real-world chemical processes, theoretical stability analysis for the MPC using online learning of RNN models has not been investigated. Motivated by the above considerations, this work summarizes our recent work on the derivation of generalization error bounds for online learning RNN models and closed-loop stability analysis for the ML-based MPC scheme. Specifically, we first derive the generalization error bounds for online learning of RNNs using i.i.d. and non-i.i.d. data samples, respectively. Closed-loop stability analysis for switched nonlinear systems under ML-based MPC using online learning is provided based on the generalization error results derived for online learning models. Finally, a practical implementation strategy that determines when to trigger the online update of RNN models is discussed.

Class of nonlinear process systems

We consider the class of continuous-time switched nonlinear systems represented by the following state-space form:

$$\dot{x}(t) = F_{\sigma}(t)(x, u_{\sigma(t)}) := f_{\sigma(t)}(x) + g_{\sigma(t)}(x)u_{\sigma(t)}(t) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u_{\sigma(t)} \in \mathbb{R}^{n_u}$ denotes the control input vector that is constrained by a nonempty set $U_{\sigma(t)} := \left\{u_{\sigma(t)} \in \mathbb{R}^{n_u} \mid u^{min}_{\sigma(t)} \leq u_{\sigma(t)} \leq u^{max}_{\sigma(t)}\right\}$, where $u^{min}_{\sigma(t)}$

[1] Corresponding author. Email: wuzhe@nus.edu.sg.
The financial support from the NUS Start-up Grant is gratefully acknowledged.

and $u^{\max}_{\sigma(t)}$ represent the minimum and maximum magnitude of the input constraint. We define a finite index set $\psi = \{1, 2, \ldots, p\}$, where p denotes the number of switching modes. $\sigma(t) : [0, \infty) \to \psi$ denotes the switching function. Throughout this manuscript, we use t^{in}_k and t^{out}_k to denote the time when the nonlinear system of Eq. 1 is switched in and out of the k-th mode, respectively. Without loss of generality, we assume that the initial time is zero ($t_0 = 0$) and the initial state is given by $x(t_0) = x_0$. Additionally, we also assume that the vector functions $f_k : \mathbb{R}^n \to \mathbb{R}^n$ and the matrix functions $g_k : \mathbb{R}^n \to \mathbb{R}^{n\times n_u}$, $k \in \psi$, are sufficiently smooth.

Recurrent neural networks (RNN)

We consider the following one-hidden-layer RNN with T sequences of L_{nn}-time-length data $(\mathbf{x}_{t,\ell}, \mathbf{y}_{t,\ell})$, $t = 1, \ldots, T$ and $\ell = 1, \ldots, L_{nn}$, to model the nonlinear system of Eq. 1:

$$\mathbf{h}_{t,\ell} = \sigma_h\left(Q\mathbf{h}_{t,\ell-1} + W\mathbf{x}_{t,\ell}\right), \quad \mathbf{y}_{t,\ell} = \sigma_y\left(V\mathbf{h}_{t,\ell}\right) \tag{2}$$

where $\mathbf{x}_{t,\ell} \in \mathbb{R}^{d_x}$, $\mathbf{h}_{t,\ell} \in \mathbb{R}^{d_h}$, and $\mathbf{y}_{t,\ell} \in \mathbb{R}^{d_y}$ are the RNN input, RNN state in the hidden layer, and the RNN output, respectively, $t = 1, \ldots, T$ and $\ell = 1, \ldots, L_{nn}$. We use $W \in \mathbb{R}^{d_h\times d_x}$, $Q \in \mathbb{R}^{d_h\times d_h}$, and $V \in \mathbb{R}^{d_y\times d_h}$ to denote the weight matrices associated with the input layer, the hidden layer, and the output layer, respectively. The element-wise nonlinear activation functions in the hidden and output layers are denoted by σ_h and σ_y, respectively. Without loss of generality, the development of RNN models are based on the following standard assumptions: 1) the RNN inputs are bounded by $\left|\mathbf{x}_{t,\ell}\right| \le B_X$, for all $t = 1, \ldots, T$ and $\ell = 1, \ldots, L_{nn}$, 2) the Frobenius norms of the RNN weight matrices are bounded as follows: $\|W\|_F \le B_{W,F}, \|Q\|_F \le B_{Q,F}, \|V\|_F \le B_{V,F}$, and 3) σ_h is a 1-Lipschitz continuous and positive-homogeneous activation function, i.e., $\sigma_h(\alpha z) = \alpha\sigma_h(z)$ for all $\alpha \ge 0$ and $z \in \mathbb{R}$. The mean squared error (MSE) is considered as the loss function $L(h(\mathbf{x}), \bar{\mathbf{y}})$ in this work, where $\bar{\mathbf{y}}$ is the true output, and $h(\cdot)$ represents the RNN model from a hypothesis class $\mathcal{H}$ that predicts the output $\mathbf{y} \in \mathbb{R}^{d_y}$ for the input $\mathbf{x} \in \mathbb{R}^{d_x}$. Since the RNN model is trained using a dataset of bounded states and inputs for the nonlinear system of Eq. 1, the locally Lipschitz property holds for the MSE loss function, i.e., the following inequality holds for all $|\mathbf{y}_\ell|, |\bar{\mathbf{y}}_\ell| \le r_\ell$, $\ell = 1, \ldots, L_{nn}$.

$$\left|L\left(\mathbf{y}_\ell, \bar{\mathbf{y}}_\ell\right) - L\left(\mathbf{y}'_\ell, \bar{\mathbf{y}}_\ell\right)\right| \le L_r\left|\mathbf{y}'_\ell - \mathbf{y}_\ell\right| \tag{3}$$

where $r_\ell > 0$ represents the upper bound of $|\mathbf{y}_\ell|$ and $|\bar{\mathbf{y}}_\ell|$, and L_r denotes the local Lipschitz constant.

Generalization error of online learning RNNs using i.i.d. data samples

Generalization error is used in statistical learning theory to measure how accurately a neural network model learned from training data can generalize to new data that has not been seen previously. In this section, we first consider a special case of switched nonlinear systems, where the system dynamics of Eq. 1 does not vary over time, and thus, Eq. 1 can be simplified to the following state-space model:

$$\dot{x} = F(x, u) := f(x) + g(x)u \tag{4}$$

The nonlinear system of Eq. 4 is assumed to have multiple steady-states x_{s_k} under $u = u_s$ (i.e., $\dot{x}_{s_k} = f(x_{s_k}) + g(x_{s_k})u_s \equiv 0$), where $k \in \psi = \{1, 2, \ldots, p\}$ and p is the number of steady-states. The nonlinear system of Eq. 4 is switched between different steady-states and is said to operate in mode k, i.e., $t \in \left[t^{in}_k, t^{out}_k\right)$, if it is required to stabilize at the steady-state x_{s_k}, $\forall k \in \psi$. In this case, online RNN models are developed using real-time process data drawn i.i.d. from the same system of Eq. 4.

Consider a set of T labeled samples collected in T rounds (X_1, Y_1), ..., (X_T, Y_T) that are drawn i.i.d. from a distribution $\mathcal{D}$. $Z_t = (X_t, Y_t) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, $t = 1, \ldots, T$, is the pair of input and output data samples, where $\mathcal{X}$ and $\mathcal{Y}$ denote the input space and the output space, respectively. To simplify the notation, we use $\mathbf{Z}^m_n$ to denote a sequence of samples $Z_n, Z_{n+1}, \ldots, Z_m$. Given a hypothesis class $\mathcal{H}$ of RNN models that map $\mathcal{X}$ to $\mathcal{Y}$, and a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ from a class of loss functions $\mathcal{F}$ associated to $\mathcal{H}$, i.e., $\mathcal{F} = \{(x, y) \to L(h(x), y) : h \in \mathcal{H}\}$, in each round $t = 1, \ldots, T$. Starting with an initial hypothesis $h_1 \in \mathcal{H}$, an online algorithm $\mathcal{A}$ generates a hypothesis h_{t+1} after processing the sample (X_t, Y_t). We use $L(h_t, Z_t)$ to represent $L(h_t(X_t), Y_t)$ for any $Z_t = (X_t, Y_t) \in \mathcal{Z}$. The aim of the online algorithm $\mathcal{A}$ is to minimize the following regret after T rounds (Rakhlin et al. (2015)):

$$\mathrm{Reg}_{\mathcal{A}}(T) = \sum_{t=1}^{T} L(h_t, Z_t) - \sum_{t=1}^{T} L(h^\star, Z_t) \tag{5}$$

where $h^\star$ is the best model achieving the minimum cumulative loss from a hypothesis class $\mathcal{H}$ in hindsight after all samples are received, i.e., $h^\star = \underset{h\in\mathcal{H}}{\arg\min}\ \sum_{t=1}^{T} L(h, Z_t)$. The generalization error of $h \in \mathcal{H}$ in online learning is defined as the expectation of L at a new data point:

$$R(h) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}[L(h(X), Y)] \tag{6}$$

Following the idea in Kuznetsov and Mohri (2016), Lemma 1 develops a generalization error bound for online learning models using i.i.d. training samples drawn from the nonlinear system of Eq. 4.

Lemma 1. *Given a set of labeled samples $\mathbf{Z}^T_1 = Z_1, \ldots, Z_T$ drawn i.i.d. from a distribution $\mathcal{D}$, and a loss function $L(\cdot, \cdot)$ that is convex with respect to its first argument and is bounded by M for some $M \ge 0$, let $h_1, \ldots, h_T$ be a sequence of hypotheses from the hypothesis class $\mathcal{H}$ generated by an online algorithm $\mathcal{A}$ processing samples $\mathbf{Z}^T_1$ sequentially. Using the ensemble of online learning models, the following inequalities hold, with probability at least $1 - \delta$ for any $\delta > 0$, for the hypothesis $h = \sum_{t=1}^{T} \lambda_t h_t$, where $\boldsymbol{\lambda} = (\lambda_1\ \ldots\ \lambda_T)$ is a weight vector bounded by a unit simplex, i.e., $\Omega_T =: \{\boldsymbol{\lambda} \in \mathbb{R}^T \mid \sum_{t=1}^{T} \lambda_t = 1$ and $\lambda_t \ge 0$ for $t = 1, \ldots, T\}$, and $h^\star$ be the optimal hypothesis from a hypothesis class $\mathcal{H}$.*

$$R(h) \le \sum_{t=1}^{T} \lambda_t L(h_t, Z_t) + M|\boldsymbol{\lambda}|\sqrt{2\log\frac{1}{\delta}} \tag{7}$$

$$\begin{aligned} R(h) \le& \frac{\mathrm{Reg}_{\mathcal{A}}(T)}{T} + \sum_{t=1}^{T} \lambda_t L(h^\star, Z_t) \\ &+ \sum_{t=1}^{T} M|\lambda_t - \frac{1}{T}| + M|\boldsymbol{\lambda}|\sqrt{2\log\frac{1}{\delta}} \end{aligned} \tag{8}$$

It is noted from Eq. 7 that the generalization error bound for an ensemble hypothesis $h=\sum_{t=1}^{T}\lambda_t h_t$ consists of two terms. The first term in the RHS of Eq. 7 represents the cumulative loss in T rounds, and the second term is an error function associated with the bound M of the loss function, the weight vector $\boldsymbol{\lambda}$, and the confidence δ. Eq. 8 is derived to connect regret with generalization error using the definition of regret in Eq. 5. Specifically, the generalization error can be bounded by regret (the first term), the loss suffered by the optimal model $h^{\star}$ (the second term), and the error functions with respect to $M,\boldsymbol{\lambda},\delta$, and T (the third and last terms). The third and last terms in the RHS of Eq. 8 are readily known once a weight vector $\boldsymbol{\lambda}$ and a confidence level δ are chosen. Therefore, if an online algorithm $\mathcal{A}$ ensures that its regret is a sublinear function of T, i.e., $\mathrm{Reg}_{\mathcal{A}}(T)=O(\sqrt{T})$, the regret term $\frac{\mathrm{Reg}_{\mathcal{A}}(T)}{T}$ in Eq. 8 converges to zero as $T\to\infty$, and the loss suffered by an ensemble hypothesis $h=\sum_{t=1}^{T}\lambda_t h_t$ is sufficiently close to the minimum loss achieved by the optimal hypothesis $h^{\star}$ using the entire dataset $\mathbf{Z}_1^T$.

Remark 1. *To achieve a better generalization performance in practice, the weight vector $\boldsymbol{\lambda}$ can be optimized for the sequence of hypotheses $h_1,\ldots,h_T$ obtained using the online algorithm $\mathcal{A}$. Specifically, we can choose a weight vector $\boldsymbol{\lambda}$ by solving the following optimization problem:*

$$\min_{\boldsymbol{\lambda}\in\Omega_T}\sum_{t=1}^{T}\lambda_t L(h_t,Z_t) \quad s.t.\ \sum_{t=1}^{T}|\lambda_t-\frac{1}{T}|\leq\alpha \tag{9}$$

where $\alpha\geq 0$ is a hyperparameter that constrains the difference between λ_t and $1/T$, and can be predetermined through a validation process. The final ensemble hypothesis is developed as $h=\sum_{t=1}^{T}\lambda_t h_t$.

Generalization error of online learning RNNs using non-i.i.d. data samples

In this section, we consider the switched systems of Eq. 1, where the system is switched between different modes with time-varying system dynamics. Since the i.i.d. assumption on training samples does not hold for the switched system of Eq. 1, the generalization error bound developed for the i.i.d. case in the previous section cannot be applied directly to the general setting of non-i.i.d. stochastic processes. Similarly to the procedure of online learning using i.i.d. data samples, we receive non-i.i.d. data samples (X_1,Y_1), …, (X_T,Y_T) in T rounds from the switched system of Eq. 1 in a sequential order. The aim of the learner is to choose a hypothesis h from the hypothesis class $\mathcal{H}$ to minimize the regret defined in Eq. 5. Given a hypothesis $h\in\mathcal{H}$ with a new data point (X_{T+1},Y_{T+1}) conditioned on the past data collected in T rounds, the generalization error is given by (Kuznetsov and Mohri (2016)):

$$R_{T+1}\left(h,\mathbf{Z}_1^T\right)=\mathbb{E}\left[L\left(h\left(X_{T+1}\right),Y_{T+1}\right)\mid\mathbf{Z}_1^T\right] \tag{10}$$

The following lemma establishes the generalization error bound for the ensemble of online learning hypotheses using non-i.i.d. training samples.

Lemma 2. *[Kuznetsov and Mohri (2016)] Given a sequence of non-i.i.d. training samples $\mathbf{Z}_1^T=Z_1,\ldots,Z_T$ drawn from the switched system of Eq. 1, $h_1,\ldots,h_T\in\mathcal{H}$ are the sequence of hypotheses developed using an online algorithm $\mathcal{A}$, a bounded loss function and a weight vector $\boldsymbol{\lambda}$ that satisfy the conditions in Theorem 1. Then, for any $\delta>0$, the following inequalities hold with probability at least $1-\delta$ for $h=\sum_{t=1}^{T}\lambda_t h_{t+1}$:*

$$R_{T+1}\left(h,\mathbf{Z}_1^T\right)\leq\sum_{t=1}^{T}\lambda_t L\left(h_{t+1},Z_{t+1}\right)+\mathrm{disc}(\boldsymbol{\lambda})+M|\boldsymbol{\lambda}|\sqrt{2\log\frac{1}{\delta}} \tag{11}$$

$$\begin{aligned}R_{T+1}\left(h,\mathbf{Z}_1^T\right)\leq&\frac{\mathrm{Reg}_{\mathcal{A}}(T)}{T}+\sum_{t=1}^{T}\lambda_t L\left(h^{\star},Z_{t+1}\right)+\mathrm{disc}(\boldsymbol{\lambda})\\&+\sum_{t=1}^{T}M|\lambda_t-\frac{1}{T}|+M|\boldsymbol{\lambda}|\sqrt{2\log\frac{1}{\delta}}\end{aligned} \tag{12}$$

Compared to the generalization error bound for the i.i.d. case in Eqs. 7-8, the generalization error bound of Eq. 11 includes an additional term $\mathrm{disc}(\boldsymbol{\lambda})$, which represents the discrepancy between target and sample distributions, and is defined as follows.

$$\mathrm{disc}(\boldsymbol{\lambda})=\sup_{h_t\in\mathcal{H}}\left|\sum_{t=1}^{T}\lambda_t\left(R_{T+1}\left(h_{t+1},\mathbf{Z}_1^T\right)-R_{t+1}\left(h_{t+1},\mathbf{Z}_1^t\right)\right)\right| \tag{13}$$

The discrepancy term characterizes the variation of data distributions due to time-varying disturbances and model uncertainty. Therefore, in addition to the optimization of the weight parameters $\boldsymbol{\lambda}$ and the convergence of regret, we also need to obtain an upper bound for the discrepancy term of Eq. 13 to ensure the boundedness of the generalization error of Eq. 12. The next lemma derives an upper bound for the discrepancy term following the results of Lemma 7 in Kuznetsov and Mohri (2016) and Theorem 2 in Kuznetsov and Mohri (2020):

Lemma 3. *Given a class of loss functions $\mathcal{F}$ and a sequence of labeled samples $\mathbf{Z}_1^T=Z_1,\ldots,Z_T$, for any $\delta>0$, the following inequality holds with probability at least $1-\delta$:*

$$\begin{aligned}\mathrm{disc}(\boldsymbol{\lambda})\leq&\widehat{\mathrm{disc}}_{\mathcal{H}}(\boldsymbol{\lambda})+\Lambda+|\boldsymbol{\lambda}|+6M\sqrt{\pi\log T}\mathcal{R}_T^{seq}(\mathcal{F})\\&+M|\boldsymbol{\lambda}|\sqrt{2\log\frac{1}{\delta}}\end{aligned} \tag{14}$$

where $\Lambda:=\inf_{\bar{h}^{\star}\in\mathcal{H}}\mathbb{E}\left[\left(L_r\left|Z_{T+1}-\bar{h}^{\star}\left(X_{T+1}\right)\right|\right)\mid\mathbf{Z}_1^T\right]$, $\widehat{\mathrm{disc}}_{\mathcal{H}}(\boldsymbol{\lambda}):=\sup_{\bar{h},h_t\in\mathcal{H}}\left|\sum_{t=1}^{T}\lambda_t\left(L\left(h_{t+1}\left(X_{T+1}\right),\bar{h}\left(X_{T+1}\right)\right)-L\left(h_{t+1},Z_{t+1}\right)\right)\right|$ can be estimated by the samples, and $\mathcal{R}_T^{seq}(\mathcal{F})$ denotes the sequential Rademacher complexity of the hypothesis class $\mathcal{F}$.

Therefore, it remains to show that there exists an upper bound for the sequential Rademacher complexity term $\mathcal{R}_T^{seq}(\mathcal{F})$. Since non-i.i.d. samples in online learning are sequentially dependent, sequential Rademacher complexity is often used to measure the richness of RNN hypothesis class while capturing sequential dependence. The definition of sequential Rademacher complexity is given below.

Definition 1. *Given a $\mathcal{Z}$-valued binary tree $\mathbf{z}$ with depth T and a function class $\mathcal{G}$ that maps from $\mathcal{Z}$ to $\mathbb{R}$, the sequential Rademacher complexity of $\mathcal{G}$ on $\mathbf{z}$ is defined as follows:*

$$\mathcal{R}_T^{seq}(\mathcal{G},\mathbf{z})=\mathbb{E}\left[\sup_{g\in\mathcal{G}}\sum_{t=1}^{T}\varepsilon_t\lambda_t g\left(z_t(\epsilon)\right)\right] \tag{15}$$

where $\boldsymbol{\epsilon} = (\varepsilon_1, \ldots, \varepsilon_{T-1})$ is a sequence of i.i.d. Rademacher random variables taken values in $\{\pm 1\}$. We use $z_t(\boldsymbol{\epsilon})$ to denote $z_t(\varepsilon_1, \ldots, \varepsilon_{t-1})$ and define $\mathcal{R}_T^{seq}(\mathcal{G}) = \sup_{\mathbf{z}} \mathcal{R}_T^{seq}(\mathcal{G}, \mathbf{z})$ as the worst-case sequential Rademacher complexity.

Let $\mathcal{F}_\ell$, $\ell = 1, \ldots, L_{nn}$, be the class of loss functions associated to the RNN hypothesis class $\mathcal{H}_\ell$ of vector-valued functions that predicts the ℓ-th RNN output $\mathbf{y}_\ell \in \mathbb{R}^{d_y}$ for the first ℓ-time-step RNN inputs $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\ell\} \in \mathbb{R}^{d_x \times \ell}$.

$$\mathcal{F}_\ell = \left\{ \mathbf{z} = (\mathbf{x}, \bar{\mathbf{y}}) \to L(h(\mathbf{x}), \bar{\mathbf{y}}) = L(h, \mathbf{z}), h \in \mathcal{H}_\ell \right\} \tag{16}$$

where $\mathbf{x}$ and $\bar{\mathbf{y}}$ denote the RNN input vector and the true output vector. Let $\mathcal{H}_{j,\ell}$, $j = 1, \ldots, d_y$, be the class of real-valued functions corresponding to the j-th component of the RNN output at the ℓ-th time step. Since the Rademacher complexity of Eq. 14 is associated with the class of loss functions, we have the following contraction inequality from Rakhlin et al. (2015) to derive an upper bound for $\mathcal{R}_T^{seq}(\mathcal{F})$ using the sequential Rademacher complexity of the hypothesis of real-valued RNN functions $\mathcal{H}_{j,\ell}$.

$$\mathcal{R}_T^{seq}(\mathcal{F}_\ell) \le 8L_r \left(1 + 4\sqrt{2}\log^{3/2}\left(eT^2\right)\right) \sum_{j=1}^{d_y} \mathcal{R}_T^{seq}(\mathcal{H}_{j,\ell}) \tag{17}$$

We next develop an upper bound for $\mathcal{R}_T^{seq}(\mathcal{H}_{j,\ell})$ following the methods in Wu et al. (2021) that peel off the weight matrices and the nonlinear activation functions layer by layer.

Lemma 4. *Given a hypothesis class $\mathcal{H}_{j,\ell}$ of real-valued functions corresponding to the j-th component of a hypothesis class $\mathcal{H}_\ell$ of vector-valued functions, and the weight matrices and activation functions satisfying the conditions in Eq. 2, the following inequality holds for the RNN model trained using a sequence of non-i.i.d. samples $\mathbf{Z}_1^T = Z_1, \ldots, Z_T$:*

$$\mathcal{R}_T^{seq}(\mathcal{H}_{j,\ell}) \le \Gamma\left(\sqrt{2\log(2)(\ell+1)} + 1\right) |\boldsymbol{\lambda}| B_X \tag{18}$$

where $\Gamma = B_{V,F} B_{W,F} \frac{(B_{Q,F})^\ell - 1}{B_{Q,F} - 1}$.

Finally, based on Lemmas 2-4, the following theorem derives a generalization error bound for the RNN models using non-i.i.d. training samples.

Theorem 1. *Let $\mathcal{F}_\ell$, $\ell = 1, \ldots, L_{nn}$, be the class of loss functions associated to the RNN hypothesis class $\mathcal{H}_\ell$ of vector-valued functions that predict the RNN output at the ℓ-th time step, and $h_1, \ldots, h_T$ be a sequence of hypotheses from the hypothesis class $\mathcal{H}_\ell$ that is generated by an online algorithm $\mathcal{A}$ and meet all the conditions in Lemmas 2-4. Given a sequence of non-i.i.d. training samples $\mathbf{Z}_1^T = Z_1, \ldots, Z_T$, and $\delta > 0$, the following inequality holds for $h = \sum_{t=1}^T \lambda_t h_{t+1}$ with probability at least $1 - \delta$:*

$$\begin{aligned} R_{T+1}(h, \mathbf{Z}_1^T) \le & \sum_{t=1}^{T} \lambda_t L(h_{t+1}, Z_{t+1}) + \widehat{\mathrm{disc}}_{\mathcal{H}}(\boldsymbol{\lambda}) + \Lambda + |\boldsymbol{\lambda}| \\ & + 2M|\boldsymbol{\lambda}|\sqrt{2\log\frac{1}{\delta}} + ML_r C_T d_y \Gamma\left(\sqrt{2\log(2)(\ell+1)} + 1\right) |\boldsymbol{\lambda}| B_X \end{aligned} \tag{19}$$

where $C_T = O\left(\sqrt{\pi \log T}\left(1 + 4\sqrt{2}\log^{3/2}\left(eT^2\right)\right)\right)$.

The weight vector $\boldsymbol{\lambda}$ can be optimized by solving Eq. 20 with $\alpha \ge 0$ (Kuznetsov and Mohri (2016)):

$$\begin{aligned} & \min_{\boldsymbol{\lambda} \in \Omega_T} \sum_{t=1}^{T} \lambda_t L(h_{t+1}, Z_{t+1}) + \widehat{\mathrm{disc}}_{\mathcal{H}}(\boldsymbol{\lambda}) \\ & \quad s.t. \sum_{t=1}^{T} |\lambda_t - \frac{1}{T}| \le \alpha, \ \lambda_T = 0 \end{aligned} \tag{20}$$

Compared to Eq. 9 for the i.i.d. case, the objective function of Eq. 20 accounts for the empirical discrepancy term $\widehat{\mathrm{disc}}_{\mathcal{H}}(\boldsymbol{\lambda})$ for non-i.i.d samples. Additionally, since the objective function of Eq. 20 depends on the sample Z_{T+1} that is unknown at $t = T$, the optimization problem of Eq. 20 includes an additional equality constraint that lets $\lambda_T = 0$.

RNN-based MPC of Switched Nonlinear Systems

In this section, we develop a Lyapunov-based MPC (LMPC) scheme using online learning RNN models for the nonlinear system of Eq. 4 with closed-loop stability analysis. Due to space limitations, we will discuss the case of online learning using i.i.d. data samples for the nonlinear system of Eq. 4 under scheduled mode transitions between multiple steady-states. To simplify the discussion of stability properties for RNN-based MPC, the RNN model of Eq. 2 is written by the following continuous-time nonlinear system:

$$\dot{\hat{x}} = F_{nn}(\hat{x}, u) \tag{21}$$

where $\hat{x} \in \mathbb{R}^n$ is the state vector of the RNN model and $u \in \mathbb{R}^{n_u}$ is the control input vector. Using the deviation variables $\hat{z}_k := \hat{x} - x_{s_k}$ for all $k \in \psi$, the RNN model of Eq. 21 can be rewritten in the deviation form of $\dot{\hat{z}}_k = F'_{nn_k}(\hat{z}_k, u)$ for each steady-state x_{s_k}. We assume that there exists a stabilizing feedback controller $u_k = \Phi_{nn_k}(z_k) \in U$ for each steady-state x_{s_k}, $k \in \psi$ such that the steady-state of the RNN model of Eq. 21 is rendered exponentially stable. This stabilizability assumption implies that for each steady-state x_{s_k}, $k \in \psi$, there exists a $\mathcal{C}^1$ control Lyapunov function $\hat{V}_k(z_k)$ such that the following inequalities hold for all z_k in an open neighborhood $\hat{D}_k$ around the origin:

$$\hat{c}_{1_k}|z_k|^2 \le \hat{V}_k(z_k) \le \hat{c}_{2_k}|z_k|^2, \tag{22a}$$

$$\frac{\partial \hat{V}_k(z_k)}{\partial z_k} F'_{nn_k}(z_k, \Phi_{nn_k}(z_k)) \le -\hat{c}_{3_k}|z_k|^2, \tag{22b}$$

$$\left| \frac{\partial \hat{V}_k(z_k)}{\partial z_k} \right| \le \hat{c}_{4_k}|z_k|, \tag{22c}$$

where $\hat{c}_{1_k}, \hat{c}_{2_k}, \hat{c}_{3_k}$, and $\hat{c}_{4_k}$, $k \in \psi$, are positive constants. A level set of Lyapunov function inside $\hat{D}_k$, i.e., $\Omega_{\hat{\rho}_k} := \left\{ z_k \in \hat{D}_k \mid \hat{V}_k(z_k) \le \hat{\rho}_k \right\}, \hat{\rho}_k > 0$, $k \in \psi$, is characterized as an estimate of the closed-loop stability region for the RNN model around each steady-state. While the closed-loop stability regions are defined with respect to the states in deviation variable form, in the following text, we will use $x \in \Omega_{\hat{\rho}_k}$ to represent that the state x is inside the stability region $\Omega_{\hat{\rho}_k}$ around the steady-state x_{s_k}, with a slight abuse of notation.

1) LMPC formulation

The LMPC using RNN models for the nonlinear system of

Eq. 4 with switching modes is formulated as follows:

$$\mathcal{J} = \min_{u\in S(\Delta)} \int_{t_q}^{t_k^{out}} L_{MPC}(\tilde{x}(t), u(t))dt \tag{23a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{nn}(\tilde{x}(t), u(t)) \tag{23b}$$

$$u(t) \in U, \; \forall\, t \in [t_q, t_k^{out}) \tag{23c}$$

$$\tilde{x}(t_q) = x(t_q) \tag{23d}$$

$$\dot{\hat{V}}_k(x(t_q) - x_{s_k}, u) \leq \dot{\hat{V}}_k(x(t_q) - x_{s_k}, \Phi_{nn_k}(x(t_q) - x_{s_k})), \text{ if } x(t_q) \in \Omega_{\hat{\rho}_k} \backslash \Omega_{\rho_{nn_k}} \tag{23e}$$

$$\hat{V}_k(\tilde{x}(t) - x_{s_k}) \leq \rho_{nn_k}, \; \forall\, t \in [t_q, t_k^{out}), \text{ if } x(t_q) \in \Omega_{\rho_{nn_k}} \tag{23f}$$

$$\hat{V}_f(\tilde{x}(t_k^{out}) - x_{s_f}) + f_e(E_P) \leq \hat{\rho}_f \tag{23g}$$

where $\tilde{x}$ and $S(\Delta)$ denote the predicted state obtained from the RNN model and the class of piecewise constant functions with sampling period Δ, respectively. The RNN-MPC of Eq. 23 is implemented with a shrinking prediction horizon, which is calculated by the difference between the switching out time t_k^{out} and the current time t_q. The objective of the RNN-MPC of Eq. 23 is to minimize the cost function of Eq. 23a and subject to the constraints of Eqs. 23b-Eq. 23g. Specifically, Eq. 23b uses the RNN model of Eq. 21 to predict the state evolution. Eq. 23c defines the input constraint. Eq. 23d defines the initial state $\tilde{x}(t_q)$ at each sampling step. Eqs. 23e-23f are the two Lyapunov-based constraints used to ensure the closed-loop stability of the nonlinear system of Eq. 4. Eq. 23g ensures that the closed-loop system state can enter the stability region $\Omega_{\hat{\rho}_f}$ of the subsequent mode f at the switching time $t = t_k^{out}$. $f_e(E_P) := \frac{\hat{c}_{4_k}\sqrt{\hat{\rho}_k}}{\sqrt{\hat{c}_{1_k}}}\sqrt{E_P} + \kappa E_P$ represents the upper bound for the difference between $\hat{V}_f(x(t_k^{out}) - x_{s_f})$ and $\hat{V}_f(\tilde{x}(t_k^{out}) - x_{s_f})$, where κ is a positive constant and E_P denotes the generalization error bound (i.e., the RHS of Eq. 11 for online learning models, where $|x - \tilde{x}| \leq \sqrt{E_P}$ holds provided that the MSE loss function is used). When Eq. 23g is removed and the shrinking prediction horizon is replaced by a fixed prediction horizon, the RNN-MPC of Eq. 23 is reduced to the LMPC design using RNN models for the system of Eq. 4 operated in a fixed mode k (i.e., the system of Eq. 4 is operated at the same steady-state for all times).

2) Integration of online learning into MPC

An RNN model is initially developed offline using the historical data for the RNN-MPC of Eq. 23 following the construction method in Wu et al. (2021), and will be updated online using real-time process data. Specifically, the initial RNN model is trained offline to approximate the nonlinear system of Eq. 4 using the data collected in the stability region around a certain steady-state. The dataset consists of time-series data generated from extensive open-loop simulation of Eq. 4 using various initial states and manipulated inputs, where $u(t) = u(t_q), \forall t \in [t_q, t_{q+1})$, $t_{q+1} := t_q + \Delta$ is implemented in a sample-and-hold fashion. The explicit Euler method is used to integrate the nonlinear system of Eq. 4 with a sufficiently small integration time step $\bar{h}_c < \Delta$. The RNN inputs are the state measurement at the current time step and the manipulated input that will be applied for the next sampling period, and the RNN output is the predicted state trajectory over one sampling period with $L_{nn} = \Delta/\bar{h}_c$. Subsequently, we apply the online-to-batch conversion by using online algorithms in the batch setting to update the RNN models. Specifically, instead of using randomly initialized weights to update the RNN model, the weights of the previous RNN model are used as the initial guess for the current RNN model. The updated RNN models are developed using only the most recent process data in a rolling window that is collected from the real-time process operation. The updated RNN models will be incorporated in RNN-MPC to replace the previous RNN models (i.e., F_{nn} in Eq. 23b) to provide a better prediction of future states. Although RNN models are updated online using real-time data of the process variables, the stability regions $\Omega_{\hat{\rho}_k}$, $k \in \psi$, characterized using the initial offline-learning RNN model remains the same for all times. As a result, feasibility is no longer guaranteed for the RNN-MPC of Eq. 23 using the updated RNN models under the controller $u_k = \Phi_{nn_k}(x - x_{s_k}) \in U$. Therefore, the online learning within RNN-MPC is implemented using the following strategy: applying the optimal solution of Eq. 23 whenever it is feasible, and applying the stabilizing controller $u_k(t) = \Phi_{nn_k}(x(t_q) - x_{s_k})$ when RNN-MPC is infeasible.

3) Closed-loop stability under RNN-MPC

Consider the nonlinear system of Eq. 4 with switching modes according to a prescribed switching schedule defined by switching times, i.e., the system is operated in the current mode k for $t \in [t_k^{in}, t_k^{out})$ and is switched to a subsequent mode f for some $k, f \in \psi$ at $t = t_k^{out} = t_f^{in}$. Closed-loop stability in the sense that the state is bounded in the stability region for all times and is ultimately bounded in the terminal set can be guaranteed for the system of Eq. 4 under the RNN-MPC of Eq. 23. In our previous work, we proved closed-loop stability for the nonlinear system at a fixed mode using offline learning RNN models (Wu et al., 2021). To prove closed-loop stability for the switched nonlinear system using online learning models, we first derive the following proposition to ensure that the closed-loop state under sample-and-hold implementation of the controller $u_k = \Phi_{nn_k}(x - x_{s_k}) \in U$ enters the stability region of the subsequent mode f in $t = t_k^{out} = t_f^{in}$.

Proposition 1. *Consider the nonlinear system of Eq. 4 and the RNN model of Eq. 21 under the controller $u_k = \Phi_{nn_k}(x - x_{s_k}) \in U$. Given $t_k^{in} \leq t < t_k^{out} = t_f^{in}$ and $(x(t_k^{in}) - x_{s_k}) \in \Omega_{\hat{\rho}_k}$, if there exist $\hat{\rho}_k$, ε_k, N_k, $\Delta > 0$, $\forall k \in \psi$, such that*

$$\hat{c}_{2_f}\left(\sqrt{\frac{\hat{\rho}_k - \varepsilon_k N_k \Delta}{\hat{c}_{1_k}}} + \left|x_{s_k} - x_{s_f}\right|\right)^2 \leq \hat{\rho}_f, \tag{24}$$

then $(x(t_f^{in}) - x_{s_f}) \in \Omega_{\hat{\rho}_f}$.

The following theorem demonstrates that under the RNN-MPC scheme, closed-loop stability of the nonlinear system of Eq. 4 can be achieved.

Theorem 2. *Consider the closed-loop system of Eq. 4 switched between different modes for some $k, f \in \psi$ under the RNN-MPC of Eq. 23, and ultimately operated in a specific terminal mode for some $z \in \psi$. Given any initial state $x(t_k^{in}) \in \Omega_{\hat{\rho}_k}$ at $t = t_k^{in}$, if the generalization error bound E_P*

is sufficiently small such that the modeling error constraint $|F(x,u) - F_{nn}(x,u))| \leq \gamma|x - x_{s_k}|$ is met, $\gamma > 0$, then for each sampling time step, closed-loop stability for the nonlinear system of Eq. 4 under the RNN-MPC of Eq. 23 is achieved with a probability at least $1-\delta$ in the sense that closed-loop state $x(t)$ is bounded in $\Omega_{\hat{\rho}_k}$ for each switching interval $t \in [t_k^{in}, t_k^{out})$ and enters the stability region $\Omega_{\hat{\rho}_f}$ of the subsequent mode f at $t = t_k^{out} = t_f^{in}$ and ultimately converges to the terminal set $\Omega_{\rho_{min_z}}$ defined by the terminal mode z.

It should be noted that the stability results in this section are derived for the nonlinear system of Eq. 4 switched between different steady-states with the process dynamics remaining unchanged. Closed-loop stability properties can be generalized to the switched nonlinear system of Eq. 1 with varying dynamics using the generalization error results in Theorem 1 for non-i.i.d. case, and appropriate assumptions and design of RNN-based MPC.

Event-triggered online machine learning within MPC

While the generalization error bounds developed for learning using i.i.d. and non-i.i.d. data samples provide theoretical accuracy guarantees for online learning RNN models, a practical implementation strategy of online learning within RNN-MPC is needed to determine when and how the online update of RNN models is triggered. In this section, we discuss two triggering mechanisms to improve the efficiency and applicability of RNN-MPC.

1) Event-trigger mechanism

Event-triggered mechanism has been applied in a number of works to reduce the frequency of the online update and adjustment of process models and control actions (Tabuada, 2007). Event-triggered control system triggers an update of control actions if a triggering condition based on state measurements is violated. In Wu et al. (2019), the event-triggered online learning is incorporated into MPC to improve RNN prediction accuracy using previously received data of closed-loop states in the presence of bounded process disturbances. Specifically, the on-line update of RNN is triggered by the violation of the following equation such that the minimal inter-event time between two triggers is not sufficiently small:

$$V(x(t)) \leq V(x(t_k)) - \varepsilon_w (t - t_k),\ t \in [t_k, t_{k+1}) \tag{25}$$

where $\varepsilon_w > 0$ is the predefined decreasing rate of the Lyapunov function value under the MPC of Eq. 23, and ρ_s is a small neighborhood around the origin. Eq. 25 implies that an RNN model needs to be updated when the the Lyapunov function value cannot decrease at the predefined rate under the stabilizing controller due to poor predictions. In Wu et al. (2019), we further demonstrate that when the MPC sampling period Δ is bounded and sufficiently small, closed-loop stability is guaranteed for the nonlinear system of Eq. 4 using event-triggered online learning of Eq. 25.

2) Error-trigger mechanism

Additionally, error-trigger mechanism can be incorporated in MPC to update RNN models online based on a moving horizon error metric $E_{rnn}(t_k)$ that indicates the prediction accuracy of RNN models at $t = t_k$ as follows (Wu et al., 2019):

$$E_{rnn}(t_k) = \sum_{i=0}^{N_b} \frac{|x_p(t_{k-i}) - x(t_{k-i})|}{|x(t_{k-i})| + \delta} \tag{26}$$

where N_b is the number of sampling periods before t_k that contribute to the quantification of the prediction error. $x_p(t_{k-i})$, $i = 0, ..., N_b$ are the predictions of the past states using RNN models, while $x(t_{k-i})$ are the past state measurements from the actual nonlinear system of Eq. 4 under the same control actions. δ is a small positive real number added to avoid the division by small numbers when $x(t_{k-i})$ approaches zero. Based on the error metric $E_{rnn}(t_k)$, we update RNN models if the accumulated error $E_{rnn}(t_k)$ exceeds the predetermined threshold E_T.

Conclusions

This work presented a summary of our recent research results on online learning of RNN models and ML-based MPC for switched nonlinear systems. The generalization error bounds for online learning RNN models using i.i.d. and non-i.i.d. data samples were first derived. Then, the online learning RNN models were incorporated into the design of MPC, with probabilistic closed-loop stability properties developed for switched nonlinear system. Finally, we discussed event-trigger and error-trigger mechanisms for practical implementation of online learning models within MPC.

References

Cesa-Bianchi, N., A. Conconi, and C. Gentile (2004). On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory 50*, 2050–2057.

Kuznetsov, V. and M. Mohri (2016). Time series prediction and online learning. In *Proceedings of the 29th Annual Conference on Learning Theory*, New York, USA, pp. 1190–1213.

Kuznetsov, V. and M. Mohri (2020). Discrepancy-based theory and algorithms for forecasting non-stationary time series. *Annals of Mathematics and Artificial Intelligence 88*, 367–399.

Rakhlin, A., K. Sridharan, and A. Tewari (2015). Online learning via sequential complexities. *Journal of Machine Learning Research 16*, 155–186.

Schwenkel, L., M. Gharbi, S. Trimpe, and C. Ebenbauer (2020). Online learning with stability guarantees: A memory-based warm starting for real-time MPC. *Automatica 122*, 109247.

Tabuada, P. (2007). Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control 52*(9), 1680–1685.

Wu, Z., D. Rincon, and P. D. Christofides (2019). Real-time adaptive machine-learning-based predictive control of nonlinear processes. *Industrial & Engineering Chemistry Research 59*, 2275–2290.

Wu, Z., D. Rincon, Q. Gu, and P. D. Christofides (2021). Statistical machine learning in model predictive control of nonlinear processes. *Mathematics 9*, 1912.

Zheng, Y., T. Zhao, X. Wang, and Z. Wu (2022). Online learning-based predictive control of crystallization processes under batch-to-batch parametric drift. *AIChE Journal 68*, e17815.