Robust Learning and Control of Time-Delay Nonlinear Systems With Deep Recurrent Koopman Operators

Minghao Han, Zhaojian Li, *Senior Member, IEEE*, Xiang Yin, *Member, IEEE*, and Xunyuan Yin

***Abstract*—In this work, we consider the problem of Koopman modeling and data-driven predictive control for a class of uncertain nonlinear systems subject to time delays. A robust deep learning-based approach–deep recurrent Koopman operator is proposed. Without requiring the knowledge of system uncertainties or information on the time delays, the proposed deep recurrent Koopman operator method is able to learn the dynamics of the nonlinear systems autonomously. A robust predictive control framework is established based on the deep Koopman operator. Conditions on the stability of the closed-loop system are presented. The proposed approach is applied to a chemical process example. The results confirm the superiority of the proposed framework as compared to baselines.**

***Index Terms*—Deep recurrent koopman operators, learning-based control, time delays, uncertain nonlinear systems.**

I. Introduction

COMPLEX industrial processes, such as chemical plants [1], oil refineries [2], and power grids [3], need advanced control strategies to achieve safe, efficient, reliable, and sustainable operation. One of the most widely used advanced control techniques is model predictive control (MPC) [4], which optimizes the future behavior of the system by using a mathematical model and current measurements. The development of MPC requires a model that can accurately capture the dynamics and constraints of the system. Typically, a first-principles model derived from physical laws and empirical correlations is used as the basis of MPC designs. However, such models can be challenging to obtain and validate for complex industrial systems exhibiting high nonlinearity.

The challenge of obtaining accurate mechanistic models has driven significant interest in the use of machine learning. In past decades, the success of machine learning, especially deep learning, has motivated its applications in system modeling and control [5]. One line of research has been focused on the training of artificial neural networks with data to approximate the dynamics of target systems [6]. The modeling problems are formulated as supervised learning problems, and neural networks are trained to infer system dynamical behaviors based on the states/measurements and control inputs. Then, based on the learned model, controllers are designed/learned based on appropriate algorithms. Additionally, research efforts have also been developed for learning-based model-free control, where control laws are directly learned out of data, e.g., reinforcement learning, adaptive dynamic programming, and iterative learning control [7].

Recently, the Koopman operator theory has attracted much attention, due to its ability to model and represent the dynamics of complex nonlinear systems in a linear form on a high-dimensional observable space [8]. The transformation of nonlinearity to linearity is a key enabler for applying linear control strategies for the analysis and control of nonlinear systems. Accordingly, practical algorithms for building Koopman linear models based on process data have been proposed, such as dynamic mode decomposition (DMD) [9] and extended dynamic mode decomposition (EDMD) [10]. DMD and EDMD encode the observables with a set of manually selected functions and solve least-squares problems to approximate the linear operator [11]. To automate the design of observable functions, deep learning was exploited, and the Deep-DMD approaches were proposed [12], [13]. Following [12], a Koopman autoencoder framework was proposed for learning Koopman models [14]. However, the conventional Koopman theory was originally developed for deterministic systems, and most of the existing Koopman-based methods have not explicitly taken into account system uncertainties/noise. As initial attempts to incorporate system uncertainties into Koopman modeling, [15] proposed a probabilistic Koopman learning framework to model and control uncertain nonlinear systems with noisy datasets. [16] proposed a probabilistic Koopman inference and control approach and applied it to power systems.

Manuscript received 2 May 2023; revised 26 August 2023; accepted 20 October 2023. Date of publication 7 November 2023; date of current version 23 February 2024. This work is supported by Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG63/22), and is partially supported by U.S. National Science Foundation Award CNS-2219488. Paper no. TII-23-1559. *(Corresponding author: Xunyuan Yin.)*

Minghao Han and Xunyuan Yin are with the School of Chemistry, Chemical Engineering, and Biotechnology, Nanyang Technological University (NTU), Singapore 637459 (e-mail: minghao.han@ntu.edu.sg; xunyuan.yin@ntu.edu.sg).

Zhaojian Li is with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: lizhaoj1@egr.msu.edu).

Xiang Yin is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yinxiang@sjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3328432.

Digital Object Identifier 10.1109/TII.2023.3328432

In the meantime, the existing few approaches have not yet addressed a crucial type of uncertainty commonly encountered during the operation of complex industrial processes, i.e., time delays. For instance, the metallurgical industry often involves processes with long and varying time delays due to the physical characteristics of the materials and the transportation of molten metals [17]. In chemical processes, lab analyzers are commonly used for obtaining concentration measurements, which also leads to significant time delays [18]. These time delays vary in nature and are unpredictable. Ignoring these delays can significantly deteriorate performance in modeling and control. These observations have highlighted the necessity of incorporating delays into learning-based Koopman modeling and control.

In this article, we present a novel learning control framework for uncertain nonlinear systems with time delays, namely deep recurrent Koopman operator (DRKO). We train a deep recurrent neural network to map the original state/measurement trajectory to a high-dimensional probabilistic distribution and learn a set of linear operators to predict the system's future dynamical behaviors. The stability of the closed-loop system is guaranteed by the proposed robust control framework, even in the presence of modeling errors. DRKO is evaluated on the modeling and control of a reactor-separator process and shows improved performance over the baseline.

The contributions of this article are summarized as follows.

1) The Koopman operator is generalized to deal with the learning and control of uncertain nonlinear systems with time delays.
2) In conjunction with linear model predictive control, the proposed pipeline can guarantee the stability of the closed-loop system in the presence of modeling errors.
3) Compared to the baseline, the proposed method demonstrates superior modeling and control performance.

II. Preliminaries

In this section, we provide a brief overview of the fundamentals of Koopman operators and describe the time-delayed uncertain nonlinear system learning problem.

A. Koopman Operator

First, we provide a brief introduction to the Koopman operator. Koopman operator theory was first proposed in [8], and it claims that a nonlinear system, e.g.,

$$x_{k+1} = f(x_k), k \in \mathbb{N} \tag{1}$$

can be converted into a linear system on an infinite-dimensional function space $\mathcal{G}$, which consists of all square-integrable real-valued functions defined on the compact domain $\mathbb{X} \subset \mathbb{R}^n$. In (1), the $x_k \in \mathbb{X}$ denotes the state vector at time instant k; $f(\cdot) : \mathbb{X} \to \mathbb{X}$ denotes the nonlinear function that characterizes the dynamics of the nonlinear system. The elements of $\mathcal{G}$ are called *observable functions* and are denoted by g. The Koopman operator $\mathcal{K} : \mathcal{G} \to \mathcal{G}$ satisfies the following equality:

$$g \circ f(x_k) = \mathcal{K}g(x_k) \tag{2}$$

where $\circ$ denotes function composition. Using the Koopman operator, the future state can be predicted as follows.

$$x_{k+1} = g^{-1}(\mathcal{K}g(x_k)). \tag{3}$$

From a practical perspective, it is typically only possible to adopt a finite-dimensional function space $\overline{\mathcal{G}} \subset \mathcal{G}$, spanned by a set of linearly independent observables $\{g | g : \mathbb{R}^n \to \mathbb{R}^h\}$, and the corresponding Koopman operator K will also be finite-dimensional. The dimensionality h of this finite observable space is typically specified by the designer/user. Although the Koopman operator was originally proposed for autonomous nonlinear systems, this concept has been extended to actuated/controlled systems in recent years [19]. For nonlinear controlled systems

$$x_{k+1} = f(x_k, u_k) \tag{4}$$

where $u_k \in \mathbb{U} \subset \mathbb{R}^m$ denotes the input vector, the Koopman operator satisfies the following condition:

$$g_x \circ f(x_k, u_k) = Ag_x(x_k) + Bg_u(x_k, u_k) \tag{5}$$

where $A \in \mathbb{R}^{h\times h}$ and $B \in \mathbb{R}^{h\times r}$ are submatrices of the Koopman operator. $g_x : \mathbb{R}^n \to \mathbb{R}^h$ and $g_u : \mathbb{R}^{n+m} \to \mathbb{R}^r$ are observables of the state and inputs, respectively. As shown in (5), for controlled systems, the future state observables $g_x(x_{k+1})$ are not only dependent on $g_x(x_k)$, but are also dependent on the input observables $g_u(x_k, u_k)$.

B. Problem Formulation

In this article, we focus on developing an efficient learning and control approach for uncertain nonlinear systems with time delays, of which the dynamics are described as follows:

$$x_{k+1} = f(x_{k-\tau_x:k}, u_{k-\tau_u:k}) + \epsilon_k \tag{6}$$

where f is an unknown nonlinear function of the past states $x_{k-\tau_x:k}$ and inputs $u_{k-\tau_u:k}$ up to unknown time delays τ_x and τ_u, respectively. $\epsilon \in \mathbb{R}^n$ denotes the stochastic noise and it is subject to unknown distribution p_ϵ.

The first step of our approach is to learn the dynamics of (6) and build a high-fidelity dynamic model in the absence of any prior knowledge of dynamics, uncertainty and time delay, utilizing a collected data set $\mathcal{D}$ that contains M trajectories with T steps of state-action pairs each, $\mathcal{D} := \{[x_k, u_k]^i_{0:T}\}_{i=1:M}$. In the next section, we will describe how to achieve this by using the proposed Koopman model and deep learning techniques.

In the second step of the proposed approach, the learning-based model is exploited to develop a predictive control scheme that can provide guaranteed closed-loop stability for nonlinear system (6). The definition of stability for the system (6) in nominal state is given as follows [20]:

Definition 1: System (6) is said to be mean-square stable (MSS) if there exists a positive constant b such that $\lim_{k\to\infty} \mathbb{E}_{x_k}\|x_k\| = 0$ holds for any initial state distribution $p(x_0)$ that $\mathbb{E}_{x_0}\|x_0\| \leq b$. If b is arbitrarily large, then the system is globally MSS.

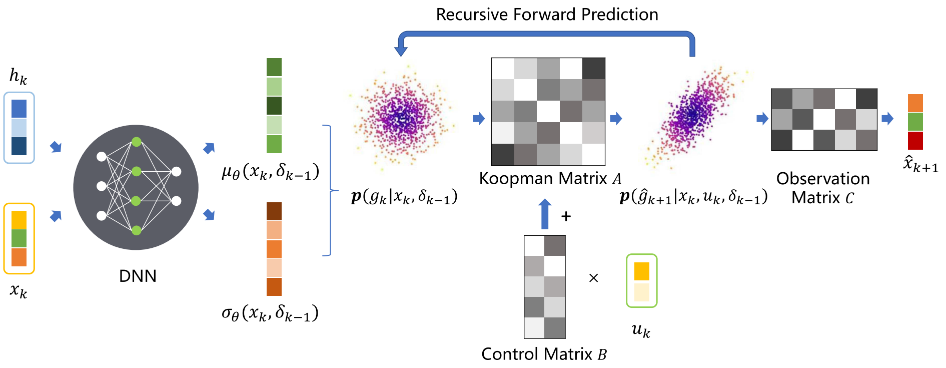

Recursive Forward Prediction
h_k
x_k
DNN
$\mu_\theta(x_k, \delta_{k-1})$
$\sigma_\theta(x_k, \delta_{k-1})$
$\boldsymbol{p}(g_k|x_k, \delta_{k-1})$
Koopman Matrix A
$\boldsymbol{p}(\hat{g}_{k+1}|x_k, u_k, \delta_{k-1})$
Observation Matrix C
$\hat{x}_{k+1}$
+
×
Control Matrix B
u_k

Fig. 1. Overview of the DRKO model pipeline. The mean and variance vectors encoded by probabilistic neural networks are used to construct the distribution of observables. The distribution is then propagated forward by the Koopman operators to predict the future distribution of observables. Finally, the observables are mapped back to the original state space with the observation matrix.

In the presence of noise, we seek to guarantee the boundedness of system (6) according to the following definition:

Definition 2 ([21]): System (6) is said to be uniformly ultimately bounded, if for positive constants b and d, there exists an instant $T(b, d)$, such that $\forall k \geq T(b, d), \mathbb{E}_{x_k}\|x_k\| < d$ holds for any initial state distribution $p(x_0)$ that $\mathbb{E}_{x_0}\|x_0\| \leq b$. d is called the uniform ultimate bound of the system. If this holds for an arbitrarily large b, then the system is globally uniformly ultimately bounded.

Remark 1: System (6) may also be viewed as a system with state order τ_x and input order τ_u. Meanwhile, the control problem we are considering in this work is also a time-delay systems control problem, in the sense that the control input can be determined based on the delayed state measurements, ensuring the stability of the closed-loop control system.

III. Deep Recurrent Koopman Operators

In this section, we elaborate on the architecture of the proposed DRKO modeling approach and elucidate how DRKO learns the dynamics of (6).

A. Model

The DRKO model consists of two building blocks: 1) a probabilistic recurrent neural network (RNN) that encodes the distribution of observables; and 2) the Koopman operator. An overview of our pipeline involving established Koopman operators is shown in Fig. 1.

1) Probabilistic RNN: To deal with both system uncertainties and unknown time delays of system (6), we present two techniques that can be leveraged to construct the observable functions, including the recurrent neural network and probabilistic encoding. Due to the existence of time delays, the state/measurement of the current time step is dependent not only on the last state and action input, but also on its historical state-input trajectories. The RNN [22] is well known for the learning and modeling of temporal sequences, and is exploited to discover the inherent dependency in the system dynamic. However, ordinary RNNs can suffer from vanishing gradients when the gradient of prediction error is calculated with respect to historical data backward in time, causing the loss of information from past data [23]. To address the potential issue of vanishing gradients in ordinary RNNs, we adopt the long-short-term memory (LSTM), which has been proven to be effective in modeling long-term dependencies [24], as the basic units of the RNN in the proposed DRKO architecture. LSTM can handle sequences with arbitrary lengths, thus it can facilitate the modeling of systems with unknown time delays. As shown in Fig. 2, the LSTM cell sequentially takes in a trajectory $\delta_{k-1} := [x_j, u_j]_{j=0:k-1}$ as input, and updates its hidden state $h_k \in \mathbb{R}^l$ and cell state $c_k \in \mathbb{R}^l$, where the dimension l is a hyperparameter and is equal to the number of LSTM cells. The hidden state at time instant k, h_k is dependent on state x_{k-1} and input u_{k-1}, and is indirectly dependent on the previous steps due to its dependency on c_{k-1}. The flow of information from past data is controlled by three trainable gates within the LSTM cell.

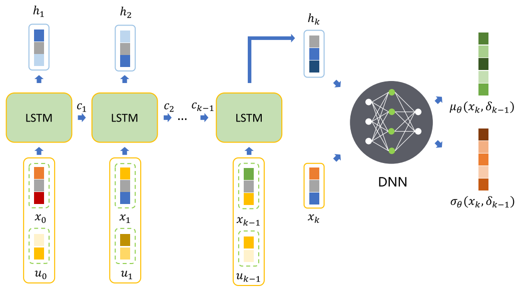


Fig. 2. Overview of the structure of the probabilistic RNN. The LSTM cells sequentially take in a trajectory $\delta_{k-1} := [x_j, u_j]_{j=0:k-1}$ as input, and updates its hidden state h_k and cell state c_k. At instant k, the hidden state h_k and state x_k are mapped to mean vector μ_k and variance vector σ_k through a neural network.

Meanwhile, we introduce probabilistic encoding of the observables to account for the uncertainties/stochasticity caused by the noise or disturbances. Instead of deploying a deterministic encoding of observables $g_k = g(x_k)$ as discussed in Section II, we propose to encode the distribution of observables $p(g_k|x_k, \delta_{k-1})$ where δ_{k-1} denotes the historical state-input trajectory up to instant $k-1$. The parameterization of this conditional distribution p is not unique. In this work, a parameterized multivariate Gaussian distribution is used to approximate the true posterior distribution, which has been widely used in probabilistic inferences [25].

Specifically, the parameterized distribution is given as $p_\theta(g_k|x_k, \delta_{k-1}) = \mathcal{N}(\mu_\theta(x_k, \delta_{k-1}), \sigma_\theta(x_k, \delta_{k-1}))$, where the mean vector $\mu_\theta(x_k, \delta_{k-1})$ and the diagonal covariance vector $\sigma_\theta(x_k, \delta_{k-1})$ are encoded using a dense neural network. Their dependence on the historical trajectory δ_{k-1} is established through the hidden state h_k. The parameter θ includes all trainable parameters in neural networks. The mapping relations in the model are shown in Fig. 2. In the following, we use μ_k and σ_k as shorthand for $\mu_\theta(x_k, \delta_{k-1})$ and $\sigma_\theta(x_k, \delta_{k-1})$, respectively.

Remark 2: In DRKO, a certain form of parameterized distribution needs to be employed to approximate the unknown distribution of the observables. In this work, we choose to use the Gaussian distribution. The reasons are mainly threefold amnd are as follows.

1) Gaussian distribution is one of the most commonly used parameterized distributions in machine learning-related research.
2) By using Gaussian distribution, a closed-form expression for linear transformation can be obtained, which streamlines the development and analysis of the proposed method.
3) By leveraging Gaussian distribution to approximate the distribution of the observables and subsequently learning a Koopman operator, more robust models can be obtained and enhanced predictability can be achieved.

Remark 3: In DRKO, the LSTM model is introduced to encode the current and previous states and inputs into a compact latent state vector h_k, which is crucial for the prediction of future states of time-delay systems. As compared to the existing approaches, our pipeline can accurately learn the dynamical behaviors of the system (6) and appropriately control the system, without requiring any prior knowledge of the dynamic parameters, noise statistics/characteristics, nonlinearity types, or time delays of the underlying nonlinear system.

2) Koopman Operators: The encoded distribution is then propagated forward by using the Koopman operators A and B as

$$g_{k+1} = Ag_k + Bu_k \tag{7a}$$

$$\hat{x}_k = Cg_k \tag{7b}$$

$$g_k \sim p_\theta(g_k|x_k, \delta_{k-1}). \tag{7c}$$

A decoder is also needed to reconstruct and predict future states. To preserve the convexity of the resulting predictive control problem, an observation matrix C is introduced as a decoder to map the observable g to the predicted state $\hat{x}$. For controlled systems, it might be possible to construct an exact decoupled linear representation in the form of (7a), yet there is no theoretical guarantee. While we cannot ensure that the exact decoupled representation can be established for any controlled nonlinear systems, from a practical perspective, it is still feasible to find an optimal decoupled approximation in a finite-dimensional space with the utilization of LSTM and dense neural networks, despite the inevitable modeling error. In Section IV, the modeling residual term r_k caused by using an approximated Koopman model and its effect on the closed-loop system has been taken into account. Accordingly, the effect of the model error on the performance of the resulting closed-loop control system is investigated through theoretical analysis and the stability of the closed-loop system is proved in the next section.

By utilizing the Koopman operators, the closed-form distribution of future observables g and predicted states $\hat{x}$ can be derived as the following closed-form:

$$p_\theta(g_{k+1}) = \mathcal{N}(A\mu_k + Bu_k, A\Lambda(\sigma_k)A^{\mathrm{T}}) \tag{8}$$

where $\Lambda(\sigma_k) := \mathrm{Diag}\{\sigma_k\}$ denotes the diagonal matrix constructed with σ_k being diagonal elements. Likewise, the distribution of $\hat{x}_{k+1}$ can also be derived

$$p_\theta(\hat{x}_{k+1}) = \mathcal{N}\left(C\left(A\mu_k + Bu_k\right), CA\Lambda\left(\sigma_k\right)A^{\mathrm{T}}C^{\mathrm{T}}\right). \tag{9}$$

This way, the best guess/prediction of the future state given by the model is $\mathbb{E}_p\hat{x}_{k+1} = C(A\mu_k + Bu_k)$. As shown in (8), the Koopman operators A and B are used to propagate both the mean vector and the covariance matrix.

B. Optimization

In this subsection, we elaborate on the implementation details of the model training process.

Given a dataset $\mathcal{D}$, the goal of DRKO learning is to find a set of parameters θ, A, B, C that minimize the multistep-ahead prediction error characterized by

$$\mathcal{L}(\theta, A, B, C) = L_{\text{re}} + L_{\text{linear}} \tag{10a}$$

$$L_{\text{re}} = \mathbb{E}_{\mathcal{D}} \sum_{j=0}^{H} \mathbb{E}_g \|x_{k+j} - Cg_{k+j|k}\|_2^2 \tag{10b}$$

$$L_{\text{linear}} = \mathbb{E}_{\mathcal{D}} \sum_{j=0}^{H} \mathbb{E}_g \|g_{k+j|k+j} - g_{k+j|k}\|_2^2 \tag{10c}$$

$$g_{k+j+1|k} = Ag_{k+j|k} + Bu_{k+j} \tag{10d}$$

$$g_{k+j|k+j} \sim p_\theta(g|x_{k+j}, \delta_{k+j-1}) \tag{10e}$$

$$j \in [0, 1, \ldots, H] \tag{10f}$$

where H denotes the prediction horizon. $g_{k+j|k}$ denotes the predicted observable at instant $k+j$ given the information available at time instant k (x_k and δ_{k-1}) and the subsequent control inputs $u_{k:k+j}$. The reconstruction loss L_{re} encourages the model to find the optimal parameters to minimize the prediction error of the states, while the linear evolution loss L_{linear} regulates the model to learn a proper observable space. By minimizing the multistep prediction error in (10a), the model is encouraged to capture crucial dependencies in the long-term behavior of the system [15].

Instead of maximizing the posterior log-likelihood function, we perform reparameterization to calculate the loss function and solve the optimization problem in (10), which has been proven to be capable of improving the stability of the training process [15]. The reparameterization injects a Gaussian noise vector $\epsilon_i \sim \mathcal{N}(0, I)$ to approximate the expectation of the observables. Consequently, the optimization problem is formulated as follows:

$$\min_{\theta,A,B,C} L_{\text{re}} + L_{\text{linear}} \tag{11a}$$

$$\text{s.t. } L_{\text{re}} = \mathbb{E}_{\mathcal{D}} \sum_{j=0}^{H} \sum_{i=0}^{N} \|x_{k+j} - C g^i_{k+j|k}\|_2^2 \tag{11b}$$

$$L_{\text{linear}} = \mathbb{E}_{\mathcal{D}} \sum_{j=0}^{H} \sum_{i=0}^{N} \|\mu_\theta(x_{k+j}, \delta_{k+j-1}) - g^i_{k+j|k}\|_2^2 \tag{11c}$$

$$g^i_{k+j+1|k} = A g^i_{k+j|k} + B u_{k+j} \tag{11d}$$

$$g^i_{k|k} = \mu_\theta(x_k, \delta_{k-1}) + \epsilon_i \sigma_\theta(x_k, \delta_{k-1}) \tag{11e}$$

$$\epsilon_i \sim \mathcal{N}(0, I), i \in [1, \ldots, N]\,. \tag{11f}$$

In (11f), each noise vector ϵ_i is sampled from the unit normal distribution $\mathcal{N}(0, I)$, and each can be transformed into a seed observable at k as in (11e). Each seed observable is then propagated forward in (11d) to predict future observables and states, which generates a seed trajectory. The expectation of prediction error can be estimated by calculating the mean prediction errors on all the seed trajectories. The estimated value converges to the true value as N goes to infinity. In practice, the generation and propagation of random noise ϵ_i can be achieved by using built-in functions in the Tensorflow Probability [26] package.

Furthermore, we introduce an entropy constraint on the distribution of observables in (11) to improve the robustness of the learned DRKO model. Since DRKO is a probabilistic model, the entropy/uncertainty of its predictions naturally decreases as the training process progresses and the prediction accuracy increases. However, if the dataset used for training does not sufficiently cover the state-input space, then the model may become overly confident in its predictions. To address this issue, we impose a constraint on the average of the entropy of the encoding such that it is greater than a minimum threshold $\mathcal{H}$, i.e., $\mathbb{E}_{\mathcal{D}} - \log p_\theta(g_k|x_k, \delta_{k-1}) \geq \mathcal{H}$. Then, a Lagrange multiplier is introduced to convert the constrained optimization problem into an unconstrained one. During training, the Lagrange multiplier value is updated by gradient descent.

Remark 4: The DRKO aims to address a general class of uncertain nonlinear systems with time delays as shown in (6). Meanwhile, when the proposed method is applied to larger-scale systems with a larger number of state variables, the training process can be more challenging. Under such circumstances, more samples may be needed and more efficient techniques may be adopted to effectively learn a good robust Koopman model.

IV. DRKO-Based Predictive Control

In this section, a predictive control scheme with a performance guarantee is proposed based on the learned DRKO.

A. Controller Design

First, in the DRKO model, the expectation of the observables is represented by the vector $\mu_k := \mathbb{E}_{g_k} g_k$. In the nominal setting, the dynamics of $\hat{\mu}_k$ can be described by the Koopman operators as follows:

$$\hat{\mu}_{k+1} = A\hat{\mu}_k + Bc_k \tag{12a}$$

$$\mathbb{E}\hat{x}_k = C\hat{\mu}_k \tag{12b}$$

where $c \in \mathbb{U}$ denotes the nominal control input and $\hat{\mu} \in \mathbb{R}^h$ denotes the nominal mean vector. The initial mean vector $\hat{\mu}_1 = \mu_\theta(x_1, \delta_0)$ is encoded by the probabilistic neural network. For this nominal system, the linear model predictive controller formulates and solves the following optimization problem:

$$V^*(\hat{\mu}_k) = \min_{c_{0:H-1}} \sum_{j=0}^{H-1} \left(\|C\hat{\mu}_{k+j}\|_Q^2 + \|c_{k+j}\|_R^2\right) + \|\hat{\mu}_{k+H}\|_P^2 \tag{13a}$$

$$\text{s.t. } \hat{\mu}_{k+j+1} = A\hat{\mu}_{k+j} + Bc_{k+j}, \tag{13b}$$

$$c_{k+j} \in \mathbb{U} \tag{13c}$$

where Q, R, and P are the positive definite weighting matrices, and $\|\cdot\|_W^2$ denotes the weighted Euclidean norm with W being a positive definite matrix.

To ensure the stability of the nominal system, the terminal weighting matrix needs to be properly designed. Specifically, P is required to satisfy the following condition [4]:

$$\|A_K\hat{\mu}_k\|_P^2 - \|\hat{\mu}_k\|_P^2 \leq -\|C\hat{\mu}_k\|_Q^2 - \|K\hat{\mu}_k\|_R^2 \tag{14}$$

where K is a state feedback controller that stabilizes the nominal system, i.e., the eigenvalues of the closed-loop transition matrix $A_K := A + BK$ are all located in the unit circle. In this work, we find P and K by solving the discrete Riccati equation

$$P = C^{\mathrm{T}}QC + A^{\mathrm{T}}PA - A^{\mathrm{T}}PB(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA \tag{15a}$$

$$K = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA. \tag{15b}$$

In the presence of noise/disturbances, the dynamic of the actual system (6) may deviate from the predictions provided by the nominal system (12), even if the learned model is globally optimal. In such cases, the dynamics of the observables can be characterized by the following equation:

$$\mu_{k+1} = \mu_\theta(f(x_k, u_k, \delta_{k-1}) + \epsilon_k, \delta_{k-1}) \tag{16}$$

and the dynamics of the actual mean vector can be described by the following stochastic Koopman model:

$$\mu_{k+1} = A\mu_k + Bu_k + v(\epsilon_k) \tag{17}$$

where $v(\epsilon_k) := \mu_\theta(f(x_k, u_k, \delta_{k-1}) + \epsilon_k, \delta_{k-1}) - \mu_\theta(f(x_k, u_k, \delta_{k-1}), \delta_{k-1})$ denotes effect of the unknown noise ϵ_k on the dynamics of mean vector.

To compensate for the error caused by the noise/disturbances, we exploit the state-feedback controller to drive the actual system state trajectory towards the nominal system. At each time instant, the actual control output is computed as follows:

$$u_k = c_k^* + K(\mu_\theta(x_k, \delta_{k-1}) - \hat{\mu}_k). \tag{18}$$

The dynamic of the error system between the actual and nominal systems, denoted by $e_k := \mu_k - \hat{\mu}_k$, is given as

$$e_{k+1} = A_K e_k + v(\epsilon_k). \tag{19}$$

In what follows, the error dynamics in (19) are analyzed to ensure the closed-loop performance of (6).

B. Stability With Optimal Learning

In this subsection, we conduct closed-loop performance analysis under the case scenario when the global optimal parameters are learned, i.e., the optimal Koopman operators A^*, B^*, C^* and probabilistic NN's parameter θ^* are available. To conduct the theoretical analysis, the following assumptions are needed.

Assumption 1: The probabilistic NN μ_θ is Lipschitz continuous, i.e. $\|\mu_\theta(x,\delta) - \mu_\theta(x+y,\delta)\| \le L\|y\|, \forall y \in R^{n+m+l}$.

Assumption 2: The energy of the system noise is bounded, i.e., there exists a finite constant ζ such that $\mathbb{E}\|\epsilon\| \le \zeta$.

Accordingly, we can obtain the following result.

Proposition 1: If Assmuptions 1 and 2 are satisfied, then system (6) controlled by the Robust MPC controller (13) and (18) is uniformly ultimately bounded with finite bound $\frac{\alpha\nu L\zeta}{1-\alpha}$, where $\alpha = \|A_K\|$, and and $\nu := \|C\|$.

Proof: The proof is divided into two parts: 1) the stability of the nominal system; and 2) the stability of the error system (19). First, the stability of the nominal system (12) is demonstrated. Solving the MPC problem (13) at instant k, one obtains the optimal control sequence

$$\{c_{k|k}^*, c_{k+1|k}^*, \ldots, c_{k+H-1|k}^*\} \tag{20}$$

and the resulting optimal mean vector trajectory

$$\{\hat{\mu}_{k|k}^*, \hat{\mu}_{k+1|k}^*, \ldots, \hat{\mu}_{k+H-1|k}^*, \hat{\mu}_{k+H|k}^*\}. \tag{21}$$

By appending the control signal produced by the feedback controller $K\hat{\mu}_{k+H|k}^*$ to (20), a suboptimal solution at next sampling instant $k+1$ is given as

$$\{c_{k|k}^*, c_{k+1|k}^*, \ldots, c_{k+H-1|k}^*, K\hat{\mu}_{k+H|k}^*\} \tag{22}$$

and

$$\{\hat{\mu}_{k|k}^*, \hat{\mu}_{k+1|k}^*, \ldots, \hat{\mu}_{k+H-1|k}^*, \hat{\mu}_{k+H|k}^*, A_K\hat{\mu}_{k+H|k}^*\}. \tag{23}$$

Based on this suboptimal solution, we can prove that the optimal value function $V^*(\hat{\mu}_k)$ is a control Lyapunov function that decreases along the trajectory. Since (22) and (23) are suboptimal, one has

$$V^*(\hat{\mu}_{k+1}) = \sum_{j=1}^{H} q\left(\hat{\mu}_{k+j}^*, c_{k+j}^*\right) + p(\hat{\mu}_{k+H+1}^*) \tag{24a}$$

$$\le \sum_{j=1}^{H-1} q(\hat{\mu}_{k+j|k}^*, c_{k+j|k}^*) + q(\hat{\mu}_{k+H|k}^*, K\hat{\mu}_{k+H|k}^*) \tag{24b}$$

$$+ p(A_K\hat{\mu}_{k+H|k}^*) \tag{24c}$$

$$= V^*(\hat{\mu}_k) + q(\hat{\mu}_{k+H|k}^*, K\hat{\mu}_{k+H|k}^*) + p(A_K\hat{\mu}_{k+H|k}^*) \tag{24d}$$

$$- q(\hat{\mu}_{k|k}^*, c_{k|k}^*) - p(\hat{\mu}_{k+H|k}^*) \tag{24e}$$

where $q(\hat{\mu}_k, c_k) = \|C\hat{\mu}_k\|_Q^2 + \|c_k\|_R^2$ denotes the stage cost, and $p(\hat{\mu}_k) = \|\hat{\mu}_k\|_P^2$ denotes the terminal cost. Note that K is a stabilizing state feedback controller gain as designed by (15), thus it follows that

$$q\left(\hat{\mu}_{k+H|k}^*, K\hat{\mu}_{k+H|k}^*\right) + p\left(A_K\hat{\mu}_{k+H|k}^*\right) - p\left(\hat{\mu}_{k+H|k}^*\right) \le 0. \tag{25}$$

Therefore, it is obtained that

$$V^*(\hat{\mu}_{k+1}) - V^*(\hat{\mu}_t) \le -q(\hat{\mu}_{k|k}^*, c_{k|k}^*) \tag{26}$$

and the optimal value function $V^*(\cdot)$ is a valid Lyapunov function. Therefore, the expectation of the nominal state $\mathbb{E}\hat{x_k} = C\hat{\mu}_k$ converges to zero as $t \to \infty$; i.e., the nominal state is mean-square stable according to Definition 1.

In the second part of this proof, let us consider the evolution of the error system in (19) from the initial time instance 1 to k governed by $e_{k+1} = A_K v(\epsilon_k) + A_K^2 v(\epsilon_{k-1}) + \cdots + A_K^k v(\epsilon_1) + A_K^k e_1$. Since $\hat{\mu}_1 = \mu_\theta(x_1, h_1)$, the initial error e_1 equals zero. Then, the l_2 norm of the error state is bounded as follows:

$$\|e_{k+1}\| = \|A_K v(\epsilon_k) + A_K^2 v(\epsilon_{k-1}) + \cdots + A_K^k v(\epsilon_1)\| \tag{27a}$$

$$\le \|A_K v(\epsilon_k)\| + \|A_K^2 v(\epsilon_{k-1})\| + \cdots + \|A_K^k v(\epsilon_1)\| \tag{27b}$$

$$\le \alpha\|v(\epsilon_k)\| + \alpha^2\|v(\epsilon_{k-1})\| + \cdots + \alpha^k\|v(\epsilon_1)\|. \tag{27c}$$

Taking the expectation over the random noise ϵ_k, and considering the fact that the random noise signal is independently distributed at different time instances, it follows that

$$\mathbb{E}\|e_{k+1}\| \le \alpha\mathbb{E}_{\epsilon_k}\|v(\epsilon_k)\| + \alpha^2\mathbb{E}_{\epsilon_{k-1}}\|v(\epsilon_{k-1})\| + \ldots \tag{28a}$$

$$+ \alpha^k\mathbb{E}_{\epsilon_1}\|v(\epsilon_1)\|. \tag{28b}$$

If Assumption 1 and 2 hold, we can further infer that

$$\mathbb{E}\|e_{k+1}\| \le \alpha L\mathbb{E}_{\epsilon_k}\|\epsilon_k\| + \alpha^2 L\mathbb{E}_{\epsilon_{k-1}}\|\epsilon_{k-1}\| + \ldots \tag{29a}$$

$$+ \alpha^k L\mathbb{E}_{\epsilon_1}\|\epsilon_1\| \tag{29b}$$

$$\le \alpha L\zeta + \alpha^2 L\zeta + \cdots + \alpha^k L\zeta \tag{29c}$$

$$= \frac{(\alpha - \alpha^k)L\zeta}{1-\alpha}. \tag{29d}$$

As $k \to \infty$, the expectation of the error state norm is bounded by $\frac{\alpha L\zeta}{1-\alpha}$. Since $\mathbb{E}x_k = \mathbb{E}C(\hat{\mu}_k + e_k)$, it can be obtained that $\lim_{k\to\infty}\mathbb{E}\|x_k\| \le \lim_{k\to\infty}\|C\hat{\mu}_k\| + \mathbb{E}\|Ce_k\| = \frac{\alpha\nu L\zeta}{1-\alpha}$ which concludes the proof of the uniform ultimate boundedness of system (6). ■

C. Stability in the Presence of Modeling Error

We further investigate the closed-loop stability of (6) governed by the controller in (18), under the case scenario when the modeling error is present. Proposition 1 in Section IV-B considers an ideal case scenario when ground-truth DRKO model parameters A, B, C, and θ are accessible. In practice, however, the optimal parameters are impossible to obtain/verify, and only suboptimal parameters $\hat{A}, \hat{B}, \hat{C}$, and $\hat{\theta}$ are available. In this case, the evolution of the mean vector (17) can be expressed as

$$\mu_{k+1} = \hat{A}\mu_k + \hat{B}u_k + v(\epsilon_k) + r_k \tag{30}$$

$$\mathbb{E}x_k = \hat{C}\mu_k + d_k \tag{31}$$

where $r_k := (A - \hat{A})\mu_k + (B - \hat{B})u_k$ denotes the dynamic residual; $d_k := (C - \hat{C})\mu_k$ denotes the observation residual. In addition, there also exists an encoding-induced deviation at each time step, $o_k := \mu_\theta(x_k, \delta_{k-1}) - \mu_{\hat{\theta}}(x_k, \delta_{k-1})$. Before proceeding, we make the following assumption.

Assumption 3: There exist positive constants $\gamma, \eta, \omega \in \mathbb{R}^+$, such that $\|\epsilon\| \leq \gamma$, $\|d\| \leq \eta$ and $\|o\| \leq \omega$, $\forall x \in \mathbb{X}$ and $u \in \mathbb{U}$.

Proposition 2: If Assumptions 1–3 hold, the nonlinear system in (6) controlled by the controller (13) and (18) is uniformly ultimately bounded by $\frac{\nu\alpha(\lambda\omega + L\zeta + \gamma)}{1-\alpha} + \eta$, where $\nu := \|\hat{C}\|$ and $\lambda := \|\hat{B}K\|$.

Proof: As the nominal system remains the same as in Proposition 1, the proof for the mean-square stability of the nominal system can be completed following the proof of Proposition 1. In the following, we prove the uniform ultimate boundedness for the error system in (30).

In the presence of the approximation residuals, the dynamics of the error system $e_k := \mu_k - \hat{\mu}_k$ are given as follows:

$$e_{k+1} = \hat{A}_K e_k - \hat{B}K o_k + v(\epsilon_k) + r_k \tag{32}$$

where $\hat{A}_K := \hat{A} + \hat{B}K$ represents the nominal closed-loop transition matrix. Considering the evaluation of the error dynamics from the initial time instant 1 to instant k, one has

$$e_{k+1}$$

$$= \hat{A}_K^k e_1 - \underbrace{\left(\hat{A}_K \hat{B}K o_k + \hat{A}_K^2 \hat{B}K o_{k-1} + \cdots + \hat{A}_K^k \hat{B}K o_1\right)}_{\sum_1^k \hat{A}_K^j \hat{B}K o_j} \tag{33a}$$

$$+ \underbrace{\hat{A}_K \epsilon_k + \hat{A}_K^2 \epsilon_{k-1} + \cdots + \hat{A}_K^k \epsilon_1}_{\sum_1^k \hat{A}_K^j \epsilon_j} \tag{33b}$$

$$+ \underbrace{\hat{A}_K v(\epsilon_k) + \hat{A}_K^2 v(\epsilon_{k-1}) + \cdots + \hat{A}_K^k v(\epsilon_1)}_{\sum_1^k \hat{A}_K^j v(\epsilon_j)}. \tag{33c}$$

At the initial time instant $k = 1$, the error state $e_1 = \mu_\theta(x_1, \delta_0) - \mu_{\hat{\theta}}(x_1, \delta_0)$. Then the l_2 norm of the error is given by

$$\|e_{k+1}\| \tag{34a}$$

$$= \left\| \hat{A}_K^k e_1 - \sum_{j=1}^{k} \hat{A}_K^j \hat{B}K o_j + \sum_{j=1}^{k} \hat{A}_K^j r_j + \sum_{j=1}^{k} \hat{A}_K^j v(\epsilon_j) \right\| \tag{34b}$$

$$\leq \left\| \hat{A}_K^k e_1 \right\| + \sum_{j=1}^{k} \left(\left\| \hat{A}_K^j \hat{B}K o_j \right\| + \left\| \hat{A}_K^j r_j \right\| + \left\| \hat{A}_K^j v(\epsilon_j) \right\| \right) \tag{34c}$$

$$\leq \alpha^k \|e_1\| + \sum_{j=1}^{k} \alpha^j \left(\lambda \|o_j\| + \|r_j\| + \|v(\epsilon_j)\| \right). \tag{34d}$$

Taking the expectation over the random noise ϵ and utilize Assumptions 1–3, it follows that:

$$\mathbb{E}\|e_{k+1}\| \leq \alpha^k \omega + \sum_{j=1}^{k} \alpha^j \left(\lambda\omega + \gamma + L\zeta\right) \tag{35a}$$

$$= \alpha^k \omega + \frac{(\alpha - \alpha^k)(\lambda\omega + L\zeta + \gamma)}{1 - \alpha}. \tag{35b}$$

In the original state space, the error between the actual and the nominal system is

$$x_k - \hat{x}_k = C\mu_k - \hat{C}\hat{\mu}_k = \hat{C}(\mu_k - \hat{\mu}_k) + d_k \tag{36}$$

which is bounded as

$$\|x_k - \hat{x}_k\| \leq \nu \|e_k\| + \eta. \tag{37}$$

As k approaches infinity, we can derive from (37) that

$$\lim_{k \to \infty} \|x_k - \hat{x}_k\| \leq \frac{\nu\alpha(\lambda\omega + L\zeta + \gamma)}{1 - \alpha} + \eta \tag{38}$$

which concludes the proof. ■

Remark 5: Assumption 3 requires that the one-step observable encoding, prediction, and reconstruction errors are bounded on the state-action space. This is practical because the learned models are typically near-optimal, and the associated one-step-ahead prediction errors are insignificant, which will be shown in the next section.

Remark 6: The entropy constraint introduced in Section III serves the purpose of mitigating the occurrence of overfitting in the deep recurrent Koopman model. This technique does not cause violations of assumptions or conditions in the controller design. Therefore, the use of this technique will not compromise the closed-loop stability.

V. Application to a Chemical Process

In this section, we will evaluate the proposed approach on a three-vessel chemical process.

A. Process and Control Problem Description

This chemical process comprises two continuously stirred tank reactors and a flash tank separator. A detailed description and modeling of the process can be found in [27]. The process is designed to convert a reactant A into a product B with a side product C, described by two reactions $A \rightarrow B$ and $B \rightarrow C$. The state vector of the process $x := [X_{A1}, X_{B1}, T_1, X_{A2}, X_{B2}, T_2, X_{A3}, X_{B3}, T_3]^{\mathrm{T}}$ contains nine variables, including the mass fractions of A and B which are denoted by X_{Ai} and X_{Bi}, and the temperatures T_i, 1,2,3 in the three vessels. The process was numerically simulated using a standard Euler integration method, and bounded process noise was added to the process to simulate process disturbances and noise. The control system manipulates the external heat inputs $Q_{1,2,3}$ to three vessels to steer the process from the initial state to a steady-state set-point $x_s = [0.18, 0.67, 480.3\,\mathrm{K}, 0.19, 0.65, 472.8\,\mathrm{K}, 0.06, 0.67, 474.9\,\mathrm{K}]^{\mathrm{T}}$. The initial state x_1 uniformly distributes in the region of $[0.8x_s, 1.2x_s]$. The heating inputs are subject to constraint $[0, 0, 0]^{\mathrm{T}} \times 10^6\,\mathrm{kJ/h} \leq u \leq [4.87, 1.68, 4.87]^{\mathrm{T}} \times 10^6\,\mathrm{kJ/h}$. Note that the set-point x_s corresponds to steady-state input $u_s = [3.25, 1.12, 3.25]^{\mathrm{T}} \times 10^6\,\mathrm{kJ/h}$, though the control system is not aware of this information.

The process noise ϵ is generated from the multivariate normal distribution $\mathcal{N}(\mathbf{0}, 10^{-4}I)$, and is clipped to the range of $[-5, 5]$ in each dimension. The sampling time period is $\Delta = 0.005$ h. From an application perspective, the mass fractions X_{Ai} and X_{Bi} are typically measured by using lab analyzers, and we consider that the use of lab analyzers introduces time delays in the measurements by 0.025 h. That is, at each new sampling instant k, the controller optimizes for control action u_k based on the measurements of X_{Ai} and X_{Bi} at sampling instant $k - 5$, instantaneous measure measurements of T_i and the historical trajectory.

A dataset comprising 10^5 samples of state-input pairs is generated by simulating the chemical process with a randomly generated input profile. In particular, after every 20 sampling periods, a constant vector u_c is uniformly sampled from the action space and the input signal is produced as $u = u_c + \epsilon_u$, where ϵ_u is sampled from the multivariate normal distribution $\mathcal{N}(\mathbf{0}, 10^{-4}u_s^2)$ at every instant, with u_s^2 (with a slight abuse of notation) denoting the element-wise multiplication of u_s. To facilitate the learning process, we adjust the states and actions by shifting and scaling them using their mean and standard deviation vectors. These normalized data samples take values around zero with a standard deviation of 1. To facilitate the learning algorithms, the states and actions are both shifted and scaled with their mean and standard deviation vectors, so that the data fed into the algorithms is normalized around zero with a standard deviation of 1. The dataset is randomly shuffled and divided into two parts, the training set and the validation set according to a 9 : 1 ratio. In addition, a test set comprising 10^4 samples is collected for training evaluation. In DRKO-based predictive control scheme, we set weighting matrices as $Q = \Lambda([1.5, 0.5, 0.5, 1.5, 0.5, 0.5, 1.5, 0.5, 0.5])$ and $R = \Lambda([0.1, 0.1, 0.1])$. The prediction horizon H is 40 and the control horizon is 10.

We also compare the performance of DRKO with a competitive baseline, the deep stochastic Koopman operator (DeSKO) method [15]. Hyperparameters of DRKO are shown in Table I. The neural network structure, the learning rate, and other hyperparameters are determined through trial and error. Both DRKO and DeSKO adopt the default random parameter initialization setting in Tensorflow [26], and no random seed specification is conducted. Therefore, in comparison to DeSKO, the proposed method does not reap the advantages of parameter randomization.

TABLE I
HYPERPARAMETERS OF DRKO

Hyperparameters	Value
Size of data set $\mathcal{D}$	10^5
Batch Size	256
Learning rate	10^{-3}
Prediction horzion H	30
Structure of $\mu_\theta(\cdot)$	$(128, 128)$
Structure of $\sigma_\theta(\cdot)$	$(128, 128)$
Number of LSTM cells	8
History horizon	8
Activation function	ELU
Dimension of observables	20
Entropy threshold $\mathcal{H}$	-20
l_2 norm regularization coefficient	0.1

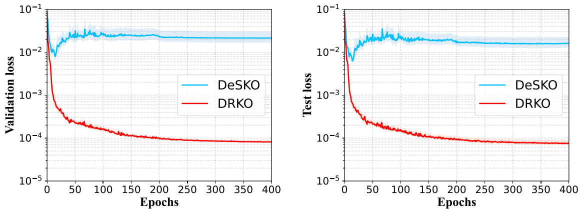


Fig. 3. Cumulative prediction error on the validation set and the test set. The Y-axis indicates the cumulative mean-squared prediction error in log space over 16 instants, and the X-axis indicates the training epochs. The shaded region shows the confidence interval (one standard deviation) over five random initializations.

B. Modeling Performance

First, we evaluate the modeling performance of DRKO in this subsection. For both DRKO and DeSKO, five models are trained with random parameter initialization. Each model is trained for 400 epochs, in each epoch the algorithm takes a batch of 256 data points from the dataset and updates the parameter until the dataset has been completely traversed. The l_2 norm of the prediction loss on the validation and test datasets in each epoch is presented in Fig. 3.

As shown in Fig. 3, both algorithms converge as training proceeds. Meanwhile, DRKO exhibits significantly higher accuracy as compared to DeSKO, with its prediction error being two orders of magnitude smaller. This improvement in performance can be attributed to its capability of incorporating historical data trajectories into the observables. Furthermore, DRKO provides a lower variance across different training trails, as indicated by the narrower shaded area, especially during the converging phase. In addition, the proposed method is more resilient against variation in the initial parameters as compared to DeSKO.

C. Control Performance

Next, we evaluate the control performance based on the two methods. For both DRKO and DeSKO, the most accurate models are selected for controller design. For each controller, 10 evaluation trials are conducted with random initial state x_1 as described in Section V-A. Each control design solves an optimization problem to find the steady-state input $\hat{u}_s$, which is used

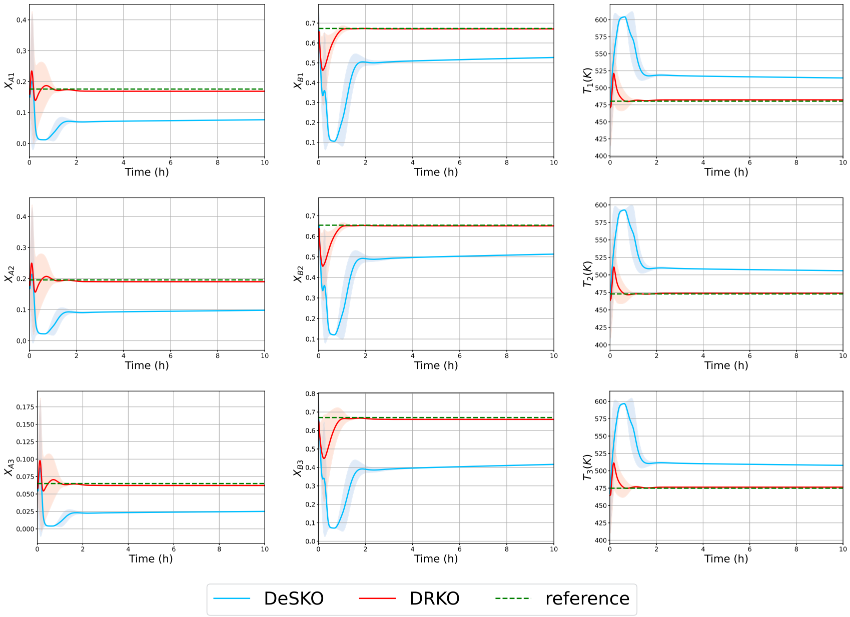


Fig. 4. State trajectories and reference signals in each dimension. The Y-axis indicates the state trajectories and the X-axis indicates the time in hours. The shaded region shows the confidence interval (one standard deviation) over 5 evaluation trials.

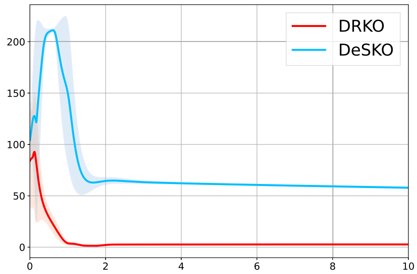


Fig. 5. Trajectory of the tracking error. The Y-axis indicates the l_2 norm of the tracking error; the X-axis indicates the time in hours. The shaded region shows the confidence interval (one standard deviation) over 5 evaluation trials.

by the corresponding predictive controller. The solution found by DRKO is $\hat{u}_s = [3.01, 1.05, 2.78]^{\mathrm{T}} \times 10^6$ kJ/h. The mean state trajectories and their standard deviations for the nine process states are presented in Fig. 4. Despite the presence of modeling error and the associated deviation of $\hat{u}_s$ from the group-truth u_s, the proposed DeSKO is able to provide good and robust control performance.

As shown in Fig. 4, starting from a randomly initialized state, DRKO is able to steer the states towards the reference with a short transition period and track the reference accurately. In comparison, DeSKO fails to track the set-point. The l_2 norms of the tracking errors for the two methods are shown in Fig. 5, which further confirm a significant improvement in the control performance of the proposed method as compared to the DeSKO baseline.

From Fig. 4, the trajectories of X_{A1} and X_{A2}, as well as X_{B1} and X_{B2} have similar trends. One primary factor that contributes to the similarity in each pair of state trajectories is the use of similar initial conditions for the two states in each pair.

VI. Conclusion

In this article, we proposed an efficient learning-based modeling and control framework for a general class of uncertain nonlinear systems with unknown delays. By exploiting LSTM networks and Koopman operators, a DRKO learning framework is proposed. Based on the learned Koopman linear model, a robust MPC controller is designed to stabilize the original system. Based on mild assumptions, the proposed learning-based predictive control method was proven to be able to provide guaranteed closed-loop stability for the considered nonlinear system. Through the application to a simulated chemical process, DRKO showed superior performance as compared to the SOTA baseline in terms of both modeling and control. Based on this work, there remain potential problems to be explored in the future: 1) In this work, an encoded multivariate Gaussian distribution was exploited to parameterize the distribution of observables; meanwhile, alternative distribution forms may be more advantageous for systems with colored noise; 2) in this work, control input constraints were incorporated, while the treatment of state constraints has been left for future study.

References

[1] J. Li, W. Li, X. Chang, K. Sharma, and Z. Yuan, "Real-time predictive control for chemical distribution in sewer networks using improved elephant herding optimization," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 571–581, Jan. 2022.

[2] A. H. González, J. L. Marchetti, and D. Odloak, "Robust model predictive control with zone control," *IET Control Theory Appl.*, vol. 3, no. 1, pp. 121–135, 2009.

[3] G. Ceusters et al., “Model-predictive control and reinforcement learning in multi-energy system case studies,” *Appl. Energy*, vol. 303, 2021, Art. no. 117634.

[4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[5] S. Lucia, D. Navarro, B. Karg, H. Sarnago, and O. Lucia, “Deep learning-based model predictive control for resonant power converters,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 1, pp. 409–420, Jan. 2021.

[6] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, “Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7559–7566.

[7] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: A brief survey,” *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[8] B. O. Koopman, “Hamiltonian systems and transformation in hilbert space,” *Proc. Nat. Acad. Sci. USA*, vol. 17, no. 5, pp. 315–318, 1931.

[9] P. J. Schmid, “Dynamic mode decomposition of numerical and experimental data,” *J. Fluid Mechanics*, vol. 656, pp. 5–28, 2010.

[10] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, “Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator,” *Chaos: An Interdiscipl. J. Nonlinear Sci.*, vol. 27, no. 10, 2017, Art. no. 103111.

[11] A. Narasingam and J. S.-I. Kwon, “Koopman lyapunov-based model predictive control of nonlinear chemical process systems,” *AIChE J.*, vol. 65, no. 11, 2019, Art. no. e16743.

[12] J. Morton, A. Jameson, M. J. Kochenderfer, and F. Witherden, “Deep dynamical modeling and control of unsteady fluid flows,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9278–9288.

[13] H. Shi and M. Q.-H. Meng, “Deep koopman operator with control for nonlinear systems,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7700–7707, Jul. 2022.

[14] O. Azencot, N. B. Erichson, V. Lin, and M. Mahoney, “Forecasting sequential data using consistent koopman autoencoders,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 475–485.

[15] M. Han, J. Euler-Rolle, and R. K. Katzschmann, “DeSKO: Stability-assured robust control with a deep stochastic Koopman operator,” in *Proc. Int. Conf. Learn. Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=hniLRD_XCA

[16] T. Zhao, M. Yue, and J. Wang, “Deep learning-based koopman modeling for online control synthesis of nonlinear power system transient dynamics,” *IEEE Trans. Indust. Informat.*, vol. 19, no. 10, pp. 10444–10453, Oct. 2023.

[17] D. M. Schneider, “Control of processes with time delays,” *IEEE Trans. Ind. Appl.*, vol. 24, no. 2, pp. 186–191, Mar./Apr. 1988.

[18] J. Lee, W. Cho, and T. F. Edgar, “Control system design based on a nonlinear first-order plus time delay model,” *J. Process Control*, vol. 7, no. 1, pp. 65–73, 1997.

[19] J. L. Proctor, S. L. Brunton, and J. N. Kutz, “Generalizing koopman theory to allow for inputs and control,” *SIAM J. Appl. Dyn. Syst.*, vol. 17, no. 1, pp. 909–930, 2018.

[20] P. Bolzern, P. Colaneri, and G. De Nicolao, “Markov jump linear systems with switching transition rates: Mean square stability with dwell-time,” *Automatica*, vol. 46, no. 6, pp. 1081–1088, 2010.

[21] J. Huang, Z. Han, X. Cai, and L. Liu, “Uniformly ultimately bounded tracking control of linear differential inclusions with stochastic disturbance,” *Math. Comput. Simul.*, vol. 81, no. 12, pp. 2662–2672, 2011.

[22] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.

[23] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.

[24] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: LSTM cells and network architectures,” *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.

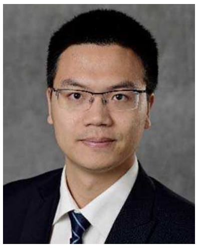
[25] K. Chua, R. Calandra, R. McAllister, and S. Levine, “Deep reinforcement learning in a handful of trials using probabilistic dynamics models,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4759–4770.

[26] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]. Available: https://www.tensorflow.org/

[27] J. Zhang and J. Liu, “Distributed moving horizon state estimation for nonlinear systems with bounded uncertainties,” *J. Process Control*, vol. 23, no. 9, pp. 1281–1295, 2013.

Minghao Han received the B.S. and the Ph.D. degrees in control science and engineering from the Department of Control Science and Engineering, Harbin Institution of Technology, Harbin, China, in 2017 and 2022, respectively.

He was a Visiting Researcher at ETH Zurich, Zurich, Switzerland. He is currently a Postdoctoral Research Fellow with Smart Process Systems Sensing and Control Lab, Nanyang Technological University, Singapore. His research interests include control theory, reinforcement learning, and their application in making complex industrial processes, autonomous vehicles and robots more intelligent, safe and reliable.



Zhaojian Li (Senior Member, IEEE) received the M.S. degree and the Ph.D. degree in aerospace engineering (flight dynamics and control) from the University of Michigan, Ann Arbor, USA, in 2013 and 2015, respectively.

He is a Red Cedar Distinguished Associate Professor with the Department of Mechanical Engineering, Michigan State University, MI, USA. From 2016 to 2017, he was an Algorithm Engineer with General Motors. He has authored or coauthored more than 50 top journal articles and several patents. His research interests include learning-based control, nonlinear and complex systems, and robotics and automated vehicles.

Dr. Li is a recipient of the NSF CAREER award. He is currently an Associate Editor for IEEE Transactions on Intelligent Vehicles, *Journal of Evolving Systems*, American Control Conference, and ASME Dynamics and Control Conference.

Xiang Yin (Member, IEEE) was born in Anhui, China, in 1991. He received the B.Eng degree from Zhejiang University, Zhejiang, China, in 2012, the M.S. degree from the University of Michigan, Ann Arbor, in 2013, and the Ph.D. degree from the University of Michigan, Ann Arbor, USA, in 2017, all in electrical engineering.

Since 2017, he has been with the Department of Automation, Shanghai Jiao Tong University, where he is an Associate Professor. His research interests include formal methods, discrete-event systems and cyber-physical systems.

Dr. Yin was the recipient of the IEEE Conference on Decision and Control (CDC) Best Student Paper Award Finalist in 2016. He is serving as the chair of the IEEE CSS Technical Committee on Discrete Event Systems, and as Associate Editor for *Journal of Discrete Event Dynamic Systems: Theory & Applications*, the *IEEE Control Systems Letters*, the IEEE Transactions on Intelligent Vehicles, and as a Member of the *IEEE CSS Conference Editorial Board*.

Xunyuan Yin received the Ph.D. degree in process control from the University of Alberta, Edmonton, AB, Canada, in 2018.

He worked as a Postdoctoral Fellow with the University of Alberta. He is currently an Assistant Professor with the School of Chemistry, Chemical Engineering and Biotechnology, Nanyang Technological University, Singapore.

He is an Associate Editor for IEEE Conference on Decision and Control, IFAC Modeling, Estimation and Control Conference, and American Control Conference. He was an Outstanding Reviewer for IEEE Transactions on Automatic Control and IEEE Transactions on Cybernetics. His research interests include optimization-based state estimation and control, learning-based process modeling and control, and their applications to complex industrial processes.

Dr. Yin is a Member of the Early Career Advisory Board of IFAC Journal Control Engineering Practice.