

# MAKO: Meta-Adaptive Koopman Operators for Learning-based Model Predictive Control of Parametrically Uncertain Nonlinear Systems

Minghao Han <sup>a,b</sup>, Kiwan Wong <sup>c</sup>, Adrian Wing-Keung Law <sup>a,d</sup>, Xunyuan Yin <sup>a,b</sup>

<sup>a</sup>*Nanyang Environment and Water Research Institute (NEWRI), Nanyang Technological University, Singapore*

<sup>b</sup>*School of Chemistry, Chemical Engineering and Biotechnology, Nanyang Technological University, Singapore*

<sup>c</sup>*Soft Robotics Lab, ETH Zurich, Switzerland*

<sup>d</sup>*Department of Civil and Environmental Engineering, National University of Singapore, Singapore*

---

## Abstract

In this work, we propose a meta-learning-based Koopman modeling and predictive control approach for nonlinear systems with parametric uncertainties. An adaptive deep meta-learning-based modeling approach, called Meta Adaptive Koopman Operator (MAKO), is proposed. Without knowledge of the parametric uncertainty, the proposed MAKO approach can learn a meta-model from a multi-modal dataset and efficiently adapt to new systems with previously unseen parameter settings by using online data. Based on the learned meta Koopman model, a predictive control scheme is developed, and the stability of the closed-loop system is ensured even in the presence of previously unseen parameter settings. Through extensive simulations, our proposed approach demonstrates superior performance in both modeling accuracy and control efficacy as compared to competitive baselines.

*Key words:* Meta-learning, Koopman operator, model predictive control, parametrically uncertain nonlinear systems

---

## 1 Introduction

Parametric uncertainties are common in nonlinear systems, often arising from factors such as variations in payload and operating conditions [1,2]. The presence of these uncertainties can cause performance degradation and instability and pose great challenges to the design of control systems. The endeavor to ensure desired con-

trol system tracking performance and system stability has driven the advancement of adaptive control methods for nonlinear systems with parametric uncertainties over recent decades.

Model predictive control (MPC) is a popular advanced control technique [3]. It optimizes the future predicted behavior of the system by utilizing a dynamic model along with current measurement data. Adaptive model predictive control (AMPC) has been proposed to address uncertainties [4,5]; however, the results on nonlinear systems have been limited. In [6], a receding horizon predictive control scheme was developed for nonlinear systems that are subject to control constraints and are linear in the unknown parameters. In [7], robust MPC was integrated with a min-max approach and a Lipschitz-based approach for online parameter update. In [8], a self-triggered AMPC method was proposed for constrained discrete-time nonlinear systems facing parametric uncertainties and additive disturbances. More results on nonlinear AMPC can be found in [9,10]. Although these results represent significant advancements, first-principle models are typically required as the foundation of control system designs. The theoretical assumption of lin-

---

\* This research is supported by the National Research Foundation, Singapore, and PUB, Singapore's National Water Agency under its RIE2025 Urban Solutions and Sustainability (USS) (Water) Centre of Excellence (CoE) Programme, awarded to Nanyang Environment & Water Research Institute (NEWRI), Nanyang Technological University, Singapore (NTU). This research is also supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG63/22). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore, and PUB, Singapore's National Water Agency.

\*\*Corresponding author: X. Yin. Email: xunyuan.yin@ntu.edu.sg.

ear dependency on uncertain parameters further limits its applicability to general nonlinear processes.

Recently, the Koopman operator theory has gained substantial research attention, owing to its capability to represent the dynamics of complex nonlinear processes in a linear manner [11]. Several algorithms have been developed to construct linear Koopman models from process data. These include dynamic mode decomposition (DMD) [12] and extended dynamic mode decomposition (EDMD) [13]. DMD represents the observables directly in the original state space, while EDMD employs a predetermined set of basis functions; both methods solve least-squares problems to approximate the linear Koopman operator. [14] proposed a sampling theorem for the exact identification of continuous-time nonlinear dynamical systems using the Koopman operators. To streamline the design of the observable functions, researchers have proposed various ML-enabled Koopman modeling and control methods [15–22]. In [23], the authors have provided convergence guarantees for the error as the capacity of neural network (NN) increases, and have derived the error upper bound with connection to the spectral property of the Koopman operator. However, these existing approaches have primarily targeted addressing a specific control task with fixed model parameters.

In this work, we aim to exploit the Koopman operator framework and machine learning (ML) to facilitate the modeling and control of parametrically uncertain nonlinear systems. Within the context of ML-based modeling, we acknowledge that uncertainties may be introduced in a beneficial manner to mitigate overfitting (e.g. Monte-Carlo dropout) as well as to aid the assessment of uncertainties of the ML predictions [24]. Nonetheless, the learning-based modeling and control of parametrically uncertain systems can be conceptualized as a multi-task problem, which can be effectively addressed using the meta-learning concept [25]. Meta-learning, often referred to as “learning to learn”, is a paradigm in ML that focuses on developing algorithms that are capable of generalizing across tasks. Meta-learning aims to extract and utilize knowledge from multiple related tasks, enabling rapid adaptation to new, unseen tasks with minimal data or computational effort. Recently, significant advancements have been achieved in integrating meta-learning with control. In [26], the optimization landscape of the model-agnostic meta-learning (MAML) algorithm was investigated, with a focus on identifying conditions that guarantee its global convergence in a single task LQR setting. [27] established sufficient conditions for the stability of the dynamical system during optimization and proved that MAML converges to a stationary point in the multi-task LQR setting. [28] introduced a MAML-based method for solving LQR problems in multi-task, heterogeneous settings, and has provided personalization guarantees for both model-based and model-free learning. In addition, meta-reinforcement learning (meta RL), a fusion of meta-learning and RL, has been developed for learning-based control in multi-task scenarios. Meta RL controllers uti-

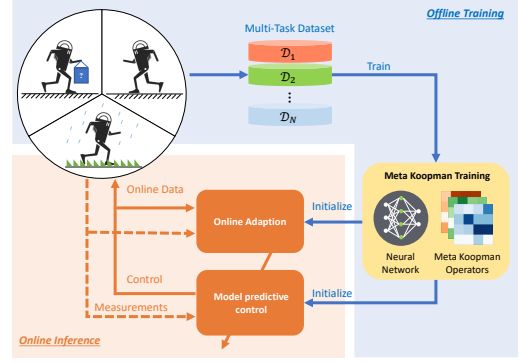


Fig. 1. An overview of the Meta Koopman pipeline.

lize previously acquired knowledge and real-time data to adapt to new tasks, wherein the system dynamics, objectives, or distribution of noise and disturbance can vary. [29] introduced a meta-learning-based MPC framework capable of fine-tuning the meta-trained NN model using online data. This method was applied to control a legged robot in the presence of changed payloads, terrains, and even a disabled leg. In [30], a novel offline meta RL strategy was proposed for tuning proportional-integral controllers in process control systems. However, the online adaptation of deep NNs is inefficient and computationally demanding. Furthermore, it is generally challenging for these meta-RL approaches to offer stability and ensure closed-loop performance.

Based on these observations, we aim to integrate meta-learning with Koopman operator theory to create a learning-based adaptive control framework for parametrically uncertain nonlinear systems. Within the Koopman operator framework, we proposed a meta-adaptive Koopman operator (MAKO) modeling approach. This approach learns from a multi-modal dataset to construct a meta-model for online adaptation. An adaptation scheme is developed to update the meta-model using online data while ensuring convergence. Based on the adaptive meta-model, a predictive control scheme is proposed for the underlying uncertain nonlinear systems. The contributions of this work include: 1) Meta-learning and Koopman operator theory are integrated for the first time to establish a learning-based adaptive MPC framework applicable to a general class of parametrically uncertain nonlinear systems. 2) We rigorously prove the convergence of both the model online adaptation and the closed-loop system. 3) Based on three benchmark systems from various fields, MAKO demonstrates good modeling accuracy and robust tracking control performance in the presence of parameter uncertainties, and it outperforms competitive baselines.

## 2 Preliminaries

### 2.1 Meta-learning

Meta-learning is concerned with developing automatic learning algorithms that can leverage data from previ-

ous tasks to quickly adapt to new tasks with trials. The new tasks may differ from the previous tasks in terms of system dynamics, noise distributions, and control objectives [25]. In this work, we focus on scenarios where the parameters of the nonlinear system vary in different task settings, that is,

$$x_{k+1} = f(x_k, u_k, \Theta), \text{ s.t. } \Theta \sim p(\Theta) \quad (1)$$

where  $f$  is an unknown nonlinear function of the state  $x_k \in \mathcal{X} \subset \mathbb{R}^n$ , the control input  $u_k \in \mathcal{U} \subset \mathbb{R}^m$ , and the system parameter  $\Theta \in \Xi \subset \mathbb{R}^l$ . We use  $\mathcal{X}$  and  $\mathcal{U}$  to denote the state space and input space, respectively. We use  $\Xi$  to denote the space of parameters. The parameter of the system, denoted by  $\Theta$ , follows an unknown distribution  $p(\Theta)$ , and each instance of  $\Theta$  corresponds to a specific task setting. In the following, we denote a sampled instance of  $\Theta$  as  $\Theta^i$ .

We formulate the supervised meta-model learning problem as

$$\min_{\theta} \mathbb{E}_{p(\Theta^i)} [\mathcal{L}(\mathcal{D}_i, \theta)] \quad (2)$$

$$\mathcal{L}(\mathcal{D}_i, \theta) = \frac{1}{T} \sum_{k=1}^T \left\| x_{k+1}^i - \hat{f}_{\theta}(x_k^i, u_k^i) \right\| \quad (3)$$

where  $\hat{f}_{\theta}$  is a parameterized model to approximate the unknown dynamic function  $f$  in (1), and  $\theta$  denotes the parameters to be optimized. We use  $\mathcal{D}_i := \{[x_k^i, u_k^i]_{k=1}^T | \Theta^i\}$  to denote the sub-dataset of  $T$  steps of state-input data collected under a task setting  $\Theta^i$ . The meta-dataset  $\mathcal{D}_{\Theta} := \{\mathcal{D}_i\}_{i=1}^N$  comprises  $N$  sub-datasets. In this work, we present a meta-Koopman framework to learn a model that can effectively adapt to new tasks.

## 2.2 The Koopman operator

In this subsection, we briefly introduce the ideas and notations of the Koopman operator theory. The Koopman theory was first formulated in [11]. According to the Koopman theory, a general nonlinear system of the form  $x_{k+1} = f(x_k), k \in \mathbb{N}$ , can be transformed into a linear system within an infinite-dimensional function space  $\mathcal{G}$ . This space encompasses all square-integrable real-valued functions defined over the compact domain  $\mathcal{X}$ . The elements of  $\mathcal{G}$ , denoted as  $\phi$ , are referred to as *observables*. The Koopman operator  $\mathcal{K} : \mathcal{G} \rightarrow \mathcal{G}$  satisfies the relation  $\phi \circ f(x_k) = \mathcal{K}\phi(x_k)$ , where  $\circ$  denotes function composition, and  $\phi \in \mathcal{G}$  represents the observable function. While initially proposed for autonomous nonlinear systems, the concept of the Koopman operator has been extended to controlled systems in recent years [31,32]. For controlled systems, the Koopman operator adheres to the following condition  $\phi_x \circ f(x_k, u_k) = A\phi_x(x_k) + B\phi_u(x_k, u_k)$ , where  $A$  and  $B$  are submatrices of the Koopman operator, and  $\phi_x$  and  $\phi_u$  represent the observables for the state  $x_k$  and control input  $u_k$ , respectively. In practical applications, it is often relevant to

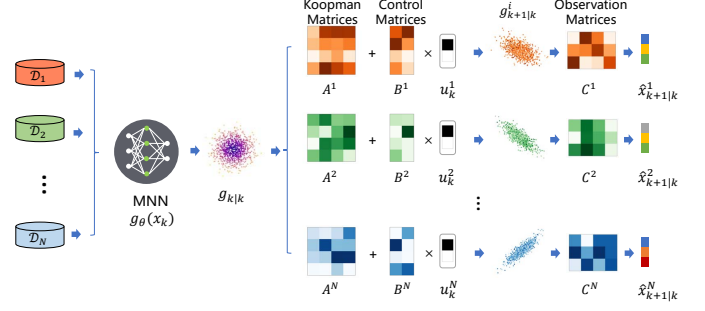


Fig. 2. An overview of the proposed MAKO pipeline. The meta-trained neural network encodes a common set of observables. On the observable space, individual Koopman operators are trained for each  $\mathcal{D}_i$  to predict future observables. The observables are transformed back to the state space with  $C^i$  to predict the future states.

find a finite-dimensional numerical approximation of  $\mathcal{K}$  within a finite-dimensional function space  $\bar{\mathcal{G}} \subset \mathcal{G}$  for the controller design [31,33]. This space is defined by a set of linearly independent observables  $\{\psi | \psi : \mathbb{R}^n \rightarrow \mathbb{R}^h\}$ . To facilitate controller design and analysis, the Koopman operator is typically approximated in a space where the control inputs act linearly [31,33], i.e.  $\phi_u(x_k, u_k) = u_k$ .

## 3 Method

In this section, we elaborate on the architecture of the proposed meta-adaptive Koopman operator (MAKO) model learning approach and elucidate how it learns and adapts to new tasks.

### 3.1 Meta-trained Koopman model

The MAKO model comprises two trainable building blocks: a meta-trained neural network (MNN) responsible for parameterizing the observable functions, and a set of linear Koopman operators for predicting future states and observables across various tasks. An overview of our pipeline involving the MNN and the Koopman operators is presented in Fig. 2.

#### 3.1.1 Meta-trained neural network

In the proposed framework, the MNN plays the key role of encoding an informative observable space, which is shared across different task settings. The key insight lies in recognizing that while the dynamics of the system vary across different tasks, the latent variables that characterize the system dynamics shall remain consistent. Inspired by this understanding, the observables of the system under different task settings are encoded as follows,

$$g_k^i = \psi_{\theta}(x_k^i), x_k^i \in \mathcal{D}_i \quad (4)$$

where  $\psi_{\theta}(\cdot)$  is a multi-layer neural network parameterized by the trainable parameters  $\theta$ . As shown in Fig. 2, the sub-datasets collected under different task settings

are passed through the same MNN. Consequently, the MNN may not distinguish the task settings and focus on extracting the key dynamic features shared across different settings.

The encoding mechanism of the MNN, which maps the original state space to a latent observable space, is conceptually similar to the encoder in Koopman autoencoders [18,20]. Nonetheless, the MNN in our approach focuses on creating a shared representation across multi-modal task settings to support adaptation rather than reconstructing the states in the nominal setting, which is different from previous autoencoders in [18,20].

### 3.1.2 Koopman operator

In the encoded observable space, the Koopman operators propagate the observables forward to predict future states. The dynamic behavior of the system varies across different task settings. Accordingly, for each task setting  $\Theta^i$ , one set of Koopman operators  $A^i$ ,  $B^i$ , and  $C^i$  will be learned to characterize the specific dynamic behavior. For a dataset consisting of  $N$  sub-datasets,  $N$  sets of Koopman operators will be learned. The dynamic behavior of the Koopman system under  $\Theta^i$  is described as:

$$g_{k+1|k}^i = A^i g_{k|k}^i + B^i u_k^i \quad (5a)$$

$$\hat{x}_{k+1|k}^i = C^i g_{k+1|k}^i \quad (5b)$$

The Koopman operators  $A^i, B^i, C^i$  in (5) and the parameters of the MNN, denoted by  $\theta$ , are trainable.

### 3.2 Meta-learning

In this subsection, we elaborate on the details of learning a meta-adaptive Koopman model that can be effectively adapted to different task settings. The goal of meta-model learning is to find a set of parameters  $\theta, A^i, B^i, C^i$  that minimize the multi-step-ahead prediction error characterized by:

$$\mathcal{L}(\theta, A^i, B^i, C^i) = \mathbb{E}_{p(\Theta^i)} \frac{1}{HT} \sum_{k=1}^T \sum_{t=1}^H \|x_{k+t}^i - C^i g_{k+t|k}^i\|_2^2 \quad (6a)$$

$$g_{k+t|k}^i = A^i g_{k+t-1|k}^i + B^i u_{k+t-1}^i \quad (6b)$$

$$g_{k|k}^i = \psi_\theta(x_k^i), \{x_k^i, u_k^i\} \in \mathcal{D}_i \quad (6c)$$

where  $H$  denotes the prediction horizon and  $T$  denotes the length of data trajectories. The MAKO model is trained to minimize the expected prediction error under the unknown task distribution  $p(\Theta^i)$ , which in practice can be approximated by using a meta-dataset  $\mathcal{D}_\Theta$  consisting of multiple sub-datasets  $\mathcal{D}_i$ ,  $i = 1, \dots, N$ .

Incorporating the concepts above, the optimization

problem for MAKO modeling is formulated as follows:

$$\min_{\theta, \{A^i, B^i, C^i\}} \frac{1}{NTH} \sum_{i=1}^N \sum_{k=1}^T \sum_{t=1}^H \|x_{k+t}^i - C^i g_{k+t|k}^i\|_2^2 \quad (7a)$$

$$\text{s.t. } g_{k+t|k}^i = A^i g_{k+t-1|k}^i + B^i u_{k+t-1}^i \quad (7b)$$

$$g_{k|k}^i = \psi_\theta(x_k^i), \{x_k^i, u_k^i\} \in \mathcal{D}_i \quad (7c)$$

**Remark 1** The proposed scheme is applicable to systems beyond linear time-invariant systems considered in [26–28]. However, we acknowledge the challenge of establishing a tight bound for the generalization and performance of the meta-trained Koopman model without access to the explicit forms of  $p(\Theta)$  and  $f(x, u, \Theta)$ . This limitation contrasts with prior work, such as [26–28], where such bounds have been more rigorously analyzed. Establishing performance guarantees for meta-learning in various types of parametrically uncertain systems may be a promising topic to explore in the future.

### 3.3 Online adaptation

We elaborate on how to adapt the meta-trained Koopman model, which is presented in the previous subsection, to new tasks using online data.

First, the Koopman operators learned on the meta-dataset can be combined to serve as an initial approximation of the exact Koopman operators in the new setting, denoted as  $\hat{A}_0, \hat{B}_0, \hat{C}_0 = \{\frac{1}{N} \sum A^i, \frac{1}{N} \sum B^i, \frac{1}{N} \sum C^i\}$ . Let  $\hat{\Psi}_k := [\hat{A}_k, \hat{B}_k]$  denote the approximated Koopman operators at instant  $k$ ,  $\hat{\Psi}_0 = [\hat{A}_0, \hat{B}_0]$ .  $X_k := [g_k^T, u_k^T]^T$  denotes the extended observable, one has the observable prediction error given by  $\tilde{g}_{k+1} := g_{k+1} - \hat{\Psi}_k X_k$ . In addition, the prediction error of the state is denoted as  $\tilde{x}_{k+1} := x_{k+1} - \hat{C}_k g_{k+1}$ . Define the cost function for the state and observable prediction error,  $J(\hat{\Psi}_k, \hat{C}_k) := \|g_{k+1} - \hat{\Psi}_k X_k\|_2^2 + \|x_{k+1} - \hat{C}_k g_{k+1}\|_2^2$ . To further refine the model, the online data generated during the exploitation stage will be used through gradient descent. The gradient of  $J$  with respect to  $\hat{\Psi}_k$  and  $\hat{C}_k$  can be obtained following

$$\nabla_{\hat{\Psi}} J_k := \frac{\partial J(\hat{\Psi}_k, \hat{C}_k)}{\partial \hat{\Psi}_k} = -X_k \tilde{g}_{k+1}^T \quad (8a)$$

$$\nabla_{\hat{C}} J_k := \frac{\partial J(\hat{\Psi}_k, \hat{C}_k)}{\partial \hat{C}_k} = -g_{k+1} \tilde{x}_{k+1}^T \quad (8b)$$

The update law for the Koopman operators is given as

$$\hat{\Psi}_{k+1} = \hat{\Psi}_k - \lambda_k \nabla_{\hat{\Psi}} J_k^T = \hat{\Psi}_k + \lambda_k \tilde{g}_{k+1} X_k^T \quad (9a)$$

$$\hat{C}_{k+1} = \hat{C}_k - \lambda_k \nabla_{\hat{C}} J_k^T = \hat{C}_k + \lambda_k \tilde{x}_{k+1} g_{k+1}^T \quad (9b)$$

where  $\lambda_k$  denotes the learning rate at time instant  $k$ . In the online adaptation, we propose to use an adaptive

learning rate adopted from [34] in the following form:

$$\lambda_k = \min \left( \frac{2 - \alpha}{X_k^T X_k}, \frac{2 - \alpha}{g_{k+1}^T g_{k+1}} \right) \quad (10)$$

where  $\alpha$  is a pre-determined hyperparameter subject to  $0 < \alpha < 2$ .

### 3.3.1 Nominal adaptation

We first consider the case where the exact Koopman operators exist on the finite-dimensional observable space. Before establishing the theoretical guarantee, we introduce the following assumptions.

**Assumption 1** *The domain  $\mathcal{U}$  is bounded and is given such that  $\mathcal{X}$  is bounded and forward invariant.*

**Assumption 2** *The lifting function  $\psi_\theta(\cdot)$  and the system dynamics  $f(\cdot, \cdot, \Theta)$  are continuous on  $\mathcal{X}$  and  $\mathcal{U}$ ,  $\forall \Theta \in \Xi$ .*

**Assumption 3** *For each task setting  $\Theta$ , there exists a set of Koopman operators  $A, B, C$  on the observable space encoded by the  $\psi_\theta(\cdot)$ , such that  $f(x, u, \Theta) = C(A\psi_\theta(x) + Bu)$ , for all  $x \in \mathcal{X}$  and  $u \in \mathcal{U}$ .*

A stricter version of Assumption 1 was adopted in [35] where  $\mathcal{X}$  is compact. Assumption 3 essentially ensures the existence of exact Koopman operators. Based on the above assumption, we present the following theorem on the convergence of the parameter approximation and model prediction errors. Let  $\tilde{\Psi}_k := \Psi - \hat{\Psi}_k$  and  $\tilde{C}_k := C - \hat{C}_k$  denote the parameter approximation error, where  $\Psi := [A, B]$  and  $C$  denote the exact Koopman operators. Before proceeding, we introduce relevant properties of the trace of matrices to facilitate the proof.

**Property 1** *For matrices  $M_1$  and  $M_2$ , and vectors  $x$  and  $y$ , all with proper dimensions, the following properties hold:*

- $\text{tr}(M_1 M_2) = \text{tr}(M_2 M_1)$
- $\text{tr}(M_1 + M_2) = \text{tr}(M_1) + \text{tr}(M_2)$
- $\text{tr}(yx^T) = \text{tr}(x^T y)$

where  $\text{tr}(\cdot)$  denotes the trace of a given matrix.

**Theorem 1** *Consider the uncertain nonlinear system (1) with uncertain parameters  $\Theta$  and the corresponding Koopman operators  $A, B$ , and  $C$ . If Assumptions 1-3 hold, with the adaptive updating laws (9) and (10), the parameter approximation errors  $\tilde{\Psi}_k$  and  $\tilde{C}_k$  are ultimately bounded and the predicted state error  $\tilde{x}$  asymptotically converges to zero.*

**Proof 1** *The proof of Theorem 1 is based on that of Theorem 1 in [34]. First, let us select a Lyapunov candidate*

$V_k := \text{tr}(\tilde{\Psi}_k^T \tilde{\Psi}_k) + \text{tr}(\tilde{C}_k^T \tilde{C}_k)$ . It can be derived that

$$V_{k+1} = V_k + \text{tr} \left( 2\lambda_k \tilde{\Psi}_k^T \nabla_{\hat{\Psi}} J_k^T + \lambda_k^2 \nabla_{\hat{\Psi}} J_k \nabla_{\hat{\Psi}} J_k^T \right) + \text{tr} \left( 2\lambda_k \tilde{C}_k^T \nabla_{\hat{C}} J_k^T + \lambda_k^2 \nabla_{\hat{C}} J_k \nabla_{\hat{C}} J_k^T \right) \quad (11)$$

By substituting (8) into (11), the second and the third terms on the right-hand-side of (11) can be computed as  $\text{tr}(2\lambda_k \tilde{\Psi}_k^T \nabla_{\hat{\Psi}} J_k^T + \lambda_k^2 \nabla_{\hat{\Psi}} J_k \nabla_{\hat{\Psi}} J_k^T) = \lambda_k \text{tr}(-2\tilde{\Psi}_k^T \tilde{g}_{k+1} X_k^T + \lambda_k X_k \tilde{g}_{k+1}^T \tilde{g}_{k+1} X_k^T) = \lambda_k (-2X_k^T \tilde{\Psi}_k^T \tilde{g}_{k+1} + \lambda_k X_k^T X_k \tilde{g}_{k+1}^T \tilde{g}_{k+1}) = \lambda_k (-2\tilde{g}_{k+1}^T \tilde{g}_{k+1} + \lambda_k X_k^T X_k \tilde{g}_{k+1}^T \tilde{g}_{k+1}) = \lambda_k (-2 + \lambda_k X_k^T X_k) \tilde{g}_{k+1}^T \tilde{g}_{k+1}$ . By taking (10) into account, one has  $\text{tr}(2\lambda_k \tilde{\Psi}_k^T \nabla_{\hat{\Psi}} J_k^T + \lambda_k^2 \nabla_{\hat{\Psi}} J_k \nabla_{\hat{\Psi}} J_k^T) \leq -\lambda_k \alpha \tilde{g}_{k+1}^T \tilde{g}_{k+1}$ . Following similar derivations as above, the following inequality holds,  $\text{tr}(2\lambda_k \tilde{C}_k^T \nabla_{\hat{C}} J_k^T + \lambda_k^2 \nabla_{\hat{C}} J_k \nabla_{\hat{C}} J_k^T) \leq -\lambda_k \alpha \tilde{x}_{k+1}^T \tilde{x}_{k+1}$ . Therefore, it follows from (11) that

$$V_{k+1} \leq V_k - \lambda_k \alpha (\tilde{g}_{k+1}^T \tilde{g}_{k+1} + \tilde{x}_{k+1}^T \tilde{x}_{k+1}) \quad (12)$$

which implies that  $V_k$  is decreasing as  $k$  increases. Furthermore, since  $V_k \geq 0$ ,  $\lim_{k \rightarrow +\infty} V_k$  exists, and the parameter estimation errors  $\tilde{\Psi}$  and  $\tilde{C}$  are ultimately bounded.

Applying (12) to all the time instants and aggregating the resulting inequalities yields:

$$V_{k+1} \leq V_1 - \alpha \sum_{i=1}^k \lambda_i (\tilde{g}_{i+1}^T \tilde{g}_{i+1} + \tilde{x}_{i+1}^T \tilde{x}_{i+1}) \quad (13)$$

Consequently,  $\alpha \sum_{k=1}^{+\infty} \lambda_k (\tilde{g}_{k+1}^T \tilde{g}_{k+1} + \tilde{x}_{k+1}^T \tilde{x}_{k+1}) \leq V_1 - \lim_{k \rightarrow +\infty} V_{k+1}$ . Since  $\mathcal{X}$  and  $\mathcal{U}$  are bounded by Assumption 1, and the lifting function  $\psi_\theta(\cdot)$  is continuous according to Assumption 2, it can be inferred that  $X_k^T X_k$  and  $g_{k+1}^T g_{k+1}$  are upper bounded for all  $k$ . Consequently, there exists a positive constant  $\lambda$  such that  $\lambda_k \geq \lambda$  for all  $k$ . Therefore, we have  $\alpha \lambda \sum_{k=1}^{+\infty} (\tilde{g}_{k+1}^T \tilde{g}_{k+1} + \tilde{x}_{k+1}^T \tilde{x}_{k+1}) \leq V_1$ . This indicates the convergence of the infinite series on the left-hand side, which implies  $\tilde{g} \rightarrow 0$  and  $\tilde{x} \rightarrow 0$ .

### 3.3.2 Robust adaptation

Assumption 3 requires the existence of a set of Koopman operators that exactly characterize the original system on a finite-dimensional observable space. However, for the general class of nonlinear systems, particularly those with parametric uncertainties, the existence of a finite-dimensional invariant subspace cannot be guaranteed [14]. In the following, we consider the more practical case where the exact Koopman operators do not exist and

the modeling errors are present, as follows:

$$\psi_\theta(f(x_k, u_k, \Theta)) = A\psi_\theta(x_k) + Bu_k + w_k \quad (14a)$$

$$x_k = C\psi_\theta(x_k) + v_k \quad (14b)$$

Following the derivation in [33], we can infer that the modeling errors  $w \in \mathcal{W} \subset \mathbb{R}^h$  and  $v \in \mathcal{V} \subset \mathbb{R}^n$  are bounded based on Assumptions 1 and 2. Therefore, there exist positive constants  $\epsilon_w$  and  $\epsilon_v$  such that  $\|w\| \leq \epsilon_w, \forall w \in \mathcal{W}$  and  $\|v\| \leq \epsilon_v, \forall v \in \mathcal{V}$ .

Next, we propose a robust adaptation scheme. The objective function containing the modeling errors is  $\bar{J}(\hat{\Psi}_k, \hat{C}_k, w_k, v_k) = \|g_{k+1} - \hat{\Psi}_k X_k - w_k\|_2^2 + \|x_{k+1} - \hat{C}_k g_{k+1} - v_k\|_2^2$ . To propose a robust adaptation scheme, we first introduce the ideal noise as

$$w_k^*, v_k^* = \min_{\substack{w_k \in \mathcal{W} \\ v_k \in \mathcal{V}}} \bar{J}(\hat{\Psi}_k, \hat{C}_k, w_k, v_k) \quad (15)$$

The gradient of  $\bar{J}$  with respect to  $\hat{\Psi}_k$  and  $\hat{C}_k$  can be obtained following  $\nabla_{\hat{\Psi}} \bar{J}_k = -X_k(\tilde{g}_{k+1} - w_k^*)^T, \nabla_{\hat{C}} \bar{J}_k = -g_{k+1}(\tilde{x}_{k+1} - v_k^*)^T$ . The robust update law at each time step is given as

$$\hat{\Psi}_{k+1} = \hat{\Psi}_k + \lambda_k(\tilde{g}_{k+1} - w_k^*)X_k^T \quad (16a)$$

$$\hat{C}_{k+1} = \hat{C}_k + \lambda_k(\tilde{x}_{k+1} - v_k^*)g_{k+1}^T \quad (16b)$$

where  $\lambda_k$  is determined following (10).

**Theorem 2** Consider the uncertain nonlinear system (1) with uncertain parameters  $\Theta$ . If Assumptions 1 and 2 hold, then under the updating laws (10), (15) and (16), the parameter approximation errors  $\tilde{\Psi}_k, \tilde{C}_k$  are ultimately bounded, and  $\lim_{k \rightarrow \infty} \|\tilde{x}_k\| \leq \epsilon_v$ .

**Proof 2** The proof of Theorem 2 follows a similar procedure as in Theorem 1. We adopt  $V_k := \text{tr}(\tilde{\Psi}_k^T \tilde{\Psi}_k) + \text{tr}(\tilde{C}_k^T \tilde{C}_k)$  as the Lyapunov function. Let  $\Delta(\tilde{\Psi}_k, \nabla_{\hat{\Psi}} \bar{J}_k) := \text{tr}(2\lambda_k \tilde{\Psi}_k^T \nabla_{\hat{\Psi}} \bar{J}_k^T + \lambda_k^2 \nabla_{\hat{\Psi}} \bar{J}_k \nabla_{\hat{\Psi}} \bar{J}_k^T)$ . With the update law described in (16),  $\Delta(\tilde{\Psi}_k, \nabla_{\hat{\Psi}} \bar{J}_k) = \lambda_k \text{tr}(-2\tilde{\Psi}_k^T(\tilde{g}_{k+1} - w_k^*)X_k^T + \lambda_k X_k(\tilde{g}_{k+1} - w_k^*)^T(\tilde{g}_{k+1} - w_k^*)X_k^T) = \lambda_k(-2X_k^T \tilde{\Psi}_k^T(\tilde{g}_{k+1} - w_k^*) + \lambda_k X_k^T X_k \|\tilde{g}_{k+1} - w_k^*\|_2^2) = \lambda_k(-2(\tilde{g}_{k+1} - w_k^*)^T(\tilde{g}_{k+1} - w_k^*) + \lambda_k X_k^T X_k \|\tilde{g}_{k+1} - w_k^*\|_2^2)$ . Consider that  $w_k^*$  is the optimal solution to (15), it can be inferred that  $(\tilde{g}_{k+1} - w_k^*)^T(\tilde{g}_{k+1} - w_k^*) \geq \|\tilde{g}_{k+1} - w_k^*\|_2^2$ . It follows that  $\Delta(\tilde{\Psi}_k, \nabla_{\hat{\Psi}} \bar{J}_k) \leq \lambda_k(-2 + \lambda_k X_k^T X_k) \|\tilde{g}_{k+1} - w_k^*\|_2^2$ . Incorporating the learning rate (10), one has  $\Delta(\tilde{\Psi}_k, \nabla_{\hat{\Psi}} \bar{J}_k) \leq -\alpha \lambda_k \|\tilde{g}_{k+1} - w_k^*\|_2^2$ . Apply similar derivation to  $\Delta(\tilde{C}_k, \nabla_{\hat{C}} \bar{J}_k)$ , and combine with (11), one has

$$V_{k+1} \leq V_k - \alpha \lambda_k (\|\tilde{g}_{k+1} - w_k^*\|_2^2 + \|\tilde{x}_{k+1} - v_k^*\|_2^2) \quad (17)$$

Therefore, it is proved that the parameter approximation errors  $\tilde{\Psi}_k$  and  $\tilde{C}_k$  are ultimately bounded. Aggregating the inequalities (17) for all time instants, it can be inferred that  $\lim_{k \rightarrow \infty} \|g_{k+1} - w_k^*\| = 0$  and  $\lim_{k \rightarrow \infty} \|\tilde{x}_{k+1} - v_k^*\| = 0$  based on Assumptions 1 and 2. In addition, according to Assumption 2,  $\mathcal{W}$  and  $\mathcal{V}$  are bounded, and  $\|w_k^*\| \leq \epsilon_w$  and  $\|v_k^*\| \leq \epsilon_v$ . Therefore, we have  $\lim_{k \rightarrow \infty} \|\tilde{x}_{k+1}\| \leq \epsilon_v$ , which concludes the proof.

**Remark 2** The continuity of  $\psi_\theta(\cdot)$ , as considered in Assumption 2, can be guaranteed by adopting activation functions that are continuous, such as ReLU, ELU, and Sigmoid etc.

**Remark 3** The proofs of Theorem 1 and Theorem 2 are built based on the theoretical results in [34]. Compared to [34], the theoretical contributions of this current work are two-fold. First, [34] relies on the assumption of a small, constant learning rate to ensure convergence, while this work employs a dynamic learning rate as defined in (10). This dynamic learning rate adapts based on the state and input data at each time step, and this eases the restrictive condition of requiring a fixed, small learning rate. Second, while [34] focuses on nominal linear systems, this work extends the Koopman-based framework to nonlinear systems with parametric uncertainties. Modeling residuals on the finite-dimensional space are further considered, which is different from the noise-free setting in [34].

## 4 Meta-Koopman-based Adaptive Model Predictive Control

In this section, we propose an adaptive model predictive control (AMPC) approach based on the learned MAKO model. First, we present the MAKO-based AMPC design. Subsequently, we establish the stability criterion for the resulting closed-loop system.

### 4.1 MAKO-based Adaptive MPC

Based on the meta Koopman model, we design an MPC scheme[36] to solve the finite horizon optimal control problem, minimizing the cumulative stage cost. In the nominal setting, the dynamics of  $\hat{g}_k$  can be described by the MAKO model as follows:

$$\hat{g}_{k+1|k} = \hat{A}_k \hat{g}_{k|k} + \hat{B}_k u_k \quad (18a)$$

$$\hat{x}_{k+1|k} = \hat{C}_k \hat{g}_{k+1|k} \quad (18b)$$

where  $\hat{g}_{k|k}$  is the nominal observable encoded by the MNN. The AMPC solves the following deterministic op-

timal tracking control problem

$$\min_{u_{k:k+T}} V_k = \sum_{t=1}^T \|\hat{C}_k \hat{g}_{k+t|k} - x_s\|_Q^2 + \|\Delta u_{k+t|k}\|_R^2 \quad (19a)$$

$$\text{s.t. } \hat{g}_{k+t+1} = \hat{A}_k \hat{g}_{k+t|k} + \hat{B}_k u_{k+t|k} \quad (19b)$$

$$\hat{C}_k \hat{g}_{k+T+1|k} = x_s \quad (19c)$$

$$\Delta u_{k+t|k} = u_{k+t|k} - u_{k+t-1|k} \quad (19d)$$

$$u_{k+t|k} \in \mathcal{U} \quad (19e)$$

where  $x_s$  is the given set-point in the original state space,  $Q$  and  $R$  are known positive definite weighting matrices, and  $\|\cdot\|_W^2$  represents the weighted Euclidean norm with  $W$  being a positive definite matrix. At each sampling instant, the AMPC problem (19) is solved and the corresponding optimal control input  $u_{k|k}^*$  is applied to the system in (1). In the meantime, the Koopman matrices  $\hat{A}_k$ ,  $\hat{B}_k$  and  $\hat{C}_k$  are updated according to the update law described in (9).

#### 4.2 Stability with learned Koopman operators

Next, we prove the stability of the closed-loop system based on the AMPC controller in (19).

**Theorem 3** Consider the nonlinear system (1) with the adaptive updating law in (9) and (10), and the MPC controller in (19). If Assumptions 1-3 hold, then the tracking error of the closed-loop system is asymptotically stable.

**Proof 3** By solving the MPC problem (19), we can obtain the feasible optimal control sequence at time  $k$ ,  $\{u_{k|k}^*, u_{k+1|k}^*, \dots, u_{k+T|k}^*\}$ , and the resulting predicted optimal observable trajectory  $\{\hat{g}_{k|k}^*, \hat{g}_{k+1|k}^*, \dots, \hat{g}_{k+T|k}^*, \hat{g}_{k+T+1|k}^*\}$ . Consider an extended input trajectory

$$\{u_{k+1|k}^*, \dots, u_{k+T|k}^*, u_{k+T+1|k}^*\}, \quad (20)$$

where the control input at time instant  $k+T+1$  remains unchanged from  $u_{k+T|k}^*$ , i.e.,  $u_{k+T+1|k}^* = u_{k+T|k}^*$ .

Due to the update of the Koopman operators, there exists a deviation  $\delta$  between the state trajectory  $\{\hat{x}_{k+t|k}^*\}$  predicted at instant  $k$  and  $\{\hat{x}_{k+t|k+1}^*\}$  predicted at  $k+1$ , where  $t$  refers to the time step within the prediction horizon.  $\delta_k^t := (\hat{C}_{k+1} \hat{A}_{k+1}^t - \hat{C}_k \hat{A}_k^t) \hat{g}_{k+1|k}^* + \sum_{j=1}^t (\hat{C}_{k+1} \hat{A}_{k+1}^{j-1} \hat{B}_{k+1} - \hat{C}_k \hat{A}_k^{j-1} \hat{B}_k) u_{k+j|k}^*$ . Denote the update step as  $\Delta \hat{\Psi}_k := \hat{\Psi}_{k+1} - \hat{\Psi}_k$  and  $\Delta \hat{C}_k := \hat{C}_{k+1} - \hat{C}_k$ . It can be observed that  $\delta_k^t$  goes to zero, if  $\Delta \hat{\Psi}_k$  and  $\Delta \hat{C}_k$  approach zero.

In the following, we adopt the value function  $V_k^* = \sum_{t=1}^T \|\hat{C}_k \hat{g}_{k+t|k}^* - x_s\|_Q^2 + \|\Delta u_{k+t|k}^*\|_R^2$  as the Lyapunov candidate. The suitability of  $V_k$  as a Lyapunov candidate has been proved in [37]. Consider

the Lyapunov candidate  $V_{k+1}^*$  at time  $k+1$ ,  $V_{k+1}^* = \sum_{t=2}^{T+1} (\|\hat{C}_{k+1} \hat{g}_{k+t|k+1}^* - x_s\|_Q^2 + \|\Delta u_{k+t|k+1}^*\|_R^2)$ . Due to optimality, the value of  $V_{k+1}^*$  is no greater than the value function of the sub-optimal solution, that is,  $V_{k+1}^* \leq \sum_{t=2}^{T+1} (\|\hat{C}_k \hat{g}_{k+t|k}^* - x_s + \delta_k^t\|_Q^2 + \|\Delta u_{k+t|k}^*\|_R^2) \leq \sum_{t=2}^{T+1} (\|\hat{C}_k \hat{g}_{k+t|k}^* - x_s\|_Q^2 + \|\delta_k^t\|_Q^2 + \|\Delta u_{k+t|k}^*\|_R^2) = V_k^* + \|\hat{C}_k \hat{g}_{k+T+1|k}^* - x_s\|_Q^2 + \|\Delta u_{k+T+1|k}^*\|_R^2 - \|\hat{C}_k \hat{g}_{k+1|k}^* - x_s\|_Q^2 - \|\Delta u_{k+1|k}^*\|_R^2 + \sum_{t=2}^{T+1} \|\delta_k^t\|_Q^2$ . It follows from (19c) and (20) that  $\hat{C}_k \hat{g}_{k+T+1|k}^* = x_s$  and  $\Delta u_{k+T+1|k}^* = 0$ . Note that  $\lim_{k \rightarrow \infty} \|\tilde{g}_k\| = 0$  and  $\lim_{k \rightarrow \infty} \|\tilde{x}_k\| = 0$  according to Theorem 1, hence  $\Delta \hat{\Psi}_k$  and  $\Delta \hat{C}_k$  go to zero, and the residual approaches zero  $\lim_{k \rightarrow \infty} \delta_k^t = 0$  as a consequence. Therefore, it can be inferred that the tracking error of the nominal system in (18) is asymptotically stable. Furthermore, incorporate the result from Theorem 1 that the state prediction error asymptotically converges to zero, the closed-loop system is asymptotically stable.

**Corollary 1** Consider the nonlinear system (1) with the adaptive updating law given by (10), (15), and (16), and the MPC controller in (19). If Assumptions 1 and 2 hold, then the tracking error of the closed-loop system is ultimately bounded.

**Proof 4** The proof of Corollary 1 follows the same procedure adopted in the proof of Theorem 3, which establishes the asymptotic stability of the tracking error of the nominal system (18). By further incorporating that  $\lim_{k \rightarrow \infty} \|\tilde{x}_{k+1}\| \leq \epsilon_v$  according to Theorem 2, the tracking error of the closed-loop system can be proved to be ultimately bounded.

## 5 Results

In this section, the proposed MAKO learning-based control framework is evaluated and compared to a competitive baseline on three benchmark examples via simulations. The codes for reproducing our results can be found at the link provided in the footnote<sup>1</sup>.

### 5.1 Simulation setup and baselines

#### 5.1.1 Cartpole

First, we consider a classic cartpole balancing problem [38]. The controller is expected to maintain the pendulum in its upright, vertical orientation. The state vector comprises  $[x, \dot{x}, \theta, \dot{\theta}]^T$ , where  $x$  denotes the horizontal position of the cart and  $\theta$  denotes the angular position of the pole in rads. The action is the horizontal force applied to the cart ( $u \in [-20, 20]$ ).  $x_{\text{threshold}}$  and  $\theta_{\text{threshold}}$  represents the maximum position and angle, respectively, with  $x_{\text{threshold}} = 10$  and  $\theta_{\text{threshold}} = 20^\circ$ . An

<sup>1</sup> <https://github.com/hithmh/Meta-Koopman>



episode terminates prematurely if  $|\theta| > \theta_{\text{threshold}}$ . The stage cost for control performance evaluation is given by  $c = 0.1 * (x/x_{\text{threshold}})^2 + (\theta/\text{threshold})^2$ . In the Cartpole system, the length of the pole  $l_p$  and the mass of the pole  $m_p$  are considered uncertain, with  $l_p \in [0.1m, 1.0m]$  and  $m_p \in [0.01kg, 0.2kg]$ . Their nominal values are  $l_p = 0.5m$  and  $m_p = 0.1kg$ . The value of  $\alpha$  is set to 1.995, and the weighting matrices are  $Q = \text{diag}([0.01, 0, 1, 0.2])$ , where  $\text{diag}(\cdot)$  denotes constructing a diagonal matrix with the given vector;  $R = 0.01$ .

### 5.1.2 Gene regulatory network (GRN)

MAKO is also applied to a biological gene regulatory network (GRN), which constitutes a synthetic three-gene regulatory network characterized by the oscillatory dynamics of mRNAs and proteins [39]. The state vector is  $[m_1, m_2, m_3, p_1, p_2, p_3]^T$ , where  $m_{1,2,3}$  denotes the concentration of mRNA for the corresponding genes and  $p_{1,2,3}$  denotes the concentration of the proteins. The control input  $u$  will be executed through light control signals capable of triggering gene expression via the activation of their photosensitive promoters. The controller is expected to maintain the concentration of protein 1  $p_1$  to 6. A more detailed description of the controlled GRN system is referred to [39]. In the GRN system, the dissociation constant  $K$  and the input scalar  $b_1$  to protein 1 are assumed to be uncertain,  $K \in [2, 8]$  and  $b_1 \in [3, 7]$ . Their nominal values are  $K = 5$  and  $b_1 = 5$ . The value of  $\alpha$  is 1.1, and the weighting matrices are  $Q = \text{diag}([0, 0, 0, 1, 0, 0])$ ,  $R = \text{diag}([0.01, 0.01, 0.01])$ .

### 5.1.3 Reactor-separator chemical process

Finally, we apply MAKO to a chemical process that involves two continuously stirred tank reactors and a flash tank separator [40]. A detailed description of this process can be found in [40]. The state vector encompasses  $[X_{A1}, X_{B1}, T_1, X_{A2}, X_{B2}, T_2, X_{A3}, X_{B3}, T_3]^T$ , including the mass fractions of A and B denoted by  $X_{Ai}$  and  $X_{Bi}$ , and the temperatures  $T_i$ , 1, 2, 3, across the three vessels. The control objective is to maintain the concentrations of A and B at a steady-state level  $x_s = [0.18, 0.67, 480.3 \text{ K}, 0.19, 0.65, 472.8 \text{ K}, 0.06, 0.67, 474.9 \text{ K}]^T$ .

The heating inputs are constrained within  $[0, 0, 0]^T \times 10^6 \text{ kJ/h} \leq u \leq [4.87, 1.68, 4.87]^T \times 10^6 \text{ kJ/h}$ . Initially, the state is distributed uniformly within the region of  $[0.8x_s, 1.2x_s]$ . The temperature of the feed stream temperatures to reactors 1 and 2 are assumed to be uncertain,  $T_{10} \in [150K, 450K]$  and  $T_{20} \in [150K, 450K]$ . Their nominal values are  $T_{10} = 300K$  and  $T_{20} = 300K$ . The value of  $\alpha$  is 1.98, and the weighting matrices are  $Q = \text{diag}([1, 1, 0, 1, 1, 0, 1, 1, 0])$ ,  $R = \text{diag}([1, 1, 1]) \times 10^{-3}$ .

The MAKO model is trained on a multi-modal dataset covering different task settings with random sampled inputs. The collected data is normalized by using the mean and standard deviation vectors. In our evaluation, the uncertain parameters are first uniformly sampled from

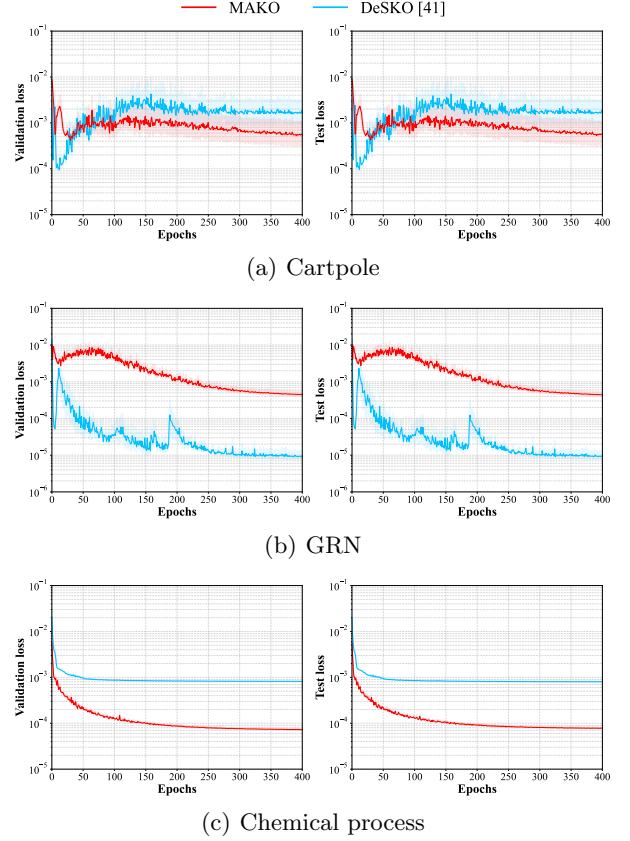


Fig. 3. Cumulative prediction errors calculated on both the validation and test datasets. The Y-axis represents the cumulative mean-squared prediction error in a logarithmic scale over 16 time steps, while the X-axis denotes the training epochs. The shaded area illustrates the confidence interval, corresponding to one standard deviation, calculated over 5 random initializations.

the parameter space, and then the inputs are uniformly sampled from the input space at each time instant to excite the system.

### 5.2 Baseline for comparison

We compare the performance of MAKO with a competitive baseline, the deep stochastic Koopman operator (DeSKO) method [41]. DeSKO can provide good modeling and control performance in various systems and has been shown to be robust to system uncertainties. DeSKO models are trained on datasets collected using nominal parameter settings. MAKO models, on the other hand, are trained on a meta-dataset comprising sub-datasets collected using randomly sampled parameter settings. The hyperparameters of MAKO are shown in Table 1 in the Appendix.

### 5.3 Modeling

In this part, we first evaluate the modeling performance of MAKO. For both MAKO and DeSKO, 5 models



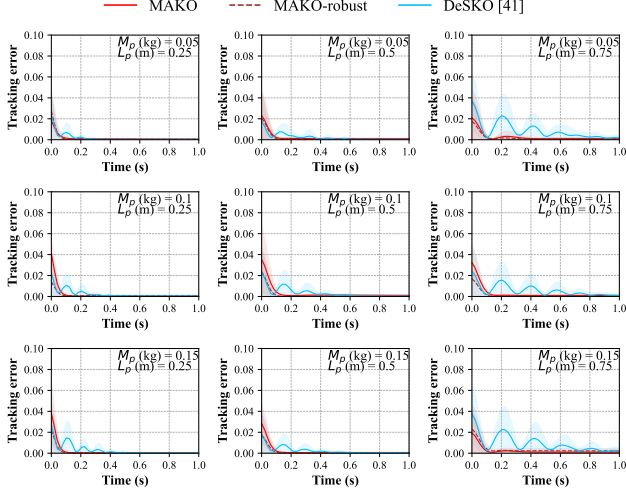


Fig. 4. The trajectories of the tracking errors given by MAKO and DeSKO on Cartpole under 9 parameter settings.

are randomly initialized and trained. Both models are trained for 400 epochs. During each epoch, the models are trained using the Adam optimizer with mini-batches of 128 data points until the entire training dataset for each model has been traversed. The  $l_2$  norm of the prediction error on the validation and test datasets in each epoch is presented in Fig. 3. As observed from Fig. 3, MAKO demonstrates good modeling performance across different benchmark systems. The average prediction errors of MAKO models over a 16-step horizon are less than  $10^{-2}$ . The modeling performance of MAKO is consistent on both the validation set and the test set. While both MAKO and DeSKO provide high modeling accuracy, DeSKO outperforms MAKO in the GRN system. This may be attributed to two factors. 1) The MAKO model is trained and evaluated on a multi-modal dataset containing diverse task settings, while DeSKO is trained specifically under the nominal parameter setting, which allows it to specialize in modeling the GRN dynamics under those specific conditions. 2) MAKO prioritizes adaptability and generalization, which may slightly compromise its predictive accuracy for a specific task. Moreover, MAKO models surpass DeSKO models in both the Cartpole system and the chemical process. MAKO also exhibits consistent prediction accuracy and low variances across different parameter initializations.

#### 5.4 Control

In this subsection, we examine the performance of the proposed MAKO-based controller. We evaluate both controllers, designed using nominal adaptation and robust adaptation, respectively, and refer to them as MAKO and MAKO-Robust. For each considered system, we uniformly take 9 sets of parameters from the respective parameter space described in Section 5.1. These parameter settings were not encountered by MAKO during its training phase, which can showcase the generalization ability of MAKO. On the other hand,

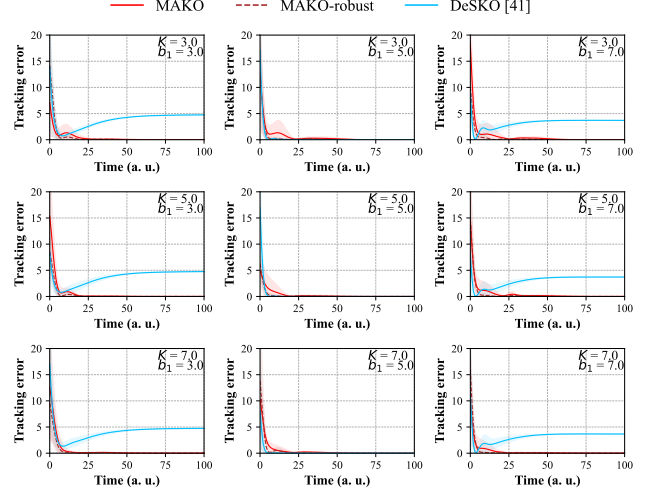


Fig. 5. The trajectories of the tracking errors given by MAKO and DeSKO on GRN under 9 parameter settings.

the nominal parameter setting, which was encountered during DeSKO’s training, is also included for comparison purposes. The cumulative tracking errors and the trajectories of the norm of tracking errors of MAKO and DeSKO for the three considered systems are shown in Fig. 4, Fig. 5, and Fig. 6, respectively. Fig. 4 to Fig. 6 demonstrate that MAKO achieves good control performance in all three benchmark systems under various parameter settings. Its cumulative tracking error is also lower than or comparable to that of DeSKO. In the Cartpole example, DeSKO achieves stabilizing performance with higher cumulative costs compared to MAKO. For the second system, DeSKO accurately tracks the given reference in GRN under 3 sets of parameter settings. In the chemical process example, DeSKO fails to stabilize the tracking error in all parameter settings. Compared to the nominal MAKO-based MPC, the MPC leveraging the robust adaptation scheme exhibits faster and more stable transient behavior, while achieving comparable or smaller steady-state tracking errors. In the current work, the simulation is conducted on a computer equipped with an i7-12700 2.10 GHz CPU. For the Cartpole system, the MAKO-robust framework achieved an average computation time of 0.0203 seconds per time step, which demonstrates its suitability for real-time control applications.

**Remark 4** While Assumption 3 can be difficult to verify on the benchmark systems, MAKO provides good performance in all three case studies, demonstrated by high modeling accuracy and robust control performance despite the lack of strict forward invariance. The simulation results demonstrate the framework’s ability to address real-world complexities and suggest its potential to handle a broader range of systems where theoretical guarantees on invariance may not strictly apply.

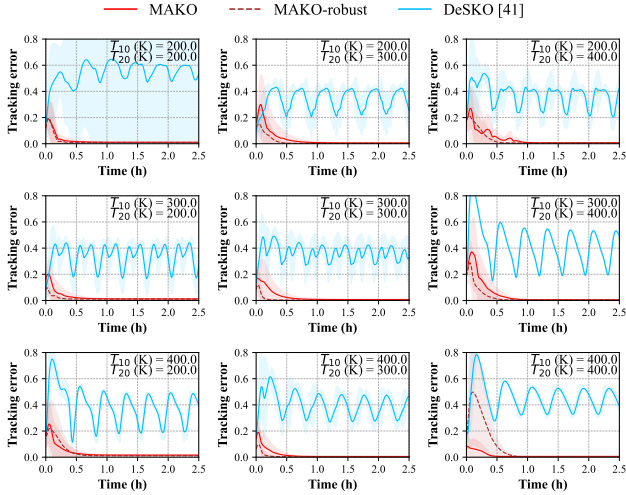


Fig. 6. The trajectories of the tracking errors given by MAKO and DeSKO on the chemical process under 9 parameter settings.

## 6 Conclusion

In this paper, meta-learning and the Koopman operator were integrated for the first time to develop a multimodal modeling approach for parametrically uncertain nonlinear systems. An adaptation scheme was designed to refine the meta-trained Koopman operator with on-line data, and the convergence of parameter approximation and state prediction errors was proven under mild assumptions. Based on the proposed MAKO model, an AMPC scheme was proposed; this control scheme ensures the stability of the closed-loop system, in the presence of previously seen and unseen parameter settings. Through extensive simulation evaluations, we demonstrated that the proposed MAKO modeling and control framework can outperform the baseline methods.

We identify the following potential topics for future research: 1) While MAKO demonstrated good performance in simulations, applying this method to real-world systems with parametric uncertainties would be of interest in future research. 2) A formal analysis of persistent excitation (PE) requirements could be crucial for examining training convergence. Specifically, understanding how trajectory richness and PE influence the quality of the learned Koopman operator and its ability to generalize across unseen tasks would provide valuable theoretical insights. 3) A more systematic investigation of trajectory length in relation to the meta-learning of Koopman operators is an important direction of future research. 4) Extending the proposed approach to higher-dimensional systems would be another interesting topic for future exploration.

## References

[1] Y. Hong, J. Wang, and D. Cheng, “Adaptive finite-time control of nonlinear systems with parametric uncertainty,”

Table 1

Hyperparameters of MAKO

Hyperparameters	Cartpole	GRN	Chemical process
Observable dimension	128	128	256
Trajectory length	250	400	500
Noise upper bound $\epsilon_w$	$10^{-6}$	$10^{-6}$	$10^{-8}$
Noise upper bound $\epsilon_v$	$10^{-4}$	$10^{-4}$	$10^{-8}$
Number of sub-datasets $N$	10	10	20
Size of sub-dataset $\mathcal{D}_i$	$5 \times 10^4$		
Batch Size	128		
Learning rate	$10^{-4}$		
Prediction horizon $H$	16		
Structure of $\psi_\theta(\cdot)$	(128,128)		
Activation function	ReLU		
$l_2$ norm regularization coefficient	0.001		

*IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 858–862, 2006.

- [2] G. Tao, “Multivariable adaptive control: A survey,” *Automatica*, vol. 50, no. 11, pp. 2737–2764, 2014.
- [3] 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.
- [4] H. Fukushima, T.-H. Kim, and T. Sugie, “Adaptive model predictive control for a class of constrained linear systems based on the comparison model,” *Automatica*, vol. 43, no. 2, pp. 301–308, 2007.
- [5] K. Zhang and Y. Shi, “Adaptive model predictive control for a class of constrained linear systems with parametric uncertainties,” *Automatica*, vol. 117, p. 108974, 2020.
- [6] D. Q. Mayne and H. Michalska, “Adaptive receding horizon control for constrained nonlinear systems,” in *IEEE Conference on Decision and Control*, 1993, pp. 1286–1291.
- [7] V. Adetola, D. DeHaan, and M. Guay, “Adaptive model predictive control for constrained nonlinear systems,” *Systems & Control Letters*, vol. 58, no. 5, pp. 320–326, 2009.
- [8] K. Zhang, C. Liu, and Y. Shi, “Self-triggered adaptive model predictive control of constrained nonlinear systems: A min-max approach,” *Automatica*, vol. 142, p. 110424, 2022.
- [9] D. DeHaan and M. Guay, “Adaptive robust MPC: A minimally-conservative approach,” in *American Control Conference*, 2007, pp. 3937–3942.
- [10] B. Xu, A. Suleman, and Y. Shi, “A multi-rate hierarchical fault-tolerant adaptive model predictive control framework: Theory and design for quadrotors,” *Automatica*, vol. 153, p. 111015, 2023.
- [11] B. O. Koopman, “Hamiltonian systems and transformation in Hilbert space,” *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.
- [12] P. J. Schmid, “Dynamic mode decomposition of numerical and experimental data,” *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.
- [13] 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 Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 10, p. 103111, 2017.
- [14] Z. Zeng, Z. Yue, A. Mauroy, J. Gonçalves, and Y. Yuan, “A sampling theorem for exact identification of continuous-time nonlinear dynamical systems,” *IEEE Transactions on Automatic Control*, vol. 69, no. 12, pp. 8402–8417, 2024.
  - [15] J. Morton, A. Jameson, M. J. Kochenderfer, and F. Witherden, “Deep dynamical modeling and control of unsteady fluid flows,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
  - [16] E. Yeung, S. Kundu, and N. Hodas, “Learning deep neural network representations for Koopman operators of nonlinear dynamical systems,” in *American Control Conference*, Jul. 2019, pp. 4832–4839.
  - [17] Y. Han, W. Hao, and U. Vaidya, “Deep learning of Koopman representation for control,” in *IEEE Conference on Decision and Control*, 2020, pp. 1890–1895.
  - [18] O. Azencot, N. B. Erichson, V. Lin, and M. Mahoney, “Forecasting sequential data using consistent Koopman autoencoders,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 475–485.
  - [19] H. Shi and M. Q.-H. Meng, “Deep Koopman operator with control for nonlinear systems,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7700–7707, 2022.
  - [20] S. A. Deka, A. M. Valle, and C. J. Tomlin, “Koopman-based neural Lyapunov functions for general attractors,” in *IEEE Conference on Decision and Control*, 2022, pp. 5123–5128.
  - [21] M. Han, Z. Li, X. Yin, and X. Yin, “Robust learning and control of time-delay nonlinear systems with deep recurrent Koopman operators,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 3, pp. 4675–4684, 2024.
  - [22] Z. Li, M. Han, D.-N. Vo, and X. Yin, “Machine learning-based input-augmented Koopman modeling and predictive control of nonlinear processes,” *Computers & Chemical Engineering*, vol. 191, p. 108854, 2024.
  - [23] W. Hao, B. Huang, W. Pan, D. Wu, and S. Mou, “Deep Koopman learning of nonlinear time-varying systems,” *Automatica*, vol. 159, p. 111372, 2024.
  - [24] Y. Wei, A. W.-K. Law, and C. Yang, “Probabilistic optimal interpolation for data assimilation between machine learning model predictions and real time observations,” *Journal of Computational Science*, vol. 67, p. 101977, 2023.
  - [25] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, “Meta-learning in neural networks: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.
  - [26] I. Molybog and J. Lavaei, “When does MAML objective have benign landscape?” in *IEEE Conference on Control Technology and Applications*, 2021, pp. 220–227.
  - [27] N. Musavi and G. E. Dullerud, “Convergence of gradient-based MAML in LQR,” in *IEEE Conference on Decision and Control*, 2023, pp. 7362–7366.
  - [28] L. F. Toso, D. Zhan, J. Anderson, and H. Wang, “Meta-learning linear quadratic regulators: A policy gradient MAML approach for the model-free LQR,” *arXiv preprint arXiv:2401.14534*, 2024.
  - [29] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, “Learning to adapt in dynamic, real-world environments through meta-reinforcement learning,” in *International Conference on Learning Representations*, 2018.
  - [30] D. G. McClement, N. P. Lawrence, J. U. Backström, P. D. Loewen, M. G. Forbes, and R. B. Gopaluni, “Meta-reinforcement learning for the tuning of PI controllers: An offline approach,” *Journal of Process Control*, vol. 118, pp. 139–152, 2022.
  - [31] M. Korda and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” *Automatica*, vol. 93, pp. 149–160, 2018.
  - [32] J. L. Proctor, S. L. Brunton, and J. N. Kutz, “Generalizing Koopman theory to allow for inputs and control,” *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.
  - [33] X. Zhang, W. Pan, R. Scattolini, S. Yu, and X. Xu, “Robust tube-based model predictive control with Koopman operators,” *Automatica*, vol. 137, p. 110114, 2022.
  - [34] B. Zhu and X. Xia, “Adaptive model predictive control for unconstrained discrete-time linear systems with parametric uncertainties,” *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 3171–3176, 2015.
  - [35] L. C. Iacob, R. Tóth, and M. Schoukens, “Koopman form of nonlinear systems with inputs,” *Automatica*, vol. 162, p. 111525, 2024.
  - [36] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
  - [37] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, *Nonlinear model predictive control*. Springer, 2017.
  - [38] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
  - [39] A. Sootla, N. Strelkowa, D. Ernst, M. Barahona, and G.-B. Stan, “On periodic reference tracking using batch-mode reinforcement learning with application to gene regulatory network control,” in *IEEE Conference on Decision and Control*, 2013, pp. 4086–4091.
  - [40] J. Liu, D. M. de la Peña, B. J. Ohran, P. D. Christofides, and J. F. Davis, “A two-tier architecture for networked process control,” *Chemical Engineering Science*, vol. 63, no. 22, pp. 5394–5409, 2008.
  - [41] M. Han, K. Wong, J. Euler-Rolle, L. Zhang, and R. K. Katschmann, “Robust learning-based control for uncertain nonlinear systems with validation on a soft robot,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 1, pp. 510–524, 2025.