Learning-based moving horizon autonomous control of a chemical reactor

Bei Sun[a], Peng Kong[a], Johan D. le Roux[b], Ian K. Craig[b], Mingfang He[c], Chunhua Yang[a,*]

[a]*School of Automation, Central South University, Changsha, 410083 China*
[b]*Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa*
[c]*School of Electronic Information and Physics, Central South University of Forestry and Technology, Changsha, 410004 China*

Abstract

This paper proposes a learning-based moving horizon autonomous control of a chemical reactor (LMHAC) approach for chemical reactor with multiple operating conditions. In the proposed LMHAC scheme, model-based control, model-free control and process modeling are integrated in a moving horizon framework. A control switching logic makes a selection between model predictive control (MPC) and adaptive dynamic programming (ADP) depending on whether the model parameters are known or unknown under the current operating condition. To be compatible with the moving horizon framework, the conventional ADP is fitted into a finite horizon composed of two different stages, namely a learning stage and a control-identification stage. In the learning stage, a constrained finite-horizon ADP (CFADP) first learns an approximated optimal controller from the collected input-state information pair generated by an initial admissible control. In the control-identification stage, the approximated optimal control is applied to the process to generate a sequence of input-state information pairs which is then utilized in turn to identify the unknown model parameters. The LMHAC framework is capable of providing the optimal or nearly optimal control for different operating conditions online and incrementally enlarge the known domain of system dynamics. The feasibility and performance of the proposed approach are illustrated via a case study.

Keywords: autonomous control; moving horizon; adaptive dynamic programming (ADP); process control; parameter identification

1. Introduction

Modeling and optimal control are fundamental tools to predict process behavior and adjust the manipulated variables of a chemical reactor, which is the primary production unit of a complex industrial process. They usually exist as counterparts in process control applications. As for the model, it is a mathematical representation of system dynamics and is crucial in design, model-based control, estimation and monitoring applications, e.g., steady-state optimization [1, 2, 3], plant design [4, 5, 6], state estimation [7, 8, 9], soft sensoring [10, 11, 12], process monitoring [13, 14, 15], fault detection and diagnosis (FDD) [16, 17, 18], control performance evaluation [19, 20], etc. As for control, model-based control and model-free control are two different approaches with their application contexts. Model-based control, e.g., model predictive control (MPC), is now the standard solution for multivariable control in many process industries [21, 22, 23]. The performance of model-based control may deteriorate when the process model is inaccurate. When a process model is not available or the model parameters are unknown, model-free control that can learn the optimal controller from the operation data can be applied as an alternative, e.g., reinforcement learning (RL)/adaptive dynamic programming (ADP) [24, 25, 26, 27, 28, 29]. Recent advancements have demonstrated the effectiveness of ADP in various practical scenarios. For instance, adaptive neural control approaches based on disturbance observers have been successfully applied to ultra-supercritical steam systems with output constraints, showcasing the practical benefits of ADP and NN integration in complex industrial processes [30]. Furthermore, event-triggered, IRL-based decentralized fault-tolerant control schemes have been developed for large-scale interconnected systems, further highlighting the potential and significance of ADP algorithms in modern intelligent process control [31]. Considering the transition of the process industry to smart and optimal manufacturing, a hypothesis is proposed: Can system modeling and optimal control (model-based control and model-free control) be integrated in an interactive and systematic framework to support the autonomous process operation of a chemical reactor?

RL/ADP is based on adaptive control, dynamic programming, and optimal control [32, 33]. However, relevant work on RL/ADP can be related to dual adaptive control with *dual* control objectives [34, 35, 36]. The dual adaptive control problem was both studied in the context of MPC [37, 38, 39] and RL/ADP [40, 41, 42]. In recent years, the comparison and integration of model-based control and model-free control have prompted the interest of researchers [43, 44, 45]. Görges [46] provided a comprehensive review of the relations between

*Corresponding author
Email addresses: sunbei@csu.edu.cn (Bei Sun), 224601032@csu.edu.cn (Peng Kong), derik.leroux@up.ac.za (Johan D. le Roux), ian.craig@up.ac.za (Ian K. Craig), t20162306@csuft.edu.cn (Mingfang He), ychh@csu.edu.cn (Chunhua Yang)

MPC and ADP, as well as the advantages and disadvantages of the two approaches. Xu *et al.* (2018) [47] designed a learning-based predictive control (LPC) scheme for adaptive optimal control of nonlinear discrete systems under stochastic disturbances. The main idea of this approach is to decompose the infinite-horizon optimal control problem into a series of finite-horizon problems with terminal constraints, which are then handled using ADP. Dong et al. (2018) [48] proposed a functional MPC approach based on ADP for the optimal control of nonlinear discrete-time systems with control constraints and disturbances. This approach uses ADP with a terminal constraint instead of MPC in each finite control interval. These two novel approaches embed ADP in the MPC's moving horizon framework, using ADP instead of MPC to derive the optimal control moves in each control interval. Since ADP does not require full knowledge of the model parameters, it enables data-driven optimal control even under model uncertainty.

An actual industrial process has a wide range of operating conditions. A first-principles dynamics model of a chemical reactor can be established using, e.g., conservation laws and reaction kinetics; however, the model parameters vary with operating conditions. The model parameters can usually be identified and validated for some operating conditions. However, for other operating conditions, the model parameters may not be available due to, for example, limited or no data samples that can be used for model identification. In summary, the process dynamics can be described using a defined model structure and varying model parameters, which may be unknown in some operating conditions. Therefore, one can select model-based or model-free control approaches based on the model parameters' uncertainty level.

Existing methods for integrating model-based and model-free control consist mainly of decomposing the original infinite horizon ADP into a series of finite horizon problems to replace MPC [47, 48]. However, MPC is often not utilized, which is more familiar to process control engineers and can be applied when the process dynamics are known. Moreover, other approaches that combine model-based and model-free strategies tend to rely on predefined switching logic or offline training, and often require significant prior model information or human intervention to handle changes in operating conditions. Therefore, a switching mechanism should be designed to autonomously select between MPC and ADP according to the uncertainty level of the model parameters. Specifically, the aim is to use the 'input-state' information collected while using ADP to identify the unknown parameters in the system model. *An integrated MPC-ADP controller could benefit significantly from plant model parameters that are gradually identified and updated over an increasingly expanding plant operating region.*

Subbarao *et al.* (2016) [49] developed a three-phase (stabilize-optimize-identify) adaptive optimal control and system identification scheme for unknown linear systems. A Model Reference Adaptive Control (MRAC) approach was used in the first stabilization phase to stabilize the system. ADP was applied to derive the optimal feedback gain in the second optimization phase. In the final identification phase, using the algebraic Ricatti equation as a bridge, the information obtained during the optimization phase was utilized to estimate the system and input matrix. However, the result in [49] is proposed for linear systems without terminal and control constraints, which cannot be applied directly to nonlinear processes.

This paper's main contributions and theoretical innovations are the development of a learning-based moving horizon autonomous control framework, a constrained finite-horizon ADP, and unknown model parameter identification based on the Hamilton Jacobi Bellman (HJB) equation. The proposed scheme integrates MPC, ADP, and process modeling into a unified framework. Unlike many existing approaches, it explicitly incorporates input constraints into the control and learning phases. By embedding the moving horizon strategy of MPC, the framework dynamically adapts to time-varying process dynamics, ensuring that the control policy remains responsive and robust as operating conditions evolve. This seamless combination enables the controller to autonomously switch between model-based and model-free strategies, optimizing performance across various operating scenarios while efficiently leveraging process knowledge whenever available. Due to the slow-varying dynamics of many complex industrial processes, the system dynamics can be considered constant during one optimization cycle of the moving horizon framework. At the beginning of each optimization cycle, a controller switching logic is used to select the appropriate controller. If the model parameters are considered known under the current operating condition, then the MPC is selected. Otherwise, the ADP is selected to replace the MPC. In order to realize the integration of ADP and MPC, three main problems are studied. Firstly, for the stable switching from MPC to ADP, a robust adaptive controller is designed as an initial admissible control to guarantee closed-loop stability. Secondly, since there is a prediction horizon and control horizon in each MPC control interval, the ADP control interval is divided into two stages: (i) a learning stage and (ii) a control-identification stage. In the learning stage, a finite horizon ADP algorithm for a nonlinear system with terminal constraints on system states is designed to fit the limited control interval. Thirdly, the approximated optimal control move is applied to the process in the control-identification stage. The 'input-state' information pair collected from the approximated optimal trajectory is used simultaneously to identify the unknown model parameters. Therefore, MPC will be selected for the next occurrence of the same operating condition as the model parameters are now known. As the optimization cycle moves forward, the proposed approach incrementally obtains the knowledge required by a model-based controller for the entire operation range. By integrating model-based and model-free control strategies, the proposed framework enables the reactor to autonomously adapt to varying and previously unknown operating conditions. This leads to improved production efficiency, reduction in energy consumption, and enhanced process safety. Furthermore, adaptive identification and control facilitate continuous optimization and reliable performance in the face of process uncertainties.

The remainder of this paper is organized as follows. Section

2 analyzes the optimal control problem of a process with multiple operating conditions, and the comparison between MPC and ADP is given. Then, the learning-based moving horizon autonomous control strategy is introduced in Section 3. Case studies are used in Section 4 to demonstrate the feasibility and performance of the proposed approach. Section 5 concludes the study.

2. Problem analysis

2.1. Optimal control of a chemical reactor with multiple operating conditions

The dynamics of a chemical reactor can be affected by both feed conditions (or inlet conditions, e.g., feed rate/composition) and reaction conditions (e.g., reaction temperature, pH, additive dosage, catalyst dosage). Therefore, Sun *et al.* (2020) [50] proposed a comprehensive state space descriptive system to describe the process dynamics. It is a three-dimensional space that includes inlet conditions, reaction conditions, and output states (technical indexes) (see Fig. 1 for illustrative purposes). A process exhibits different dynamics under different combinations of inlet and reaction conditions. On the one hand, the underlying physicochemical rules of a process can be modeled using conservation laws and reaction kinetics, which differential equations can describe with a determined structure. On the other hand, however, a change in the inlet or reaction conditions affects the type, proportion, and environment in which the main and side reactions occur. As a result, the model parameters vary under different operating conditions (Fig. 1). The precise identification of these model parameters relies on the quality and quantity of data samples that are generally unavailable for all operating conditions. Moreover, changing a supplier or operating strategy can introduce new operating conditions for which the model parameters cannot be identified when sufficient data samples are unavailable.

The dynamics of a generic chemical reactor can be expressed as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{u}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{1}$$

in which, $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of system states, $\boldsymbol{u} \in \mathbb{R}^m$ is the vector of usually constrained control inputs, $\boldsymbol{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are locally Lipschitz functions whose structures are fixed and can be derived based on mass/energy balances and the underlying reaction kinetics, $\boldsymbol{\theta}_{\mathrm{f}} \in \mathbb{R}^{N_{\mathrm{f}}}$ and $\boldsymbol{\theta}_{\mathrm{g}} \in \mathbb{R}^{N_{\mathrm{g}}}$ are vectors of the model parameters in $\boldsymbol{f}(\cdot)$ and $\mathbf{g}(\cdot)$, respectively. $\boldsymbol{\theta}_{\mathrm{f}} \in \mathbb{R}^{N_{\mathrm{f}}}$ and $\boldsymbol{\theta}_{\mathrm{g}} \in \mathbb{R}^{N_{\mathrm{g}}}$ take different values under different operating conditions.

The optimal control problem for system (1) can be formulated as:

$$\begin{aligned} \min_{\boldsymbol{u}} \quad & J(\boldsymbol{x}(0), \boldsymbol{u}) = \int_{t_0}^{t_f} \gamma(\boldsymbol{x}(t), \boldsymbol{u}(t))\mathrm{d}t \\ s.t. \quad & \begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{u} \\ \boldsymbol{u}_{\mathrm{low}} \leqslant \boldsymbol{H}_u(\boldsymbol{u}) \leqslant \boldsymbol{u}_{\mathrm{up}} \\ \boldsymbol{x}_{\mathrm{low}} \leqslant \boldsymbol{H}_x(\boldsymbol{x}) \leqslant \boldsymbol{x}_{\mathrm{up}} \\ \boldsymbol{x}(0) = \boldsymbol{x}_0 \end{cases} \end{aligned} \tag{2}$$

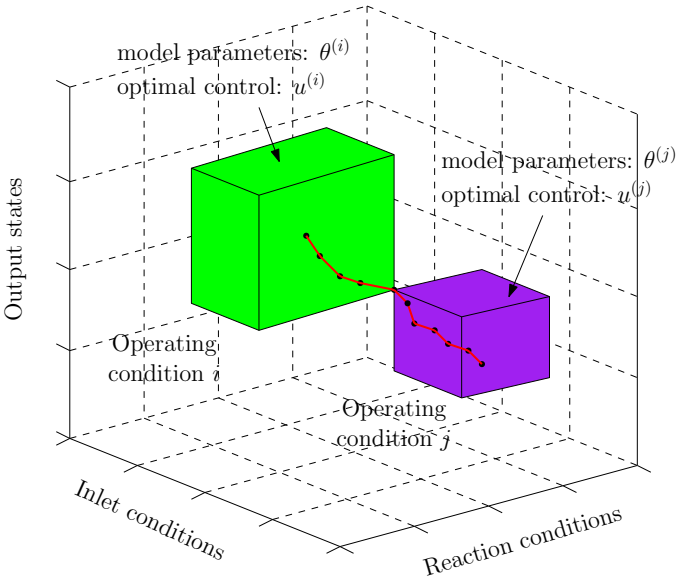

Figure 1: Illustration of comprehensive state space [50]

where $[t_0, t_f]$ is the horizon of interest, $\gamma(\cdot)$ is the stage cost function, specifying the instantaneous cost associated with state $x(t)$ and control input $u(t)$, and serves as the objective function to be minimized over the horizon. $\boldsymbol{H}_u(\cdot)$ and $\boldsymbol{H}_x(\cdot)$ are user-defined functions for handling input and state constraints. They can be used to represent scaling, normalization, or other (possibly nonlinear) transformations applied to $\boldsymbol{u}$ and $\boldsymbol{x}$, respectively. This notation allows the expression of general input and state constraints in a unified form. $\boldsymbol{u}_{\mathrm{low}}$ and $\boldsymbol{u}_{\mathrm{up}}$ are the lower and upper bounds of $\boldsymbol{H}_u(\cdot)$, respectively, $\boldsymbol{x}_{\mathrm{low}}$ and $\boldsymbol{x}_{\mathrm{up}}$ are the lower and upper bounds of $\boldsymbol{H}_x(\cdot)$, respectively. $\boldsymbol{x}_0$ is the initial value of $\boldsymbol{x}$. In this paper, $H_u(\cdot)$ and $H_x(\cdot)$ are defined as:

$$\begin{aligned} H_u(u) &= \frac{2u - (u_{\mathrm{up}} + u_{\mathrm{low}})}{u_{\mathrm{up}} - u_{\mathrm{low}}}, \\ H_x(x) &= \frac{2x - (x_{\mathrm{up}} + x_{\mathrm{low}})}{x_{\mathrm{up}} - x_{\mathrm{low}}}, \end{aligned} \tag{3}$$

where u_{low}, u_{up}, x_{low}, and x_{up} are the bounds specified in the constraint expressions above.

As the process moves continuously in the comprehensive state space with multiple operating conditions (see Fig. 1), the model parameters can take different values under different operating conditions. Moreover, the model parameters are unknown under certain operating conditions. Therefore, it is necessary to construct a framework that integrates model-based and model-free control approaches to derive optimal or suboptimal controllers for the entire operation range.

2.2. Differences and connections between model-based and model-free optimal control approaches

MPC and ADP are typical and distinct approaches to solving an optimal control problem. These two approaches have differences and connections. On the one hand, MPC is model-based. It circumvents the exhaustive online solution of the

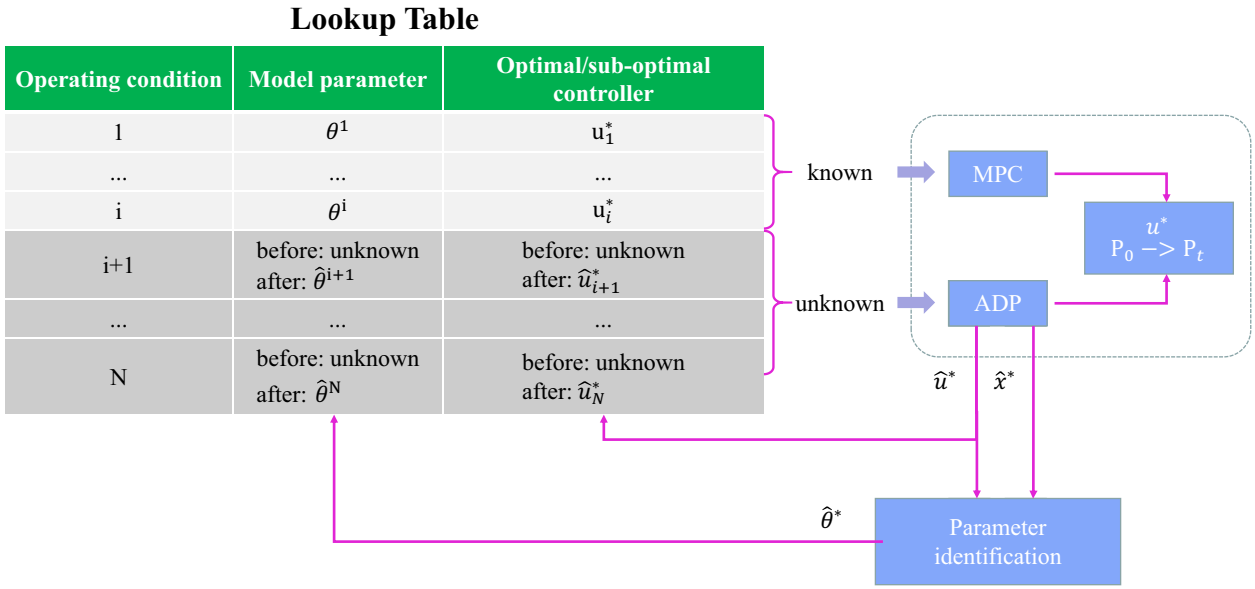

Figure 2: Differences and connections between model-based control and model-free control

HJB equation by repeatedly solving the corresponding closed-loop or open-loop optimal control problem. The application of MPC requires a validated system model. However, real-world processes usually operate over wide regions, making obtaining a comprehensive and precise process model expensive or even impossible. Therefore, the performance of MPC, which is usually based on a system model established offline, cannot be guaranteed if the model parameters deviate too far from nominal values under certain operating conditions. On the other hand, ADP can achieve model-free control. It learns the closed-loop optimal or nearly optimal controller from the "excitation-response" data collected by applying an initial admissible control to the system. ADP can, therefore, be used as an alternative to MPC when the model parameters are unknown.

Regardless of the differences stated above, the objectives of MPC and ADP are the same, i.e., to approximate the optimal control which forces the system to move from an initial operating point P_0 to a desired one P_t, as shown in Fig. 2. Although ADP is often adopted as an alternative to MPC when the full system model is unavailable, a theoretical connection between MPC and ADP can still be established through the HJB equation. When the functional forms of the system dynamics and input matrices are known, and sufficient data for the system states and control inputs are available, the unknown model parameters can be estimated by leveraging the relationship implied by the HJB equation. Specifically, process data collected during ADP-based control can be utilized to approximate optimal policies and identify the initially unknown model parameters. This approach enables the incremental construction of a lookup table that associates operating conditions, model parameters, and controllers. Then, if the same operating condition emerges again, model-based control, which has a lower computational burden than ADP, can be applied. In addition, the identified process model can be used to support various other model-based applications.

3. Learning-based moving horizon autonomous control based on integration of MPC, ADP and process modeling

This section proposes a learning-based moving horizon autonomous control (LMHAC) scheme for chemical reactors with multiple operating conditions. As shown in Fig. 1, varying process conditions form a trajectory in comprehensive state space that spans different operating conditions. LMHAC runs through this comprehensive state space in a moving horizon manner, and based on the slow time-varying nature of industrial processes, the process dynamics can be regarded as constant within a control interval of appropriate length. It should be noted that this assumption may not always be valid for processes with rapid or abrupt dynamics. In such cases, the length of the control interval can be adaptively shortened to better capture the process dynamics.

In each interval $[t_i, t_{i+1}]$, the aim of LMHAC is twofold:

(i) Approximate a solution for the following finite-horizon optimal control problem:

$$\min_{\boldsymbol{u}} \; J(\boldsymbol{x}(t_i), \boldsymbol{u}) = \int_{t_i}^{t_{i+1}} \gamma(\boldsymbol{x}(t), \boldsymbol{u}(t))\, \mathrm{d}t + F_{\text{terminal}}(\boldsymbol{x}(t_{i+1})), \tag{4}$$

the terminal error penalty function is defined as $F_{\text{terminal}}(\boldsymbol{x}(t_{i+1})) = \|\boldsymbol{x}(t_{i+1}) - \boldsymbol{x}^\star\|^2_{\mathbf{Q}_f}$, with $\mathbf{Q}_f$ being a positive semi-definite weighting matrix and $\boldsymbol{x}^\star$ being the desired steady-state (set-point) for the current operating condition.

(ii) Estimate the model parameters if they are unknown under the current operating condition by solving the following

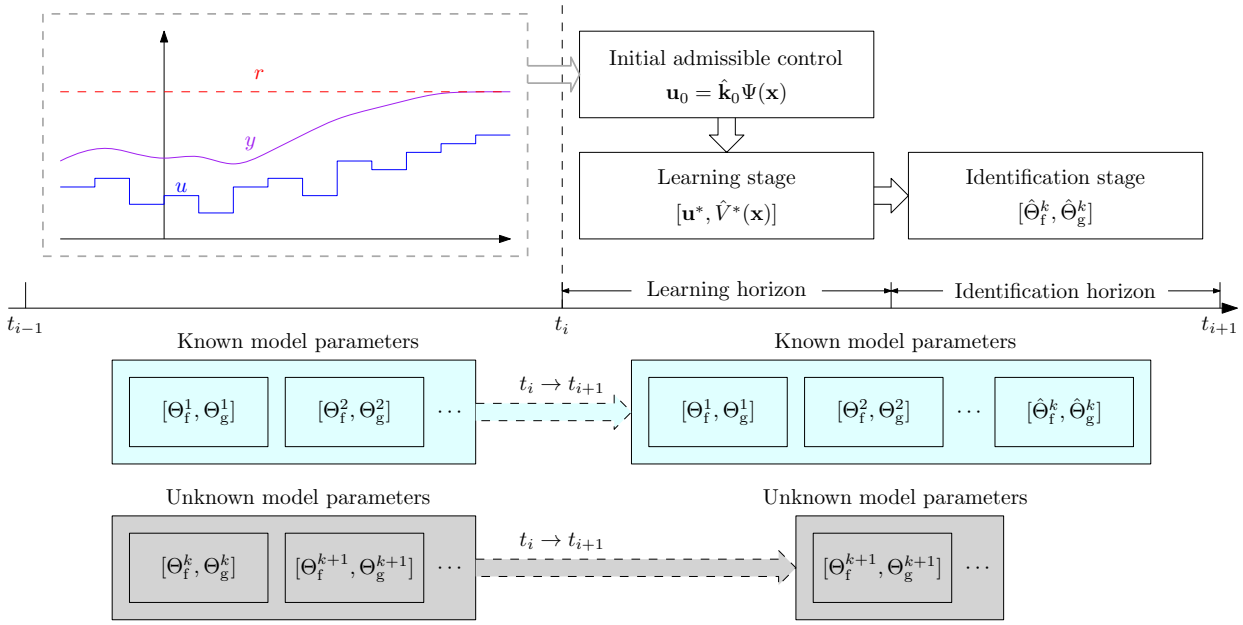

Figure 3: Basic idea of the learning-based moving horizon autonomous control strategy

optimization problem:

$$\min_{\hat{\boldsymbol{\theta}}_{\mathrm{f}}, \hat{\boldsymbol{\theta}}_{\mathrm{g}}} J_{\mathrm{para}}(\hat{\boldsymbol{\theta}}_{\mathrm{f}}, \hat{\boldsymbol{\theta}}_{\mathrm{g}}) = \sum_{k=1}^{N_5} \xi_k^2, \tag{5}$$

where $\hat{\boldsymbol{\theta}}_{\mathrm{f}}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{g}}$ are the estimated model parameters, and ξ_k denotes the approximation error at sample k.

The continuous running of LMHAC incrementally shrinks the space of operating conditions with unknown model parameters. In addition, it can provide an approximated optimal controller, i.e., a stable controller, under any type of operating condition by switching between model-based control and model-free control. As a result, both the approximated optimal controller and model parameters for different operating conditions can be obtained incrementally (Fig. 3). The obtained model parameters and approximated optimal controller information for different operating conditions can form a lookup table to better support process operation.

3.1. The LMHAC framework

The overall framework for the LMHAC is shown in Fig. 4. The management block sends the economic parameters to the real-time optimizer block, which periodically computes the optimal working point. MPC and ADP are fitted in the moving horizon framework. A lookup table is required to enable the operation of the LMHAC framework. The lookup table stores the model parameters and optimal/sub-optimal controller for each operating condition. If an operating condition's model parameters/optimal controller are unknown, then the corresponding entry in the lookup table is labeled unknown.

At the beginning of each control interval, the controller switching logic refers to the lookup table (as shown in Fig. 2). MPC is selected if the model parameters are known under the current operating condition. Otherwise, ADP is used to derive the optimal controller by learning from online production data. The unknown model parameters are then identified using the 'input-state' data collected by ADP and saved to the model parameter lookup table. The algorithm runs continuously to incrementally obtain optimal/approximated optimal controllers and knowledge about the process dynamics for different operating conditions as the 'process' moves through the comprehensive state space. The overall LMHAC algorithm is shown in Table 1.

Since the theory of MPC is mature, this paper only applies MPC in the case study, and the details of the MPC design are not included. This paper mainly focuses on how to fit the ADP in the moving horizon framework to derive an approximated optimal control when the model parameters are unknown and identify the unknown parameters afterward. To apply ADP in a finite control interval and identify the unknown model parameters, three main issues need to be addressed:

(i) First, to switch from an operating condition with known model parameters (where MPC is the controller) to another operating condition with unknown model parameters (where ADP is the controller), an initial admissible control is required to guarantee stability during the transition of the learning stage. In addition, ADP requires an initial admissible control, which iteratively converges to the optimal/nearly optimal control by learning from its interaction with the process. This framework's initial admissible control is designed based on the previously adjoining operating condition model and Lyapunov stability theory.

(ii) Second, to guarantee the convergence of the proposed

Table 1: The LMHAC Algorithm

Algorithm 1	LMHAC Algorithm
Step 1	Determine the structure of the process model and classify the process into different operating conditions.
Step 2	Identify the model parameters for the operating conditions with sufficient data samples and create a lookup table between operating conditions and model parameters.
Step 3	Determine the length of each optimization horizon.
Step 4	Choose between MPC and ADP based on whether the model parameters are known or unknown under the current operating condition.
Step 5	If the model parameters are known, use the conventional MPC control and go to **Step 4** when the current horizon is finished. Otherwise, go to **Step 6**.
Step 6	Design an initial admissible control for ADP (see Section 3.2).
Step 7	Learning stage: Iteratively improve the controller by learning from the 'act-response' information. When the stop criteria are met, finish the learning stage, and the result is the approximated optimal control (see Section 3.3).
Step 8	Control-identification stage: Apply the approximated optimal control to the process and identify the unknown model parameters. Go to **Step 4** when the current horizon is finished (see Section 3.4).

approach in finite time and the satisfaction of input constraints, the conventional infinite horizon setup of the ADP is converted to a finite horizon problem with constraints on the control input, i.e., Constrained Finite-horizon Adaptive Dynamic Programming (CFADP).

(iii) Third, for the identification, the HJB equation is used as a bridge to extract the value of the unknown model parameters from the 'input-state' information pair collected using ADP. The identified model parameters are stored in the model parameter lookup table, which is filled incrementally to provide a more comprehensive understanding of the process dynamics, as shown in Fig. 2, Fig. 3, Fig. 4.

3.2. Design of initial admissible control

3.2.1. Process dynamics reformulation

Assume that in the transition from one operating condition (operating condition I) to another (operating condition II), the variation of model parameters reflects the difference in process dynamics between the two operating conditions. If the process dynamics under operating condition I and operating condition II are:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{u}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{6}$$

And:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}^{'}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}}^{'})\boldsymbol{u}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0^{'} \tag{7}$$

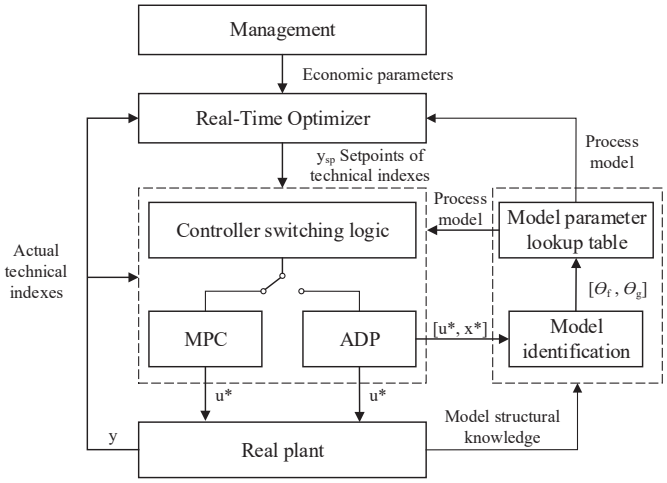

Figure 4: The framework for learning-based moving horizon autonomous control

respectively, where $[\boldsymbol{\theta}_{\mathrm{f}}, \boldsymbol{\theta}_{\mathrm{g}}]$ is known, and $[\boldsymbol{\theta}_{\mathrm{f}}^{'}, \boldsymbol{\theta}_{\mathrm{g}}^{'}]$ is unknown, then the process dynamics under operating condition II can be reformulated as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{u} + \boldsymbol{d}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0^{'} \tag{8}$$

where:

$$\boldsymbol{d} = [\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}^{'}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}}^{'})\boldsymbol{u}] - [\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{u}] \tag{9}$$

accounts for the dynamic difference between the two operating conditions. $\boldsymbol{d}$ is bounded by a constant $\boldsymbol{D}$ which is unknown. $\boldsymbol{x}_0$ and $\boldsymbol{x}_0^{'}$ are initial states under operating condition I and operating condition II.

3.2.2. Admissible control design based on a Lyapunov function

An admissible control is required to guarantee stability during the learning stage transition. According to the definition of admissible control [51][52] and Lyapunov stability theory [53], if one can design a control and find a continuously differentiable positive definite function of the system state on a domain of attraction so that the derivative of the function is negative semi-definite. The function is a Lyapunov function, the control is admissible, and the closed-loop system is stable. The design goal here is to obtain an initial admissible control law that guarantees the stability of the closed-loop system when switching from a known to an unknown operating condition. This control law will serve as the initial policy for the subsequent ADP iteration, providing a safe and feasible starting point for the learning-based controller.

Therefore, the key is the design of the Lyapunov function and the corresponding control law. To start with, denote:

$$\boldsymbol{u} = \boldsymbol{H}(\boldsymbol{v}), \tag{10}$$

where $\boldsymbol{v}$ is an unconstrained auxiliary control variable introduced to facilitate the controller design in an unconstrained space. Designing the control law in **v**-space allows the use

of standard Lyapunov and backstepping techniques without directly handling actuator saturation. The function $H(\cdot)$ maps $\mathbf{v}$ to the actual physical control input $\mathbf{u}$, ensuring that $u_{\min} \leq u \leq u_{\max}$ is always satisfied and the constructed control law inherently respects the input constraints. In this paper, $H(\mathbf{v})$ is given by:

$$H(v) = u_{\min} + \frac{u_{\max} - u_{\min}}{2} [\tanh(v) + 1], \tag{11}$$

where $u_{\min}$ and $u_{\max}$ denote the physical lower and upper limits of the control input, respectively. Furthermore, we define an intermediate variable:

$$\boldsymbol{r} = \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{H}(\boldsymbol{v}), \tag{12}$$

where $\mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})$ denotes the system's input gain. Here, $\mathbf{r}$ acts as an intermediate or virtual input that combines the actual physical input $\mathbf{u}$ with its gain into a single term. For the purpose of the recursive design, we also introduce the auxiliary variable $\boldsymbol{w}$ as:

$$\dot{\boldsymbol{r}} = \boldsymbol{w}, \tag{13}$$

which enables the systematic design of the control law in a backstepping-like manner. And define the following:

$$z_1 = \boldsymbol{x} - \boldsymbol{x}_r \tag{14}$$

$$z_2 = \boldsymbol{r} - \boldsymbol{p} \tag{15}$$

Where $\boldsymbol{x}_r$ is the set-point of the system state with constant value within a control interval, z_1 is the tracking error, and $\boldsymbol{p}$ is an intermediate variable to facilitate control design.

Consider the following Lyapunov function:

$$V_1 = \frac{1}{2} z_1^2 \tag{16}$$

whose derivative is:

$$\dot{V}_1 = z_1^{\mathrm{T}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + z_2 + \boldsymbol{p} + \boldsymbol{d}] \tag{17}$$

Since $z_1^{\mathrm{T}}\boldsymbol{d} \leqslant |z_1^{\mathrm{T}}|\boldsymbol{D}$:

$$\dot{V}_1 \leqslant z_1^{\mathrm{T}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + z_2 + \boldsymbol{p}] + |z_1^{\mathrm{T}}|\boldsymbol{D} \tag{18}$$

If:

$$\boldsymbol{p} = -c_1 z_1 - \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) - \mathrm{sgn}(z_1)\hat{\boldsymbol{D}} \tag{19}$$

where $\hat{\boldsymbol{D}}$ is a design variable to handle the unknown $\boldsymbol{D}$, and c_1 is a positive design parameter, then:

$$\begin{aligned} \dot{V}_1 &\leqslant -c_1 z_1^2 + z_1^{\mathrm{T}} z_2 - |z_1^{\mathrm{T}}|\hat{\boldsymbol{D}} + |z_1^{\mathrm{T}}|\boldsymbol{D} \\ &= -c_1 z_1^2 + z_1^{\mathrm{T}} z_2 + |z_1^{\mathrm{T}}|\widetilde{\boldsymbol{D}} \end{aligned} \tag{20}$$

where $\widetilde{\boldsymbol{D}} = \boldsymbol{D} - \hat{\boldsymbol{D}}$. In addition, consider the following Lyapunov function:

$$V_2 = \frac{1}{2} z_2^2 \tag{21}$$

whose derivative is:

$$\begin{aligned} \dot{V}_2 &= z_2^{\mathrm{T}}\{\boldsymbol{w} - [-c_1\dot{z}_1 - \dot{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) - \mathrm{sgn}(z_1)\dot{\hat{\boldsymbol{D}}}]\} \\ &= z_2^{\mathrm{T}}\{\boldsymbol{w} + c_1[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \boldsymbol{r}] + \dot{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathrm{sgn}(z_1)\dot{\hat{\boldsymbol{D}}} + \mathrm{c}_1\boldsymbol{d}\} \\ &\leqslant z_2^{\mathrm{T}}\{\boldsymbol{w} + c_1[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \boldsymbol{r}] + \dot{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) \\ &\quad + \mathrm{sgn}(z_1)\dot{\hat{\boldsymbol{D}}} + \mathrm{c}_1\mathrm{sgn}(z_2)\boldsymbol{D}\} \end{aligned} \tag{22}$$

If:

$$\begin{aligned} \boldsymbol{w} \quad = \quad & -c_2 z_2 - z_1 - c_1[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \boldsymbol{r}] - \dot{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) \\ & -\mathrm{sgn}(z_1)\dot{\hat{\boldsymbol{D}}} - \mathrm{c}_1\mathrm{sgn}(z_2)\hat{\boldsymbol{D}} \end{aligned} \tag{23}$$

Where c_2 is a positive design parameter, then:

$$\dot{V}_2 \leqslant -c_2 z_2^2 - z_2^{\mathrm{T}} z_1 + c_1|z_2^{\mathrm{T}}|\widetilde{\boldsymbol{D}} \tag{24}$$

Consider the following Lyapunov function:

$$V_3 = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2 + \frac{1}{2\eta}\widetilde{\boldsymbol{D}}^2 \tag{25}$$

If:

$$\dot{\hat{\boldsymbol{D}}} = \eta(-|z_1| - c_1|z_2|) \tag{26}$$

where η is a positive design parameter, then V_3 has a nonpositive derivative:

$$\begin{aligned} \dot{V}_3 &\leqslant -c_1 z_1^2 - c_2 z_2^2 + |z_1^{\mathrm{T}}|\widetilde{\boldsymbol{D}} + c_1|z_2|\widetilde{\boldsymbol{D}} - \frac{1}{\eta}\widetilde{\boldsymbol{D}}^{\mathrm{T}}\dot{\hat{\boldsymbol{D}}} \\ &= -c_1 z_1^2 - c_2 z_2^2 \end{aligned} \tag{27}$$

(26) is a user-defined equation and is part of the overall controller to guarantee that the system composed of (6), (10)-(13) is stable.

As $\dot{V}_3$ is negative semi-definite, the system is stable, and the state $\boldsymbol{x}$ stays in a neighborhood of its target value $\boldsymbol{x}_r$, which indicates $\boldsymbol{u}$ is an admissible control.

The admissible control law derived above is theoretically guaranteed to stabilize the system under parameter uncertainties. It provides the initial policy u_0 for the ADP learning stage in our LMHAC framework. In the following ADP learning stage, the initial admissible control policy designed in the previous subsection is used as the starting point for policy iteration, ensuring that each learning episode begins with a stable and feasible controller.

3.3. Learning stage

3.3.1. Preliminaries of ADP

Consider system (1), with the following cost function to be minimized:

$$\begin{aligned} \min_{\boldsymbol{u}} \quad & J(\boldsymbol{x}(t_0), t_0) = \int_{t_0}^{t_f} [Q(\boldsymbol{x}) + W(\boldsymbol{u})]\mathrm{d}t \\ s.t. \quad & \begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{u} \\ \boldsymbol{u}_{\mathrm{low}} \leqslant \boldsymbol{u} \leqslant \boldsymbol{u}_{\mathrm{up}} \\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \end{cases} \end{aligned} \tag{28}$$

where

$$Q(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}^*)^{\mathrm{T}}\mathbf{Q}(\boldsymbol{x} - \boldsymbol{x}^*) \tag{29}$$

$$W(\boldsymbol{u}) = 2\sum_{i=1}^{m}\int_0^{u_i} h_i^{-1}(v_i)R_i dv_i \tag{30}$$

$Q(\boldsymbol{x})$ and $W(\boldsymbol{u})$ are positive definite functions, where $\mathbf{Q} \in \mathbb{R}^{m\times m}$ and $\mathbf{R} = \mathrm{diag}(R_1, \ldots, R_m) \in \mathbb{R}^{m\times m}$ are diagonal matrices with positive diagonal elements. Each $h_i(\cdot)$ is a strictly monotonic, continuous, bounded, and odd C^p ($p \geqslant 1$) function ($i = 1, 2, \cdots, m$), which maps the unbounded domain $\mathbb{R}$ to the feasible range of the i-th control input, thus ensuring the input constraint is satisfied. The inverse function $h_i^{-1}(\cdot)$ exists and is unique due to strict monotonicity. The factor 2 is included to maintain consistency with the quadratic form in the linear case, i.e., $W(\boldsymbol{u}) = \boldsymbol{u}^{\mathrm{T}}\mathbf{R}\boldsymbol{u}$, and to generalize the penalty for the nonlinear input mapping.

The optimal solution of problem (28) is

$$\boldsymbol{u}^*(\boldsymbol{x}) = -\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\mathbf{g}^{\mathrm{T}}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\frac{\partial J^*}{\partial \boldsymbol{x}}) \tag{31}$$

where $\boldsymbol{H}(\cdot)$ is a set of functions that maps the input constraints of the process. The obtaining of $\boldsymbol{u}^*(\boldsymbol{x})$ relies on the iterative solution of the following Hamilton-Jacobi-Bellman (HJB) equation:

$$\begin{aligned}&\frac{\partial J}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) - \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\mathbf{g}^{\mathrm{T}}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\frac{\partial J}{\partial \boldsymbol{x}})]\\ &+Q(\boldsymbol{x}) + 2\int_0^{-\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\mathbf{g}^{\mathrm{T}}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\frac{\partial J}{\partial \boldsymbol{x}})} (\boldsymbol{H}^{-1}(\boldsymbol{v}))^{\mathrm{T}}\mathbf{R}d\boldsymbol{v} = 0\end{aligned} \tag{32}$$

where $J(0) = 0$. The existence and uniqueness of the solution of (32) has been proved in [54]. The solution $\mathbf{u}^*$ and J^* of (32) can be approximated using a Policy Iteration (PI) approach [55]:

1. Choose an initial stabilizing admissible controller $u_0(\boldsymbol{x})$.
2. For $i \geqslant 0$, solve following Lyapunov equation for J_i

$$\begin{aligned}&\frac{\partial J_i}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})u_i(\boldsymbol{x})] + Q(\boldsymbol{x})\\ &+2\int_0^{u_i(\boldsymbol{x})} (\boldsymbol{H}^{-1}(\boldsymbol{v}))^{\mathrm{T}}\mathbf{R}d\boldsymbol{v} = 0\end{aligned} \tag{33}$$

3. Obtain an updated control law $u_{i+1}(\boldsymbol{x})$

$$\boldsymbol{u}_{i+1}^*(\boldsymbol{x}) = -\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\mathbf{g}^{\mathrm{T}}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\frac{\partial J_i}{\partial \boldsymbol{x}}) \tag{34}$$

Theorem 1. *For system (1) and problem (28), starting from an initial admissible control $\boldsymbol{u}_0$, the sequences of cost and approximated optimal control law, i.e., $\{J_i\}_{i=0}^{\infty}$ and $\{\boldsymbol{u}_{i+1}\}_{i=0}^{\infty}$ can be generated via the PI approach defined by (33) and (34), and*

1. $0 \leqslant J_{i+1} \leqslant J_i$,
2. *$\boldsymbol{u}_i$ is bounded and admissible,*
3. *If J^* and $\boldsymbol{u}^*$ exist, then $J_i \to J^*$, $\boldsymbol{u}_i \to \boldsymbol{u}^*$.*

Proof. The proof of this theorem follows the same lines of reasoning as the proof of **Lemma 1** and **Theorem 1** in [55], and is omitted here for brevity.

3.3.2. A finite horizon ADP with input constraints

For the finite horizon optimal control of nonlinear systems, the performance index is formulated as follows:

$$J(\boldsymbol{x}(t_0), t_0) = F_{\mathrm{terminal}}(\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f}\gamma(\boldsymbol{x}, \boldsymbol{u})\mathrm{d}t \tag{35}$$

According to [56], an infinitesimal equivalent to (35) is:

$$-\frac{\partial J}{\partial t} = \gamma(\boldsymbol{x}, \boldsymbol{u}) + \frac{\partial J}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{u}] \tag{36}$$

with boundary condition:

$$J(\boldsymbol{x}(t_f), t_f) = F_{\mathrm{terminal}}(\boldsymbol{x}(t_f)) \tag{37}$$

The optimal control for this problem is:

$$\boldsymbol{u}^* = -\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})^{\mathrm{T}}\frac{\partial J^*}{\partial \boldsymbol{x}}) \tag{38}$$

With J^*, the solution of the following time-varying HJB equation:

$$\begin{aligned}&\frac{\partial J}{\partial t} + \boldsymbol{x}^{\mathrm{T}}\mathbf{Q}\boldsymbol{x} + 2\int_{\mathbf{0}}^{-\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}_{\mathrm{g}})^{\mathrm{T}}\frac{\partial J}{\partial \boldsymbol{x}})} (\boldsymbol{H}^{-1}(\boldsymbol{v}))^{\mathrm{T}}\mathbf{R}\mathrm{d}\boldsymbol{v}\\ &+\frac{\partial J}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) - \mathbf{g}^{\mathrm{T}}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})^{\mathrm{T}}\frac{\partial J}{\partial \boldsymbol{x}})] = 0\end{aligned} \tag{39}$$

(39) cannot be solved analytically. Therefore, an iterative solution approach is applied. To start the iteration, the control input is decomposed as a combination of an iterative control and excitation noise:

$$\boldsymbol{u} = \boldsymbol{u}_i + \boldsymbol{e}_i, \tag{40}$$

where $\boldsymbol{u}_i$ is the control signal at iteration i, which will be updated in subsequent iterations, and $\boldsymbol{e}_i$ is a bounded excitation noise vector introduced at iteration i to ensure persistent excitation for policy improvement and parameter identification.

The original system (8) can be re-written as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{u}_i + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{e}_i, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{41}$$

In this formulation, the dynamic difference term $\boldsymbol{d}$ in Eq. (8) is compensated or identified through the excitation provided by $\boldsymbol{e}_i$ during the iterative learning process. Therefore, Eq. (41) serves as an equivalent reformulation of Eq. (8), making it more suitable for the ADP-based iterative optimization and identification procedure.

Assumption 1: The closed loop system (41) is ISS (Input-to-State Stable) [57] when the exploration noise $\boldsymbol{e}_i$, is considered as the input. This assumption holds in industrial chemical reactors, as these systems are designed for stability under bounded disturbances. The admissible initial control in our method further ensures the ISS condition is met in practice.

By applying (38) and (36), for each $\boldsymbol{u}_i$, the difference of the cost function along the trajectory of (41) on the time interval $[t_k, t_{k+1}]$ is

$$\begin{aligned}&J_i(\boldsymbol{x}(t_{k+1}), t_{k+1}) - J_i(\boldsymbol{x}(t_k), t_k)\\ &= \int_{t_k}^{t_{k+1}}[-\gamma(\boldsymbol{x}, \boldsymbol{u}_i) - \mu(\boldsymbol{u}_{i+1})\boldsymbol{e}_i]\mathrm{d}t\end{aligned} \tag{42}$$

where $\mu(\boldsymbol{u}_{i+1}) = 2(\boldsymbol{H}^{-1}(\boldsymbol{u}_{i+1}))^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}$. It is observed from (42) that:

(i) there is no element of the system model (1) in (42).

(ii) (42) could serve as a scheme for the iteration of $\boldsymbol{u}_i$ and J_i.

Let $\Omega \subset \mathbb{R}^n \times [t_0, t_f]$ be a compact set in the joint state-time space that contains all possible $(\boldsymbol{x}, t)$ pairs visited by the system during learning and control. All basis function approximations and parameter estimations are performed on this set. Approximating the control policy and cost function on a compact set Ω using 'linear-in-the-weight' networks gives:

$$\hat{J}_i = \sum_{j=1}^{N_1} \hat{c}_{i,j}\phi_j(\boldsymbol{x}, t) = \hat{\boldsymbol{c}}_i^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}, t) \tag{43}$$

And:

$$\hat{\boldsymbol{u}}_i(\boldsymbol{x}) = -\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\hat{\boldsymbol{k}}_i\boldsymbol{\psi}(\boldsymbol{x}, t)) \tag{44}$$

Where:

$$\boldsymbol{\phi}(\boldsymbol{x}, t) = [\phi_1(\boldsymbol{x}, t) \quad \phi_2(\boldsymbol{x}, t) \quad \dots \quad \phi_{N_1}(\boldsymbol{x}, t)]^{\mathrm{T}} \tag{45}$$

$$\boldsymbol{\psi}(\boldsymbol{x}, t) = [\psi_1(\boldsymbol{x}, t) \quad \psi_2(\boldsymbol{x}, t) \quad \dots \quad \psi_{N_2}(\boldsymbol{x}, t)]^{\mathrm{T}} \tag{46}$$

$$\hat{\boldsymbol{c}}_i = [\hat{c}_{i,1} \quad \hat{c}_{i,2} \quad \dots \quad \hat{c}_{i,N_1}]^{\mathrm{T}} \tag{47}$$

$$\hat{\boldsymbol{k}}_i = \begin{bmatrix} \hat{k}_{i_{1,1}} & \hat{k}_{i_{1,2}} & \dots & \hat{k}_{i_{1,N_2}} \\ \hat{k}_{i_{2,1}} & \hat{k}_{i_{2,2}} & \dots & \hat{k}_{i_{2,N_2}} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{k}_{i_{m,1}} & \hat{k}_{i_{m,2}} & \dots & \hat{k}_{i_{m,N_2}} \end{bmatrix} \tag{48}$$

N_1 and N_2 are two sufficiently large positive integers. $\{\phi_j(\boldsymbol{x}, t)\}_{j=1}^{\infty}$ and $\{\psi_j(\boldsymbol{x}, t)\}_{j=1}^{\infty}$ are two infinite sequences of time-dependent smooth basis functions. In practical implementation, only a finite number of basis functions (specified by N_1 and N_2) are used to approximate the value function and policy. According to standard results in approximation theory, for any compact set Ω and any desired approximation accuracy, there always exist finite N_1 and N_2 such that the truncated expansions using the N_1 and N_2 basis functions can approximate the target functions to within that accuracy. $\{\phi_j(\mathbf{0}, t)\} = 0$ and $\{\psi_j(\mathbf{0}, t)\} = 0$ for all $j = 1, 2, ...$ In addition, for the same t, the function value increases with $\boldsymbol{x}$. Moreover, as there are two parts in the cost function, i.e., the integrated intermediate cost and the terminal error cost, there exist two different types of elements in the basis function:

- $\boldsymbol{\phi}^{(\mathrm{A})}(\boldsymbol{x}, t)$: For the same $\boldsymbol{x}$, the function value increases with the evolution of t, which is used to approximate the terminal cost.
- $\boldsymbol{\phi}^{(\mathrm{B})}(\boldsymbol{x}, t)$: For the same $\mathbf{x}$, the function value does not increase with the evolution of t, which is used to approximate the intermediate cost.

In this work, we adopt the 'linear-in-the-weight' approximation structure for both value function and policy. Compared with NN-based actor/critic methods, this approach provides more transparent parameter estimation, computational efficiency, and theoretical convergence guarantees, which are especially important for safety-critical industrial process control. While NN-based actor-critic architectures are more flexible, they may suffer from overfitting and lack of interpretability for real-time applications.

If (43) and (44) are substituted into (42), it is possible to approximate (42) as:

$$\begin{aligned} &\hat{\boldsymbol{c}}_i^{\mathrm{T}}[\boldsymbol{\phi}(\boldsymbol{x}(t_{k+1}), t_{k+1}) - \boldsymbol{\phi}(\boldsymbol{x}(t_k), t_k)] \\ &- \int_{t_k}^{t_{k+1}} [(\hat{\boldsymbol{k}}_{i+1}\boldsymbol{\psi}(\boldsymbol{x}, t))^{\mathrm{T}}(\boldsymbol{u} - \hat{\boldsymbol{u}}_i)]\mathrm{d}t \\ = &\int_{t_k}^{t_{k+1}} [-\boldsymbol{x}^{\mathrm{T}}\mathbf{Q}\boldsymbol{x} - 2\int_{\mathbf{0}}^{\hat{\boldsymbol{u}}_i} (\boldsymbol{H}^{-1}(\boldsymbol{u}))^{\mathrm{T}}\mathbf{R}d\boldsymbol{u}]\mathrm{d}t + \epsilon_i \end{aligned} \tag{49}$$

where ϵ_i is the approximation error of (49), originating from basis function truncation and numerical integration.[52] Since all approximations are performed on the compact set $\Omega \subset \mathbb{R}^n \times [t_0, t_f]$, there exists a constant $\bar{\epsilon} > 0$ such that

$$\|\epsilon_i\| \leq \bar{\epsilon}, \quad \forall(\mathbf{x}, t) \in \Omega, \ \forall i \geq 0. \tag{50}$$

Under this boundedness assumption and Assumption 1, the iteration in (53) ensures that the value function and policy converge to neighborhoods of their optimal counterparts, with neighborhood sizes proportional to $\bar{\epsilon}$. Equation (49) describes the batch update of the value function and policy parameters using temporal-difference information over the segment $[t_k, t_{k+1}]$. Here, $\hat{\boldsymbol{c}}_i$ is the parameter vector for the value function approximation with basis vector $\boldsymbol{\phi}(\boldsymbol{x}, t)$, and $\hat{\boldsymbol{k}}_{i+1}$ is the parameter matrix for the policy approximation with basis $\boldsymbol{\psi}(\boldsymbol{x}, t)$. The term $\otimes$ denotes the Kronecker product, and $\mathrm{vec}(\cdot)$ denotes matrix vectorization. This formulation provides a data-driven approach for updating the value and policy parameters simultaneously.

Consider a sufficient long time sequence $\{t_k\}_{k=0}^{N_3}$ with $N_3 \geqslant N_1 + mN_2$, then

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}(t_1), t_1)^{\mathrm{T}} - \boldsymbol{\phi}(\boldsymbol{x}(t_0), t_0)^{\mathrm{T}} \\ \boldsymbol{\phi}(\boldsymbol{x}(t_2), t_2)^{\mathrm{T}} - \boldsymbol{\phi}(\boldsymbol{x}(t_1), t_1)^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}(t_{N_3}), t_{N_3})^{\mathrm{T}} - \boldsymbol{\phi}(\boldsymbol{x}(t_{N_3-1}), t_{N_3-1})^{\mathrm{T}} \end{bmatrix} \hat{\boldsymbol{c}}_i \\ &- \begin{bmatrix} \int_{t_0}^{t_1} (\boldsymbol{u} - \hat{\boldsymbol{u}}_i)^{\mathrm{T}} \otimes \boldsymbol{\psi}(\boldsymbol{x}, t)^{\mathrm{T}}\mathrm{d}t \\ \int_{t_1}^{t_2} (\boldsymbol{u} - \hat{\boldsymbol{u}}_i)^{\mathrm{T}} \otimes \boldsymbol{\psi}(\boldsymbol{x}, t)^{\mathrm{T}}\mathrm{d}t \\ \vdots \\ \int_{t_{N_3-1}}^{t_{N_3}} (\boldsymbol{u} - \hat{\boldsymbol{u}}_i)_i^{\mathrm{T}} \otimes \boldsymbol{\psi}(\boldsymbol{x}, t)^{\mathrm{T}}\mathrm{d}t \end{bmatrix} \mathrm{vec}(\hat{\boldsymbol{k}}_{i+1}) = \\ &\begin{bmatrix} \int_{t_0}^{t_1} [-Q(\boldsymbol{x}) - W(\boldsymbol{u})]\mathrm{d}t \\ \int_{t_1}^{t_2} [-Q(\boldsymbol{x})) - W(\boldsymbol{u})]\mathrm{d}t \\ \vdots \\ \int_{t_{N_3-1}}^{t_{N_3}} [-Q(\boldsymbol{x}) - W(\boldsymbol{u})]\mathrm{d}t \end{bmatrix} + \boldsymbol{\epsilon} \end{aligned} \tag{51}$$

This equation stacks the temporal-difference equations over all sampled trajectory intervals into a linear system. The left matrix multiplies the value function coefficients, the middle block corresponds to the policy coefficients, and the right side stacks the observed costs and residuals $\boldsymbol{\epsilon}$. This batch equation

is used to jointly solve the value function and policy parameters from data via least-squares.

In addition, the basic functions for approximating the terminal error should satisfy the following:

$$\begin{bmatrix} \boldsymbol{\phi}^{(\mathrm{A})}(\boldsymbol{x}_1, t_f)^{\mathrm{T}} \\ \boldsymbol{\phi}^{(\mathrm{A})}(\boldsymbol{x}_2, t_f)^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\phi}^{(\mathrm{A})}(\boldsymbol{x}_{N_4}, t_f)^{\mathrm{T}} \end{bmatrix} \hat{\boldsymbol{c}}_i^{(\mathrm{A})} = \begin{bmatrix} F_{\mathrm{terminal}}(\boldsymbol{x}_1) \\ F_{\mathrm{terminal}}(\boldsymbol{x}_2) \\ \vdots \\ F_{\mathrm{terminal}}(\boldsymbol{x}_{N_4}) \end{bmatrix} \tag{52}$$

where $\hat{\boldsymbol{c}}_i^{(\mathrm{A})}$ denotes the elements of $\hat{\boldsymbol{c}}_i$ corresponding to basis functions $\boldsymbol{\phi}^{(\mathrm{A})}(\boldsymbol{x}, t)$, $[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{N_4}]$ are N_4 different terminal states, $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \cdots, \epsilon_{N_3}]^{\mathrm{T}}$. If N_3 and N_4 are sufficiently large and $\hat{\boldsymbol{u}}_i(\boldsymbol{x})$ are given, then (51) and (52) could serve as an equation array to obtain the unknown parameters in $\hat{\boldsymbol{c}}_i$ and $\hat{\boldsymbol{k}}_{i+1}$. This terminal equation ensures that the value function approximation matches the true terminal cost for all sampled terminal states, i.e., it enforces consistency at the terminal time. Here, $\boldsymbol{\phi}^{(\mathrm{A})}$ is the set of basis functions corresponding to terminal states, and $\hat{\boldsymbol{c}}_i^{(\mathrm{A})}$ are the corresponding coefficients.

In the solution process, $\hat{\boldsymbol{c}}_i^{(\mathrm{A})}$ is first determined using (52). Then, the other elements in $\hat{\boldsymbol{c}}_i$, denoted as $\hat{\boldsymbol{c}}_i^{(\mathrm{B})}$, can be obtained from (51) by keeping $\hat{\boldsymbol{c}}_i^{(\mathrm{A})}$ constant:

$$\begin{bmatrix} \hat{\boldsymbol{c}}_i \\ \mathrm{vec}(\hat{\boldsymbol{k}}_{i+1}) \end{bmatrix} = (\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\theta})^{-1}\boldsymbol{\theta}^{\mathrm{T}}\Pi + \boldsymbol{\epsilon}_{\mathrm{PI}} \tag{53}$$

where $\boldsymbol{\theta} = [\Xi_{\boldsymbol{\phi}} \quad -\mathbf{I}_{\boldsymbol{\psi}u} + \mathbf{I}_{\boldsymbol{\psi}\hat{u}_i}]$, $\Pi = [-\mathbf{M}_{\boldsymbol{x}} - \mathbf{M}_{\boldsymbol{u}}]$, $\boldsymbol{\epsilon}_{\mathrm{PI}}$ is the approximation error of (53), and:

$$\Xi_{\boldsymbol{\phi}} = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}(t_1), t_1)^{\mathrm{T}} - \boldsymbol{\phi}(\boldsymbol{x}(t_0), t_0)^{\mathrm{T}} \\ \boldsymbol{\phi}(\boldsymbol{x}(t_2), t_2)^{\mathrm{T}} - \boldsymbol{\phi}(\boldsymbol{x}(t_1), t_1)^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}(t_{N_3}), t_{N_3})^{\mathrm{T}} - \boldsymbol{\phi}(\boldsymbol{x}(t_{N_3-1}), t_{N_3-1}))^{\mathrm{T}} \end{bmatrix} \tag{54}$$

$$\mathbf{I}_{\boldsymbol{\psi}u} = \begin{bmatrix} \int_{t_0}^{t_1} \boldsymbol{u}^{\mathrm{T}} \otimes \boldsymbol{\psi}(\boldsymbol{x}, t)^{\mathrm{T}} \mathrm{d}t \\ \int_{t_1}^{t_2} \boldsymbol{u}^{\mathrm{T}} \otimes \boldsymbol{\psi}(\boldsymbol{x}, t)^{\mathrm{T}} \mathrm{d}t \\ \vdots \\ \int_{t_{N_3-1}}^{t_{N_3}} \boldsymbol{u}^{\mathrm{T}} \otimes \boldsymbol{\psi}(\boldsymbol{x}, t)^{\mathrm{T}} \mathrm{d}t \end{bmatrix} \tag{55}$$

$$\mathbf{I}_{\boldsymbol{\psi}\hat{u}_i} = \begin{bmatrix} \int_{t_0}^{t_1} -\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\hat{\boldsymbol{k}}_i\boldsymbol{\psi}(\boldsymbol{x}, t))^{\mathrm{T}} \otimes \boldsymbol{\psi}(\boldsymbol{x}, t)^{\mathrm{T}} \mathrm{d}t \\ \int_{t_1}^{t_2} -\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\hat{\boldsymbol{k}}_i\boldsymbol{\psi}(\boldsymbol{x}, t))^{\mathrm{T}} \otimes \boldsymbol{\psi}(\boldsymbol{x}, t)^{\mathrm{T}} \mathrm{d}t \\ \vdots \\ \int_{t_{N_3-1}}^{t_{N_3}} -\boldsymbol{H}(\frac{1}{2}\mathbf{R}^{-1}\hat{\boldsymbol{k}}_i\boldsymbol{\psi}(\boldsymbol{x}, t))^{\mathrm{T}} \otimes \boldsymbol{\psi}(\boldsymbol{x}, t)^{\mathrm{T}} \mathrm{d}t \end{bmatrix} \tag{56}$$

$$\mathbf{M}_{\boldsymbol{x}} = \begin{bmatrix} \int_{t_0}^{t_1} \boldsymbol{x}^{\mathrm{T}}\mathrm{Q}\boldsymbol{x} \mathrm{d}t \\ \int_{t_1}^{t_2} \boldsymbol{x}^{\mathrm{T}}\mathrm{Q}\boldsymbol{x} \mathrm{d}t \\ \vdots \\ \int_{t_{N_3-1}}^{t_{N_3}} \boldsymbol{x}^{\mathrm{T}}\mathrm{Q}\boldsymbol{x} \mathrm{d}t \end{bmatrix} \tag{57}$$

$$\mathbf{M}_{\boldsymbol{u}} = \begin{bmatrix} \int_{t_0}^{t_1} 2\int_{\mathbf{0}}^{\hat{\boldsymbol{u}}_i} (\boldsymbol{H}^{-1}(\boldsymbol{v}))^{\mathrm{T}}\mathbf{R}d\boldsymbol{v}\mathrm{d}t \\ \int_{t_1}^{t_2} 2\int_{\mathbf{0}}^{\hat{\boldsymbol{u}}_i} (\boldsymbol{H}^{-1}(\boldsymbol{v}))^{\mathrm{T}}\mathbf{R}d\boldsymbol{v}\mathrm{d}t \\ \vdots \\ \int_{t_{N_3-1}}^{t_{N_3}} 2\int_{\mathbf{0}}^{\hat{\boldsymbol{u}}_i} (\boldsymbol{H}^{-1}(\boldsymbol{u}))^{\mathrm{T}}\mathbf{R}d\boldsymbol{u}\mathrm{d}t \end{bmatrix} \tag{58}$$

Equations (53)-(58) present the batch least-squares solution for value function and policy parameters. Here, $\boldsymbol{\theta}$ is the regression matrix built from stacked differences in basis functions and controls, and Π is the stacked vector of state and control costs. The explicit forms $\Xi_{\boldsymbol{\phi}}$, $\mathbf{I}_{\boldsymbol{\psi}u}$, $\mathbf{I}_{\boldsymbol{\psi}\hat{u}_i}$, $\mathbf{M}_{\boldsymbol{x}}$, and $\mathbf{M}_{\boldsymbol{u}}$ aggregate the batch trajectory data. This framework supports robust and simultaneous identification of both value and policy parameters.

Assumption 2: There exist $L_0 > 0$ and $\delta > 0$, such that for all $L > L_0$

$$\frac{1}{L}\sum_{k=1}^{L} \theta_k^{\mathrm{T}}\theta_k \geqslant \mathrm{I}_{\mathrm{N}_1+\mathrm{N}_2} \tag{59}$$

where θ_k is the kth row of $\boldsymbol{\theta}$. In real reactor operation, sufficient data excitation is provided by natural process variations or safe, bounded exploration signals, which are standard and effective for guaranteeing convergence.

By using (53), starting from an initial admissible control $\hat{\boldsymbol{u}}_0$, two sequences, $\{\hat{J}_i\}_{i=0}^{\infty}$ and $\{\hat{\boldsymbol{u}}_{i+1}\}_{i=0}^{\infty}$, are generated. Under **Assumption 1** and **Assumption 2**, when the convergence criterion is met (i.e., $|\hat{\boldsymbol{c}}_i - \hat{\boldsymbol{c}}_{i-1}|^2 \leqslant \boldsymbol{\epsilon}_{\mathrm{converge}}$, where $\boldsymbol{\epsilon}_{\mathrm{converge}} > 0$ is a sufficiently small predefined threshold), the iteration stops, and the resulting $[\hat{\boldsymbol{c}}^*, \hat{\boldsymbol{k}}^*]$ is an approximated solution of (53). In addition, according to **Theorem 1**, the control policy $\hat{\boldsymbol{u}}^*$ and $\hat{J}^*$ converge to the optimal controller and optimal cost, respectively.

3.4. Control-identification stage

According to (36), the optimal control and cost function satisfy:

$$\frac{\partial J^*}{\partial t} + \gamma(\boldsymbol{x}, \boldsymbol{u}^*) + \frac{\partial J^*}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{g}})\boldsymbol{u}^*] = 0 \tag{60}$$

This equation is the continuous-time HJB equation, representing the necessary condition for optimality in terms of the value function J^* and the optimal control $\boldsymbol{u}^*$. As shown in Fig. 5, after the learning stage $[0, t_{\mathrm{learn}}]$, if the approximated optimal control is obtained and applied, then (60) is approximated as:

$$\frac{\partial \hat{J}^*}{\partial t} + \gamma(\boldsymbol{x}, \hat{\boldsymbol{u}}^*) + \frac{\partial \hat{J}^*}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}, \hat{\boldsymbol{\theta}}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \hat{\boldsymbol{\theta}}_{\mathrm{g}})\hat{\boldsymbol{u}}^*] = \xi \tag{61}$$

where ξ is the approximation error of (61), $\hat{\boldsymbol{\theta}}_{\mathrm{f}}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{g}}$ are estimations of $\boldsymbol{\theta}_{\mathrm{f}}$ and $\boldsymbol{\theta}_{\mathrm{g}}$. Substituting (43) into (61) gives:

$$\begin{aligned} &\frac{\partial(\hat{\boldsymbol{c}}^{*^{\mathrm{T}}}\boldsymbol{\phi}(\boldsymbol{x}, t))}{\partial t} + \gamma(\boldsymbol{x}, \hat{\boldsymbol{u}}^*) \\ &+ \frac{\partial(\hat{\boldsymbol{c}}^{*^{\mathrm{T}}}\boldsymbol{\phi}(\boldsymbol{x}, t))}{\partial \mathbf{x}}[\boldsymbol{f}(\boldsymbol{x}, \hat{\boldsymbol{\theta}}_{\mathrm{f}}) + \mathbf{g}(\boldsymbol{x}, \hat{\boldsymbol{\theta}}_{\mathrm{g}})\hat{\boldsymbol{u}}^*] = \xi \end{aligned} \tag{62}$$

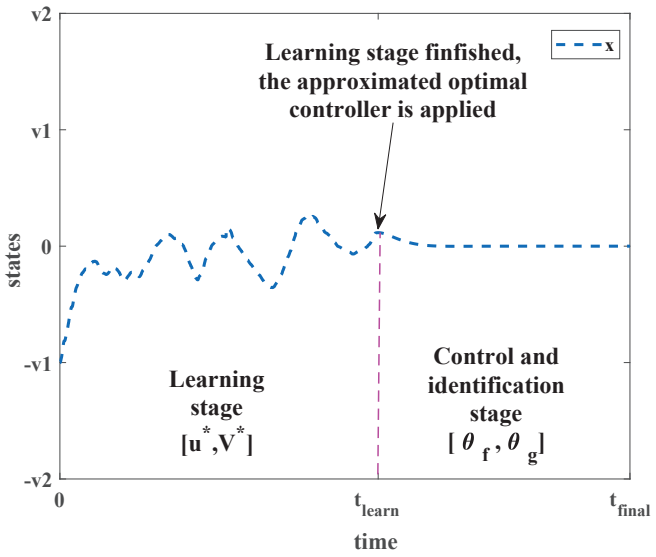

Figure 5: Different stages in the learning-based moving horizon autonomous control process

This equation explicitly expresses the residual using the value function approximation $\hat{J}^*(\boldsymbol{x},t) = \hat{\boldsymbol{c}}^{*^{\mathrm{T}}}\boldsymbol{\phi}(\boldsymbol{x},t)$, where $\hat{\boldsymbol{c}}^*$ are the learned parameters and $\boldsymbol{\phi}(\boldsymbol{x},t)$ is the basis function vector.

Therefore, if the model structure has not changed, the unknown model parameter set $\boldsymbol{\theta}_{\mathrm{f}}$ and $\boldsymbol{\theta}_{\mathrm{g}}$ in the system model (1) can be estimated using (62). If the information collected from the system evolution trajectory during the control-identification stage $[t_{\mathrm{learn}}, t_{\mathrm{final}}]$ is not sufficient to identify $\boldsymbol{\theta}_{\mathrm{f}}$ and $\boldsymbol{\theta}_{\mathrm{g}}$, another optimal solution for a new set of performance matrices ($\mathbf{Q}$, $\mathbf{R}$) in $\gamma(\cdot)$ have to be determined until enough information is collected [49]. A poor identification result may be obtained if the model structure is unsuitable for the current operating condition. Therefore, the identification result can be used to indicate a change in the model structure.

If N_5 points are chosen from the system trajectory during the interval $[t_{\mathrm{learn}}, t_{\mathrm{final}}]$, such that $N_5 \geqslant N_1 + mN_2$ is a sufficiently large positive integer, then the parameter identification problem can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\hat{\boldsymbol{\theta}}_{\mathrm{f}},\hat{\boldsymbol{\theta}}_{\mathrm{g}}} \quad & J_{\mathrm{para}}(\hat{\boldsymbol{\theta}}_{\mathrm{f}},\hat{\boldsymbol{\theta}}_{\mathrm{g}}) = \boldsymbol{\xi}^2 \\ s.t. \quad & \begin{cases} \boldsymbol{\theta}_{\mathrm{f}}^{\min} \leqslant \hat{\boldsymbol{\theta}}_{\mathrm{f}} \leqslant \boldsymbol{\theta}_{\mathrm{f}}^{\max} \\ \boldsymbol{\theta}_{\mathrm{g}}^{\min} \leqslant \hat{\boldsymbol{\theta}}_{\mathrm{g}} \leqslant \boldsymbol{\theta}_{\mathrm{g}}^{\max} \end{cases} \end{aligned} \tag{63}$$

where $\boldsymbol{\xi} = [\xi_1, \xi_2, \cdots, \xi_{N_5}]$ is the approximation error. Therefore, by using the HJB equation as a bridge, the process model parameters can be identified. To guarantee the identification performance, the range of the model parameters can be first determined based on process knowledge and then by an optimization algorithm to find the optimal model parameters that produce the minimum approximation error.

In practice, the implementation involves several computational steps. First, the partial derivatives in (61) are approximated using the basis function representations and the collected system trajectories. The optimization problem in (63) is typically solved using nonlinear least squares algorithms, aiming to minimize the approximation error $\boldsymbol{\xi}$ over a sufficiently rich set of sampled data points. Main computational challenges include the high dimensionality of the parameter space, potential non-convexity of the objective function, and sensitivity to measurement noise. Careful initialization and proper regularization are often required to ensure convergence and identify physically meaningful parameters. For large-scale problems, parallel computing strategies can be adopted to reduce computational costs.

4. Case study

To demonstrate the feasibility and performance of the LMHAC approach, a continuous stirred tank reactor (CSTR) that forms part of a hydrometallurgical purification process is selected as a test case [50]. Although the model parameters are obtained from real industrial data, the LMHAC approach is evaluated only in simulation. We adopt polynomial and time-modulated basis functions for the value function and policy approximation. This choice is motivated by the strong nonlinear approximation capability and computational efficiency of polynomials for chemical reactor dynamics and the ability of time-modulated terms to better capture finite-horizon effects. The specific combinations and orders are selected based on the system's nonlinearity and practical convergence observed in preliminary simulations.

4.1. Case 1

4.1.1. Process description

The purification process is essential in the hydrometallurgy process described here [50]. The function of the purification process is to remove the impurities, such as metal ions, from a sulfate solution. These impurities are harmful to the subsequent electrowinning process and also to the final metal product quality. A purification process consists of several cascaded CSTRs and a thickener. In each CSTR, an additive removes the metal ion impurities. The reaction involved can be described as follows

$$\mathrm{Im}^{m+} + \mathrm{Ad} = \mathrm{Im} + \mathrm{Ad}^{m+} \tag{64}$$

where Im and Ad are the impurity and additive, respectively. m is the number of electrons exchanged between the impurity and additive. After retention in the reactors, the solution is delivered to the thickener, where solid-liquid separation takes place. The purified overflow is transported through the following process:

Reaction (64) is essentially an oxidation-reduction reaction or electrode reaction. According to the mass balance principle and electrode reaction mechanism [50], the dynamics of a single reactor can be modeled as follows:

$$\frac{dc}{dt} = \frac{f_{\mathrm{in}}}{V}c_{\mathrm{in}} - \frac{f}{V}c - rc \tag{65}$$

$$r = A_0\beta g_s \exp\left(-\frac{E_e + 2\eta\mathrm{F}(e_{\mathrm{orp}} - e_{\mathrm{eq}})}{\mathrm{R}T}\right) \tag{66}$$

where c, r and f are the outlet impurity ion concentration (g/m^3), reaction rate (h^{-1}), and outlet flow rate (m^3/h) of the reactor respectively. V is the volume of the reactor (m^3). c_{in} and f_{in} are the impurity ion concentration (g/m^3) and the flow rate (m^3/h) of the inlet solution of the reactor, respectively. e_{orp} is the Oxidation Reduction Potential (ORP) (V). The physical meanings of the other parameters in the process model are listed in Table 2.

Table 2: Physical meaning of the model parameters [50]

Parameter (unit)	Physical meaning
$A_0(s^{-1})$	frequency factor of the reaction
$\beta(-)$	reaction surface area available on a unit area of the crystal nucleus
$g_s(-)$	weight of crystal nucleus per unit volume of the reactor
$E_e(\text{J}\cdot\text{mol}^{-1})$	standard activation energy of the reaction
$\eta(-)$	variation factor between the electrode potential and the cathode activation energy
$e_{\text{eq}}(\text{V})$	equilibrium potential of the cathode reaction
$T(\text{K})$	reaction temperature
$\text{F}(\text{C}\cdot\text{mol}^{-1})$	Faraday constant, F = 96485
$\text{R}(\text{J}\cdot\text{mol}^{-1}\text{K}^{-1})$	ideal gas constant, R = 8.314

In practice, A_0, β and g_s can be combined as a new parameter $A_\beta = A_0\beta g_s$. Therefore, the overall reactor model is:

$$\frac{dc}{dt} = \frac{f_{\text{in}}}{V}c_{\text{in}} - \frac{f}{V}c - A_\beta \exp(-\frac{E_e + 2\eta\text{F}(e_{\text{orp}} - e_{\text{eq}})}{\text{R}T}c) \quad (67)$$

In (67), F and R are constants. The solution temperature T is usually measured online and is constant. It is also considered a constant in the simulation. $\eta \in (0, 1)$ represents the influence of the ORP change on the activation energy of the cathode reactions. In this simulation, a balanced influence of the ORP change on the cathode and anode reactions is assumed, i.e., $\eta = 0.5$. A_β, E_e, and e_{eq} represent physical quantities and have different values under different operating conditions. Taking cobalt removal as an example, if the copper ion concentration or the crystal content in the reactor is higher, then A_β is higher. If the copper ion concentration in the reactor is lower, then e_{eq} is more negative, and E_e is higher. To reduce the number of parameters to estimate, (67) can be reformulated as:

$$\frac{dc}{dt} = \frac{f_{\text{in}}}{V}c_{\text{in}} - \frac{f}{V}c - Acu_{\text{origin}} \quad (68)$$

where $u_{\text{origin}} = e^{\alpha e_{\text{orp}}}$, $\alpha = -\frac{2\eta F}{\text{R}T}$, $A = A_\beta \exp(-\frac{E_e - 2\eta\text{F}e_{\text{eq}}}{\text{R}T})$. The only parameter to identify is A, corresponding to $\boldsymbol{\theta}$.

4.1.2. Case study setup

In the case study, the process is simulated using industrial data obtained from a real plant. For commercial reasons, the data are scaled and desensitized. Four data sets corresponding to four typical operating conditions (named as W_I, W_II, W_III and W_IV) were selected to identify the model parameters which are then used to simulate the process. It was assumed that the value of model parameter A was known under condition W_I and W_III, and was unknown under operating condition W_II and W_IV. The value of the model parameter under the four operating conditions and the feature of the four operating conditions are shown in Table 3.

Table 3: Values of model parameters under different operating conditions

Operating condition	Description	A
W_I	Normal operating condition	2.0188×10^{-8}
W_II	The flow rate of inlet solution is large, the retention time in the reactor is decreased	1.9957×10^{-8}
W_III	The concentration of another more active impurity in the inlet solution is high, a large amount of additive is consumed in removing another impurity	1.7809×10^{-8}
W_IV	The inlet flow rate is large, and the concentration of another more active impurity is high	1.4033×10^{-8}

The set points of the outlet impurity ion concentration of the reactor under each operating condition were determined by a higher-level economic optimization unit. The initial value of c in (68) is 8 mg/L. The optimal set points for the four operating conditions are 5.5 mg/L, 8.0 mg/L, 6.0 mg/L, and 8.5 mg/L, respectively. The length of each optimization horizon is 2 hours. For ADP, the first hour is used for the learning stage and the second hour for the control-identification stage. In each case, the original system model is reformulated as:

$$\frac{dx}{dt} = (\frac{f_{\text{in}}}{V}c_{\text{in}} - \frac{f}{V}c^*) - \frac{f}{V}x - \bar{A}(x + c^*)u' \quad (69)$$

Where:

$$x = c - c^*$$
$$\bar{A} = A_\beta \exp(-\frac{E_e + 2\eta\text{F}(\bar{e} - e_{\text{eq}})}{\text{R}T})$$
$$\bar{e} = \frac{e_{\text{up}} + e_{\text{low}}}{2}$$

c^* is the set-point of outlet impurity ion concentration. $[e_{\text{low}}, e_{\text{up}}]$ is the range of ORP specified in operational guidelines. (69) corresponds to (1) in Section 2.

The control input is further formulated as follows:

$$u' = \exp(-\frac{2\eta\text{F}(e_{\text{orp}} - \bar{e})}{\text{R}T})$$
$$= \frac{1}{2}[\exp(-\frac{2\eta\text{F}(e_{\text{up}} - \bar{e})}{\text{R}T}) + \exp(-\frac{2\eta\text{F}(e_{\text{low}} - \bar{e})}{\text{R}T})] + u$$

where $u \in [\exp(-\frac{2\eta\text{F}(e_{\text{low}} - \bar{e})}{\text{R}T}), \exp(-\frac{2\eta\text{F}(e_{\text{up}} - \bar{e})}{\text{R}T})]$ corresponds to $\boldsymbol{u}$ in (1). The ORP range for operating condition W_II is $[-0.515, -0.545]$, and for operating condition W_IV is $[-0.535, -0.565]$.

Nonlinear MPC is applied under operating conditions W_I and W_III with known model parameters. The finite horizon ADP

algorithm approximates the optimal controller under operating conditions W_{II} and W_{IV}. For comparison, the nonlinear MPC controller is also applied under operating conditions W_{II} and W_{IV}. The prediction horizon and control horizon for the MPC are 100 and 50, respectively. The *fminunc* function in Matlab is the solver used to solve the objective function.

The basis functions for the approximation of the cost function and control policy are:

$$\boldsymbol{\phi}(\boldsymbol{x},t) = [x^2, x^4, x^6, x^8, x^2\tau, x^4\tau, x^6\tau, x^8\tau, x^2e^{-\tau}]^{\mathrm{T}}$$
$$\boldsymbol{\psi}(\boldsymbol{x},t) = [x, xe^{-\tau}, x\tau^2, x^2\tau, x^2e^{-\tau}, x^2\tau^2, x^3\tau, x^3e^{-\tau}, x^3\tau^2]^{\mathrm{T}}$$

where $\tau = (t_{\mathrm{learn}} - t)/t_{\mathrm{learn}}$, $t_{\mathrm{learn}} = 60\mathrm{min}$, $t_{\mathrm{final}} = 120\mathrm{min}$. The design parameters for the initial admissible control are $c_1 = 0.01$, $c_2 = 0.01$, $\eta = 0.01$. The initial value of $\hat{\boldsymbol{k}}$ is:

$$\hat{\boldsymbol{k}} = [-0.01 \quad -0.01 \quad -0.01 \quad -0.01$$
$$-0.01 \quad -0.01 \quad -0.01 \quad -0.01 \quad -0.01]$$

The stopping criteria is if the iteration number reaches 200 or the norm difference of $\hat{\boldsymbol{k}}$ is less than 10^{-8}. Excitation noise is added to the initial admissible control as follows:

$$u = -H(v + e_{\mathrm{noise}})$$
$$e_{\mathrm{noise}} = 100\sum_{i=-3}^{2}\sin(10^{\mathrm{i}}\mathrm{t})$$

where v can be obtained using the result in Section 3.2, see (10)-(13) and (23)-(26).

4.1.3. Results and discussion

The control results of the LMHAC approach are shown in Figs. 6 to 12. Fig. 6 shows the outlet impurity ion concentration trajectory under the four different operating conditions (OCs). It can be observed that the LMHAC approach can drive the process to its set point under all OCs, which indicates the feasibility of LMHAC. Under operating conditions W_{I} and W_{III}, the optimal control is derived using MPC. Under operating conditions W_{II} and W_{IV}, an initial admissible control is first applied to guarantee the stability of the process. It is observed that the outlet impurity ion concentration converges toward its set point during the learning stage of the ADP control period. In the control-identification stage, the approximated optimal control is obtained by learning from the 'input-state' information collected during the learning stage. Therefore, the approximated optimal control drives the outlet impurity ion concentration trajectory during the control-identification stage.

In order to analyze the control performance when the model parameters are unknown, the results for operating conditions W_{II} and W_{IV} are discussed in detail. During the learning stage $[t_0, t_{\mathrm{learn}}]$ of the ADP control period (see Fig. 5), the initial admissible control is applied to the system. As shown by the red solid lines in Figs. 7 and 10, the system state moves asymptotically towards the equilibrium point under operating conditions W_{II} and W_{IV} during $[t_0, t_{\mathrm{learn}}]$, which indicates the feasibility of the initial admissible control. The approximated optimal control is derived via iteration when the learning stage is finished. As shown in Figs. 9 and 12, the iteration stopped after 8 and 9 iterations under the two operating conditions. The resulting weight vector $\hat{\boldsymbol{k}}$ associated with control input under operating condition W_{II} is:

$$\hat{\boldsymbol{k}} = [-1.935146 \quad -1.408442 \quad 1.302982$$
$$-0.964120 \quad 0.293709 \quad 1.760781$$
$$0.001295 \quad -0.069812 \quad 0.296231]$$

and for operating condition W_{IV} is:

$$\hat{\boldsymbol{k}} = [-1.920634 \quad -1.481995 \quad 1.166985$$
$$-1.343727 \quad 0.650264 \quad 2.047220$$
$$-0.832558 \quad 0.602731 \quad 0.883857]$$

The initial admissible control is replaced by the approximated optimal control during the control-identification stage $[t_{\mathrm{learn}}, t_{\mathrm{final}}]$. The state trajectories driven by the approximated optimal controls under operating conditions W_{II} and W_{IV} are shown by the blue solid lines in Figs. 7 and 10. For comparison, the red dashed lines show the state trajectories driven by the initial admissible control. The performance of the approximated optimal control is better than the initial admissible control. However, the advantage of the approximated optimal control is not obvious due to the relatively small initial state value at t_{learn}.

The 'input-state' information collected during the control-identification stage $[t_{\mathrm{learn}}, t_{\mathrm{final}}]$ is then used to identify the unknown model parameter A of (68). The parameter identification problem is solved using particle swarm optimization (PSO). The parameter identification results and average relative errors (AREs) under operating conditions W_{II} and W_{IV} are shown in Table 4, which lists the identification results of 10 different runs. As PSO is a swarm-based random optimization algorithm, the identification results of different runs are different. However, in different runs, the AREs of parameter identification under both operating conditions are small. This indicates acceptable identification accuracy. The simulation assumes that the model structure is the same for all four operating conditions. If the model structure changes under operating conditions W_{II} and W_{IV}, the parameter identification result may deteriorate, which can indicate that the model structure has changed.

To compare the performance of MPC and ADP, MPC was also applied to the process on the interval $[t_0, t_{\mathrm{final}}]$ under operating conditions W_{II} and W_{IV} given the value of the unknown model parameter. The approximated optimal control given by the CFADP is also applied to the system on the interval $[t_0, t_{\mathrm{final}}]$ under operating conditions W_{II} and W_{IV}, as shown by the blue dashed line and the purple solid line in Figs. 7 and 10. In addition, the corresponding control input trajectories obtained by these two approaches are shown in Figs. 8 and 11. In these figures, the input u is transferred to the ORP, which is the physically manipulated variable of the process. It can be observed from these results that the approximated optimal control can reach the set point in finite time under both operating conditions. Because of an approximation

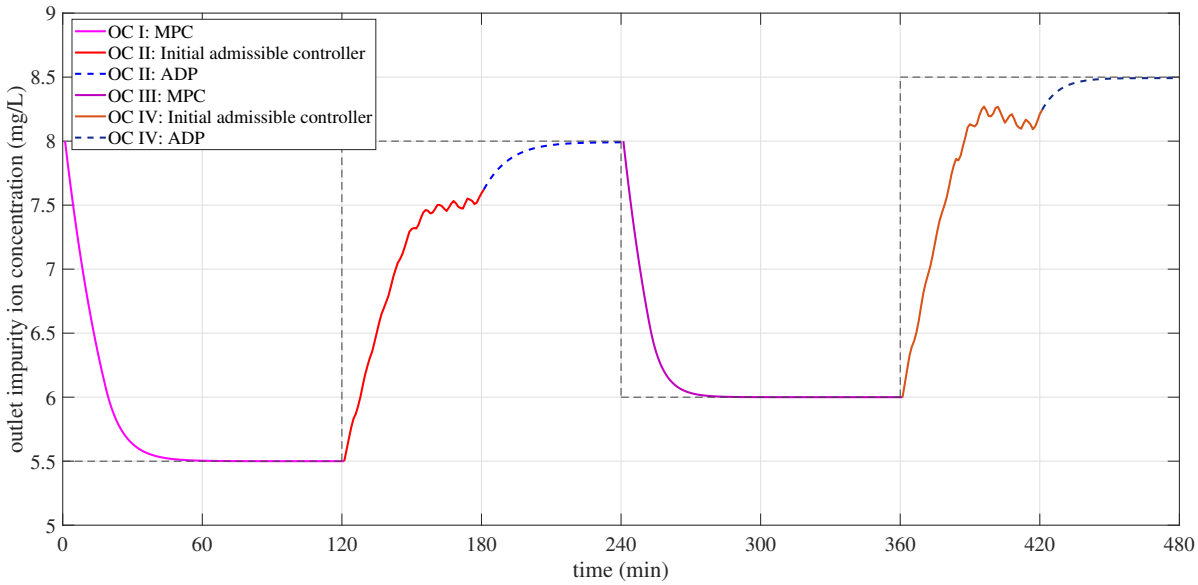

Figure 6: State trajectories during the experiment

error, there is a small difference between the state and input trajectories generated by MPC and ADP. So, the control input and control performance obtained using ADP and MPC are close. This indicates the feasibility of using ADP instead of MPC when the model parameters (and structure) are unknown. Therefore, using LMHAC, the optimal or approximated optimal control can be obtained under operating conditions with known and unknown model parameters. Moreover, the unknown model parameters can be identified by approximating the HJB equation along the system trajectory driven by the approximated optimal control. By analyzing the identification result, one can judge whether or not the model structure has changed. This can incrementally enlarge the known domain of model parameters in the comprehensive state space.

4.2. Case 2

Case 1 presents the application of LMHAC to a single state system. To further test its ability, especially the performance of CFADP, another CSTR case with 2 states is studied in this subsection [58].

4.2.1. Process description

Consider a CSTR where a reversible reaction $2A \rightleftharpoons B$ takes place, where A and B are two different species. The dynamics of the CSTR is

$$\begin{aligned} \dot{x_1} &= -2k_a x_1^2 - c_1 x_1 + 2k_b x_2 + \frac{F}{V}u \\ \dot{x_2} &= k_a x_1^2 + c_2 x_1 - (\frac{F}{V} + k_b)x_2 \end{aligned} \tag{70}$$

where $x_1 = C_A - \bar{C}_A$, $x_2 = C_B - \bar{C}_B$ are the deviation of the outlet concentrations of species A and B, respectively. k_a and

Table 4: Model parameter identification results using LMHAC

Operating condition	Real value	Identified value	ARE
II	1.9957×10^{-8}	2.0780×10^{-8}	4.1238%
		2.0127×10^{-8}	0.8522%
		2.0142×10^{-8}	0.9273%
		2.0177×10^{-8}	1.1035%
		2.0281×10^{-8}	1.6238%
		2.0307×10^{-8}	1.7517%
		1.9138×10^{-8}	4.1020%
		1.9446×10^{-8}	2.5589%
		1.9449×10^{-8}	2.5450%
		2.0607×10^{-8}	3.2580%
IV	1.4033×10^{-8}	1.4206×10^{-8}	1.2338%
		1.4264×10^{-8}	1.6408%
		1.4157×10^{-8}	0.8814%
		1.4062×10^{-8}	0.2018%
		1.4598×10^{-8}	4.0231%
		1.4257×10^{-8}	1.5963%
		1.4343×10^{-8}	2.2066%
		1.4399×10^{-8}	2.6074%
		1.4403×10^{-8}	2.6339%
		1.4427×10^{-8}	2.8073%

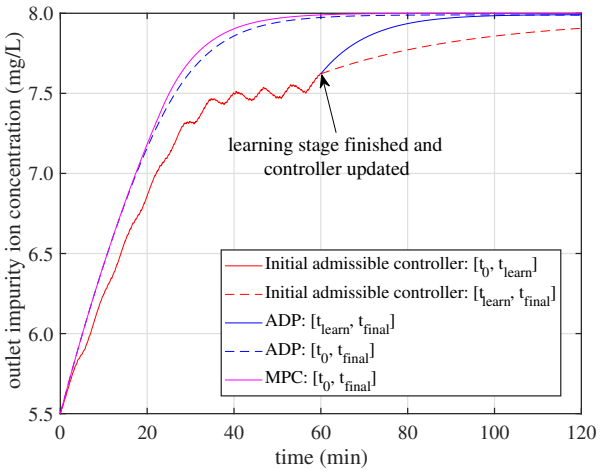

Figure 7: State trajectories using the initial admissible control and ADP under operating condition W_{II} (t_{learn} = 60min, t_{final} = 120min)

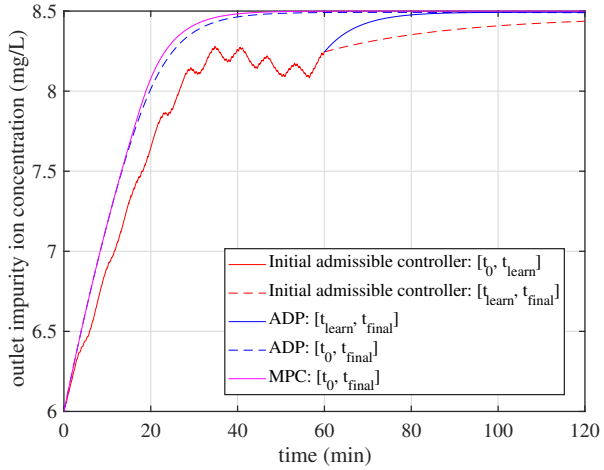

Figure 10: State trajectories using the initial admissible control and ADP under operating condition W_{IV} (t_{learn} = 60min, t_{final} = 120min)

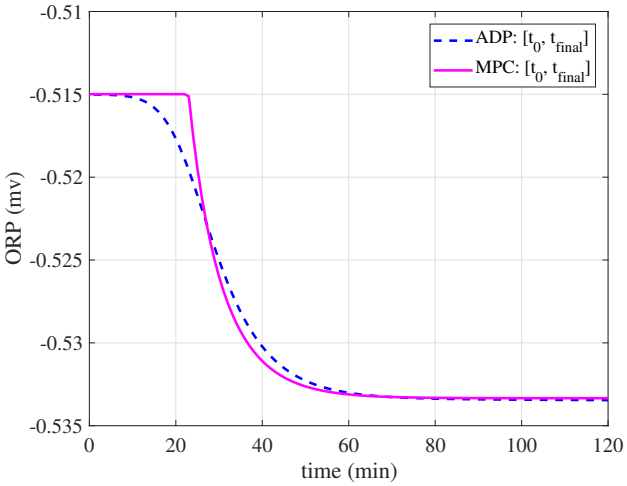

Figure 8: Input trajectories using MPC and ADP under operating condition W_{II}

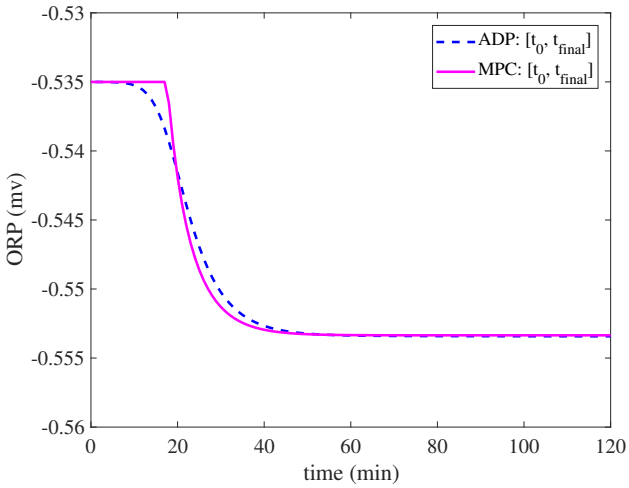

Figure 11: Input trajectories using MPC and ADP under operating condition W_{IV}

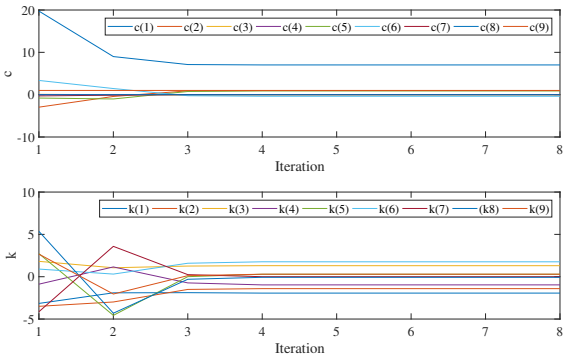

Figure 9: Iteration of $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{k}}$ under operating condition W_{II}

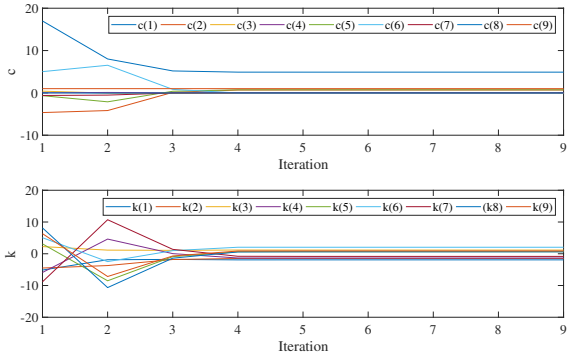

Figure 12: Iteration of $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{k}}$ under operating condition W_{IV}

k_b are the forward and reverse reaction rate constants of species B, respectively. $u = C_{A\text{in}} - \bar{C}_{A\text{in}}$ is the inlet concentration of species A. V and F are the volume and flux, respectively. $c_1 = 4k_a\bar{C}_A + F/V$ and $c_2 = 2k_a\bar{C}_A$ are constants. k_b is assumed unknown. The details of model parameters are listed below.

Table 5: Model parameter values

Variable	Value
k_a	0.05
k_b	0.01
c_1	0.338
c_2	0.159
F/V	0.02
$\bar{C}_A$	1.59
$\bar{C}_B$	4.21
$C_{A\text{in}}$	10
$x_1(0)$	0
$x_2(0)$	-1.71
u	[-10, 10]

4.2.2. Case study setup

In the simulation, $\mathbf{Q}_s = \text{diag}(10, 10)$, $\mathbf{R} = 0.1$, $\mathbf{H}(v) = 10\tanh(v/10)$. The basis functions in Φ include x_1^2, x_2^2, x_1x_2, $x_1^4\tau$, $x_2^4\tau$, $x_1^2x_2^2e^{-\tau}$, $x_1x_2^3e^{-\tau}$, $x_1^3x_2e^{-\tau}$. The basis functions in Ψ include x_1, x_2, x_1x_2, x_1^2, $x_1^2\tau$, $x_1^2e^{-\tau}$, x_2^2, $x_2^2\tau$, $x_2^2e^{-\tau}$. The initial weight is $\hat{\mathbf{k}}^{(0)} = [-1 \quad -1 \quad -1 \quad -1]$. The exploration noise was set such that

$$\mathbf{u} = -\mathbf{H}(0.5\mathbf{R}^{-1}\hat{\mathbf{k}}^{(0)}\Psi + e_{\text{noise}})$$

$$e_{\text{noise}} = 100\sum_{i=-3}^{2}\sin(10^i t)$$

4.2.3. Results and discussion

The results of applying LMHAC are shown in Fig. 13. The resulting weight vector $\hat{\boldsymbol{k}}$ associated with the control input is:

$$\begin{aligned}\hat{\boldsymbol{k}} = [&-0.311975 \quad 1.635121 \quad -0.998924\\ &2.071811 \quad -1.763905 \quad -1.430438\\ &0.050369 \quad 0.023781 \quad -0.029703]\end{aligned}$$

The parameter identification results and average relative errors (AREs) are shown in Table 6, which lists the identification results of 10 different runs. Similar to Case 1, the identification error is within 5%. The initial admissible controller and the controller derived by CFADP can both force the states to converge. However, the result obtained using CFADP has no steady-state deviation. This illustrates the effectiveness of introducing two different parts in the basis function to approximate two different costs, i.e., the integrated intermediate cost and the terminal error cost.

Table 6: Model parameter identification results for Case 2

Real value	Identified value	ARE
0.01	0.010126	1.2611%
	0.010175	1.7534%
	0.009948	0.5240%
	0.010179	1.7946%
	0.010112	1.1247%
	0.009978	0.2152%
	0.010409	4.0924%
	0.010219	2.1921%
	0.010140	1.4001%
	0.010216	2.1564%

5. Conclusions

This paper proposed a learning-based moving horizon autonomous control (LMHAC) framework integrating MPC, ADP, and process modeling to handle chemical reactors operating under known and unknown model parameters.

The proposed approach enables autonomous switching between MPC and a constrained finite horizon ADP (CFADP), ensuring stability through a Lyapunov-based initial admissible control and incrementally expanding the known parameter domain via online identification using the HJB equation. Two simulation case studies validated the feasibility and performance of LMHAC. In the first case, involving an industrial purification process, the framework drove the process outputs to their respective set-points under all four tested operating conditions. Under unknown-parameter conditions W_{II} and W_{IV}, the CFADP converged to its near-optimal policy within 8 and 9 iterations, respectively, and the identified model parameter values achieved average relative errors below 5% over 10 independent runs. The resulting control performance was comparable to MPC using the true model. In the second case, a two-state CSTR, the proposed method again achieved convergence without steady-state deviation, with parameter identification errors within 5%. These results confirm that LMHAC can consistently deliver optimal or near-optimal control and accurate parameter identification in nonlinear chemical processes, even under parameter uncertainty.

Future work will address robustness against measurement noise and disturbances, guarantee global stability, and extend the framework to high-dimensional and large-scale systems for industrial deployment.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding information

This work was supported by the Science and Technology Innovation Program of Hunan Province (2022RC1089), and Central South University Innovation-Driven Research Programme (2023CXQD040).

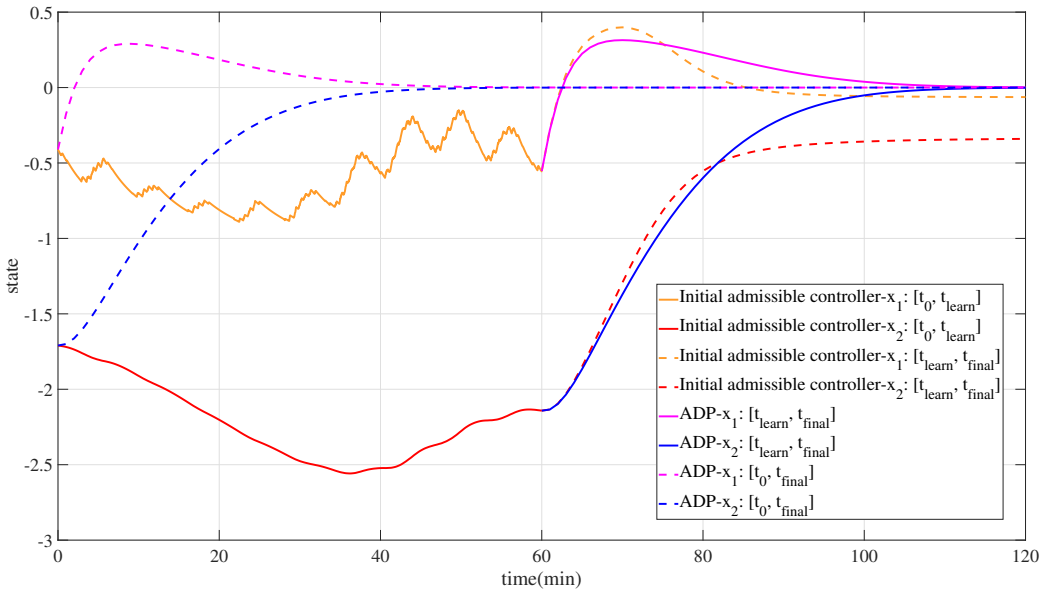

Figure 13: State trajectories using the initial admissible control and ADP

References

[1] A. Caspari, C. Tsay, A. Mhamdi, M. Baldea, A. Mitsos, The integration of scheduling and control: Top-down vs. bottom-up, Journal of Process Control 91 (2020) 50–62.

[2] J. Liu, H. Sun, Y. Zhang, J. Hu, T. Zou, Steady-state sequence optimization with incremental input constraints in two-layer model predictive control, ISA Transactions 128 (2022) 144–158.

[3] J. Wang, M. Wei, X. Xing, Static gain estimation for nonlinear dynamic systems from steady-state values hidden in historical data, ISA Transactions 120 (2022) 78–88.

[4] A. D. Celebi, S. Sharma, A. V. Ensinas, F. Maréchal, Next generation cogeneration system for industry–combined heat and fuel plant using biomass resources, Chemical Engineering Science 204 (2019) 59–75.

[5] M. Rafiei, L. A. Ricardez-Sandoval, New frontiers, challenges, and opportunities in integration of design and control for enterprise-wide sustainability, Computers and Chemical Engineering 132 (2020) 106610.

[6] L. Stander, M. Woolway, T. L. Van Zyl, Surrogate-assisted evolutionary multi-objective optimisation applied to a pressure swing adsorption system, Neural Computing and Applications 37 (2025) 739–755.

[7] J. Kager, C. Herwig, I. V. Stelzer, State estimation for a penicillin fed-batch process combining particle filtering methods with online and time delayed offline measurements, Chemical Engineering Science 177 (2018) 234–244.

[8] R. Ortega, A. Bobtsov, D. Dochain, N. Nikolaev, State observers for reaction systems with improved convergence rates, Journal of Process Control 83 (2019) 53–62.

[9] X. Zhang, Q. Liu, F. Ding, A. Alsaedi, T. Hayat, Recursive identification of bilinear time-delay systems through the redundant rule, Journal of the Franklin Institute 357 (2020) 726–747.

[10] W. Xiong, X. Shi, Soft sensor modeling with a selective updating strategy for gaussian process regression based on probabilistic principle component analysis, Journal of the Franklin Institute 355 (2018) 5336–5349.

[11] Z. Yang, Z. Ge, Monitoring and prediction of big process data with deep latent variable models and parallel computing, Journal of Process Control 92 (2020) 19–34.

[12] X. Shi, W. Xiong, Adaptive ensemble learning strategy for semi-supervised soft sensing, Journal of the Franklin Institute 357 (2020) 3753–3770.

[13] G. Shao, Z. He, W. Xiao, G. He, X. Ruan, X. Jiang, On-line monitoring and analysis of membrane-assisted internal seeding for cooling crystallization of ammonium persulfate, Chemical Engineering Science 263 (2022) 118081.

[14] V. Botelho, J. O. Trierweiler, M. Farenzena, MPC model monitoring and diagnosis for non-square systems, Journal of Process Control 97 (2021) 26–44.

[15] P. Tang, K. Peng, R. Jiao, A process monitoring and fault isolation framework based on variational autoencoders and branch and bound method, Journal of the Franklin Institute 359 (2022) 1667–1691.

[16] M. Zhong, T. Xue, S. X. Ding, A survey on model-based fault diagnosis for linear discrete time-varying systems, Neurocomputing 306 (2018) 51–60.

[17] W. Bounoua, A. Bakdi, Fault detection and diagnosis of nonlinear dynamical processes through correlation dimension and fractal analysis based dynamic kernel pca, Chemical Engineering Science 229 (2021) 116099.

[18] M. Xia, T. Yu, K. Shi, S. He, Observer-based event-impulse mixed triggered fault detection for nonlinear semi-markov jump systems, Journal of the Franklin Institute 359 (2022) 5078–5096.

[19] E. J. Meyer, M. C. Olivier, L. Matumba, I. K. Craig, Model predictive control simulation, implementation and performance assessment of a coal comminution circuit, Minerals Engineering 144 (2019) 106024.

[20] Y. Wang, Y. Chen, J. Zhang, Q. Zhang, Reliability evaluation method for pid feedback control system considering performance degradation, Journal of the Franklin Institute 361 (2024) 106814.

[21] Z. Sun, Y. Deng, J. Wang, H. Li, H. Cao, Improved cascaded model-free predictive speed control for pmsm speed ripple minimization based on ultra-local model, ISA transactions 143 (2023) 666–677.

[22] C. Zhou, L. Jia, Y. Zhou, A two-stage robust iterative learning model predictive control for batch processes, ISA transactions 135 (2023) 309–324.

[23] D. Li, K. Lu, Y. Cheng, H. Wu, H. Handroos, S. Yang, Y. Zhang, H. Pan, Nonlinear model predictive control—cross-coupling control with deep neural network feedforward for multi-hydraulic system synchronization control, ISA transactions 150 (2024) 30–43.

[24] Y. Wang, Adaptive job shop scheduling strategy based on weighted q-learning algorithm, Journal of Intelligent Manufacturing 31 (2020) 417–432.

[25] S. Mayer, T. Classen, C. Endisch, Modular production control using deep reinforcement learning: proximal policy optimization, Journal of

Intelligent Manufacturing 32 (2021) 2335–2351.
[26] A. Kuhnle, J. P. Kaiser, F. Theiß, N. Stricker, G. Lanza, Designing an adaptive production control system using reinforcement learning, Journal of Intelligent Manufacturing 32 (2021) 855–876.
[27] X. Hou, J. Zhang, C. He, C. Li, Y. Ji, J. Han, Crash mitigation controller for unavoidable t-bone collisions using reinforcement learning, ISA transactions 130 (2022) 629–654.
[28] Z. Yan, F. Xu, J. Tan, H. Liu, B. Liang, Reinforcement learning-based integrated active fault diagnosis and tracking control, ISA transactions 132 (2023) 364–376.
[29] A. Nabeel, A. Lasheen, A. L. Elshafei, E. A. Zahab, Fuzzy-based collective pitch control for wind turbine via deep reinforcement learning, ISA transactions 148 (2024) 307–325.
[30] Z. Zhou, J. Zhang, Y. Wang, D. Yang, Z. Liu, Adaptive neural control of superheated steam system in ultra-supercritical units with output constraints based on disturbance observer, IEEE Transactions on Circuits and Systems I: Regular Papers 72 (2025) 2701–2711.
[31] Y. Liang, Y. Luo, H. Su, X. Zhang, H. Chang, J. Zhang, Event-triggered explorized irl-based decentralized fault-tolerant guaranteed cost control for interconnected systems, Neurocomputing 615 (2025) 128837.
[32] D. P. Bertsekas, Dynamic programming and optimal control, volume I, 3 ed., Athena scientific, 2012.
[33] D. P. Bertsekas, Dynamic programming and optimal control, volume II, 3 ed., Athena scientific, 2011.
[34] ´A. Á. Feldbâum, Dual control theory problems, IFAC Proceedings Volumes 1 (1963) 541–550.
[35] J. Sternby, A simple dual control problem with an analytical solution, IEEE Transactions on Automatic Control 21 (1976) 840–844.
[36] Q. Li, H. Yang, Y. Xia, H. Zhao, Switched model predictive control for nonholonomic mobile robots under adaptive dwell time, IEEE Transactions on Cybernetics 54 (2024) 3444–3453.
[37] R. R. Bitmead, M. Gevers, V. Wertz, Adaptive optimal control and GPC: robustness analysis, in: Proc. European Control Conf., Grenoble, France, 1991, pp. 1099–1104.
[38] P. Li, Y. Kang, Y. Zhao, T. Wang, Networked dual-mode adaptive horizon mpc for constrained nonlinear systems, IEEE Transactions on Systems, Man, and Cybernetics: Systems 51 (2021) 7435–7449.
[39] K. Kumar, S. C. Patwardhan, S. Noronha, Development of adaptive dual predictive control schemes based on wiener–hammerstein models, Journal of Process Control 119 (2022) 68–85.
[40] J. Duan, Z. Liu, S. E. Li, Q. Sun, Z. Jia, B. Cheng, Adaptive dynamic programming for nonaffine nonlinear optimal control problem with state constraints, Neurocomputing 484 (2022) 128–141.
[41] Y. Xu, T. Li, Y. Yang, S. Tong, C. L. P. Chen, Simplified adp for event-triggered control of multiagent systems against fdi attacks, IEEE Transactions on Systems, Man, and Cybernetics: Systems 53 (2023) 4672–4683.
[42] G. Zhao, J. Sun, Y. Yan, H. Zhang, A nonlinear critical surface with input constraints on costate-adaptive dynamic programming, IEEE Transactions on Circuits and Systems II: Express Briefs 70 (2023) 4088–4092.
[43] D. Ernst, M. Glavic, F. Capitanescu, L. Wehenkel, Reinforcement learning versus model predictive control: a comparison on a power system problem, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39 (2009) 517–529.
[44] B. Depraetere, M. Liu, G. Pinte, I. Grondman, R. Babuška, Comparison of model-free and model-based methods for time optimal hit control of a badminton robot, Mechatronics 24 (2014) 1021–1030.
[45] C. Gao, D. Wang, Comparative study of model-based and model-free reinforcement learning control performance in hvac systems, Journal of Building Engineering 74 (2023) 106852.
[46] D. Görges, Relations between model predictive control and reinforcement learning, IFAC-PapersOnLine 50 (2017) 4920–4928.
[47] X. Xu, H. Chen, C. Lian, D. Li, Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances, IEEE Transactions on Neural Networks and Learning Systems 29 (2018) 6202–6213.
[48] L. Dong, J. Yan, X. Yuan, H. He, C. Sun, Functional nonlinear model predictive control based on adaptive dynamic programming, IEEE Transactions on Cybernetics 49 (2018) 4206–4218.
[49] K. Subbarao, P. Nuthi, G. Atmeh, Reinforcement learning based computational adaptive optimal control and system identification for linear systems, Annual Reviews in Control 42 (2016) 319–331.
[50] B. Sun, C. Yang, Y. Wang, W. Gui, I. Craig, L. Olivier, A comprehensive hybrid first principles/machine learning modeling framework for complex industrial processes, Journal of Process Control 86 (2020) 30–43.
[51] R. W. Beard, G. N. Saridis, J. T. Wen, Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation, Automatica 33 (1997) 2159–2177.
[52] Y. Jiang, Z.-P. Jiang, Robust adaptive dynamic programming and feedback stabilization of nonlinear systems, IEEE Transactions on Neural Networks and Learning Systems 25 (2014) 882–893.
[53] H. K. Khalil, J. Grizzle, Nonlinear systems, Prentice hall Upper Saddle River, 2002.
[54] S. E. Lyashevskiy, Constrained optimization and control of nonlinear systems: New results in optimal control, in: Proceedings of the 35th IEEE Conference on Decision and Control, 1996, volume 1, IEEE, 1996, pp. 541–546.
[55] M. Abu Khalaf, F. L. Lewis, Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach, Automatica 41 (2005) 779–791.
[56] T. Cheng, F. L. Lewis, M. Abu Khalaf, Fixed-final-time-constrained optimal control of nonlinear systems using neural network HJB approach, IEEE Transactions on Neural Networks 18 (2007) 1725–1737.
[57] H. K. Khalil, Nonlinear Systems, Prentice Hall, New Jersey, 2002.
[58] V. Manousiouthakis, D. J. Chmielewski, On constrained infinite-time nonlinear optimal control, Chemical Engineering Science 57 (2002) 105–114.