

RESEARCH ARTICLE

2D Iterative Learning Control with Deep Reinforcement Learning Compensation for the Non-repetitive Batch Processes

Jianan Liu^{1,2,3} | Zike Zhou^{1,2,3} | Wenjing Hong^{1,2,3} | Jia Shi^{1,2,3}

¹Institute of Artificial Intelligence, Xiamen University, Xiamen, China

²Department of Chemical and Biochemical Engineering, Xiamen University, Xiamen, China

³Innovation Laboratory for Sciences and Technologies of Energy Materials of Fujian Province (IKKEM), Xiamen University, Xiamen, China

Correspondence

Jia Shi, Department of Chemical and Biochemical Engineering, Xiamen University, Xiamen, Fujian 361005, China.
Email: jshi@xmu.edu.cn

Abstract

Iterative learning control (ILC) is an advantage control strategy widely used in batch systems. Nevertheless, designing an effective iterative learning control scheme is still a critical problem for complex batch processes with non-repetitive nature and model mismatch. In this paper, we propose a two-dimensional iterative learning control-reinforcement learning (2D ILC-RL) control scheme composed of a two-dimensional ILC controller and a two-dimensional DRL compensator. Based on the 2D system theory, the 2D ILC controller is proposed to ensure the primary control performance and its stability and convergence are verified. Meanwhile, the DRL compensator counteracts the negative impact of the model mismatch and the non-repetitive nature. In addition, we proposed a real-time implementation scheme to guarantee the safety of the practical batch systems compared to the conventional online training method. Finally, the simulation results in two chemical batch processes demonstrate the proposed control method's effectiveness, significant control performance, and strong robustness.

KEYWORDS

iterative learning control, deep reinforcement learning, non-repetitive batch processes, system model mismatch

1 | INTRODUCTION

Periodic/repetitive/batch systems are extensively used in the modern manufacturing industry, such as fine chemical engineering^{1,2}, integrated circuit manufacturing³, electrical power systems⁴, computer numerical control (CNC)⁵, industrial robots^{6,7}, etc. Because of the less system information requirements, utilizing the repetitive nature of the batch processes, and superior control performance, iterative learning control (ILC) is an effective advanced control method for the batch systems⁸. Since Arimoto originally developed the ILC for the repeatable motion control of the manipulator⁹, ILC has become an essential branch of control engineering for batch processes after decades of development. Nevertheless, there are multiple practical hurdles to deploying ILC in the natural batch system.

Theoretically, the primary strict assumption of the ILC is that the process natures are entirely repeated, which means not only the process dynamics and control target are batch-invariant, but also the disturbance and initial states are completely repeated from batch to batch. However, in many practical batch processes, the system dynamics are time-varying, iteration-varying, and even time-iteration-varying,

such as the slowly time-varying pharmaceutical crystallization¹⁰, the mass-dependent pick-and-place robot manipulators¹¹, and the position-dependent permanent magnet synchronous motor¹². When repeatability cannot be guaranteed, the control performance of the ILC deteriorates, which leads to weak robustness in iterative learning. Most studies have focused almost exclusively on plant-invariant and time-varying batch processes^{13,14}, but little research has been done on iteration-varying batch systems¹⁵.

For the class of known iteration-varying batch processes, the higher-order internal model (HOIM) ILC utilizes the internal-model principle to describe the iteration-varying pattern of the system and then design the updating law to eliminate the effects of non-repetitive nature¹⁶. Yin extended the HOIM to the continuous time nonlinear dynamic systems to describe iteration-varying parameters and the adaptive control theory was successfully applied to the HOIM to guarantee the convergence of the tracking errors¹⁷. Zhou proposed the contraction mapping method based on HOMI ILC to ensure zero-error tracking along the batch direction in the discrete-time systems¹⁸. The HOIM ILC is an effective control scheme for non-repetitive batch processes. Nevertheless, knowing the form of the non-repetitive variation and coefficients is necessary

to construct the internal model. Typically, the iteration-varying pattern and parameters are unknown in practical batch processes. For the unknown iteration-varying model parameters, some studies focus on designing observers to estimate the output error and construct the control law by incorporating this error. For example, Tan proposed a robust D-type ILC based on the linear state observers¹⁹. Nevertheless, the observer-based ILC control scheme applies to linear systems, not nonlinear systems, because the observers are designed based on linear system models.

In addition to the non-repetitive plant, there is an unavoidable model mismatch in the practical control system design, which reduces the control performance. More precisely, a model mismatch universally exists between the existing system and the nominal model from the first principle or the model identification. Therefore, developing an effective ILC control scheme remains an open problem for non-repetitive systems with model mismatch.

Recently, with the rapid development of artificial intelligence technology, deep reinforcement learning (DRL), which combines reinforcement learning (RL) with deep neural network (DNN), has become a powerful method for solving optimization problems in complex dynamic systems. DRL has not only shown superintelligence in chess²⁰, computer games²¹, and intelligent trading systems²² but also achieved good control performance in industrial batch processes, such as nonlinear semi-batch polymerization²³. One of the advantages of DRL is that it can find the optimal solution by constantly exploring the environment. Therefore, the agent comprehends and restrains the unknown system uncertainty for the complex batch processes by interacting with the environment. Furthermore, the strong generalization of DNN guarantees the robustness of the control system to solve the weak robustness caused by the non-repetitive nature effectively. Thus, the DRL is considered an effective method to supplement the ILC control scheme to overcome the non-repetitive nature and model mismatch.

While DRL has impressive potential for the complex batch system^{23,24}, it remains a critical bottleneck when deploying for the real-world system. The DRL agent uses random exploration to find the optimal policy through interaction with the environments. However, this random exploration may give an unsafe control input to the industrial batch systems, leading to system instability and operation in dangerous areas.

To solve the aforementioned challenges of designing an effective control scheme for non-repetitive uncertainty batch processes, in this paper, we propose a novel two-dimensional iterative learning control-reinforcement learning (2D ILC-RL) control scheme comprised of a 2D iterative learning control as the feedback controller and a two-dimensional DRL agent as the tracking error compensator. Derived from the two-dimensional theory, the 2D iterative learning controller (2D ILC) eliminates the influence of the fixed model mismatch and time-varying model parameters while ensuring essential control performance. In addition, the convergence analysis is presented by robust stability theory. Based on the DRL theory, the 2D DRL compensator is proposed to find the proper compensating input signal to reject the tracking error

caused by iteration-varying parameters. In virtue of the primary control performance of the 2D ILC controller, the exploration of the 2D DRL compensator is safe in the batch systems. Moreover, practical implementation of the proposed method is proposed to guarantee further safety and reduce the training batches in the actual batch control systems. Finally, the numerical simulations are conducted on the linear injection molding batch process and the nonlinear continuously stirred tank reactor to demonstrate the effectiveness and feasibility of the proposed control scheme. Experimental simulation results demonstrate a significant performance advantage of the proposed algorithm.

The rest paper is organized as follows: the mathematical description and design objectives are shown in section 2. Section 3 presents the 2D ILC-RL control scheme. In section 4, the proposed control scheme is simulated on the linear injection molding process and the nonlinear batch reactor to demonstrate its applicability and effectiveness. Finally, some conclusions about this paper are illustrated in section 6.

2 | PROBLEM FORMULATION

In this paper, we consider a general class of the non-repetitive batch process with system model mismatch that is described by the following discrete-time system model:

$$\begin{cases} x_{k,t+1} = (A + \Delta A_t)x_{k,t} + (B + \Delta B_t)u_{k,t} + f(x_{k,t}, u_{k,t}, \varepsilon_t, \zeta_k) \\ y_{k,t} = Cx_{k,t} \\ x_{k,0} = x_{0,0} \end{cases}, \quad (1)$$

$$t = 0, 1, \dots, T-1; k = 1, 2, \dots$$

where k is the batch (repetition or trial) index; t denotes the discrete-time; T represents the fixed time duration of the batch; $x_{k,t}, u_{k,t}, y_{k,t}$ represent the state, input, and output of the batch process at time t of the k^{th} batch, respectively; A, B, C are the known nominal system matrices with appropriate dimensions; the function $f(\cdot, \cdot, \cdot, \cdot)$ illustrates the unknown nonlinear non-repetitive dynamics; ε_t is unknown time-dependent parameters indicating the time-varying system characteristics; ζ_k is unknown batch-dependent parameters indicating the batch-varying system characteristics; $\Delta A_t, \Delta B_t$ denote the admissible time-varying model mismatch, which is represented as:

$$\begin{aligned} \Delta A_t &= E_1 \Delta_t^A F_1 \\ \Delta B_t &= E_2 \Delta_t^B F_2 \end{aligned} \quad (2)$$

with

$$(\Delta_t^A)^T \Delta_t^A < I \quad (\Delta_t^B)^T \Delta_t^B < I \quad (3)$$

where E_1, F_1, E_2, F_2 are the real constant matrices characterizing the structure of the model mismatch; Δ_t^A, Δ_t^B are the unknown perturbation matrices.

Assumption 1. The initial state $x_{k,0}$ is identical in all iterations.

Assumption 2. The nominal models A, B, C are known according to the first principle or the model identification.

Remark 1. When the known dynamic characteristics are nonlinear, the nominal models A, B, C are obtained by linearizing the nonlinear function with the model mismatch considered as the function $f(\cdot, \cdot, \cdot, \cdot)$.

At any time t of the k^{th} batch, the tracking error is defined as:

$$e_{k,t} = y_{k,t}^r - y_{k,t} \quad (4)$$

where $y_{k,t}^r$ is the reference trajectory. Typically, for the classical ILC control scheme²⁵, the control signal $u_{k,t}$ is designed by the equation (5) to minimize the tracking error $e_{k,t}$.

$$u_{k,t} = u_{k-1,t} + L(I_{k,t}) \quad (5)$$

$$\lim_{k \rightarrow \infty} e_{k,t} \rightarrow 0 \quad (6)$$

where L is the iterative updating law function; $I_{k,t}$ indicates any observable information at time t of the k^{th} batch. Usually, the tracking error $e_{k,t}$ could converge to a minimum value from batch to batch by using the conventional ILC control strategies (5) for the known repetitive batch processes without the model mismatch. Nevertheless, for the non-repetitive batch process with the model mismatch and uncertainty, the control performance of the traditional ILC deteriorates, even unable to meet the operational requirement. Consequently, the design objective of this paper is proposed as follows:

Design objective: For non-repetitive batch processes with time-varying, iteration-varying nature and system model mismatch (1), an advanced control scheme based on the ILC and DRL is designed to meet the following requirements:

- The tracking error converges to the minimum along the time and batch index as quickly as possible.
- The influence of the plant mismatch is compensated.
- The impact of the non-repetitive nature should be as small as possible.

3 | 2D ILC-RL CONTROL SCHEME

In this section, we propose a two-dimensional iterative learning control-reinforcement learning (2D ILC-RL) control scheme, which is composed of a two-dimensional ILC controller and a two-dimensional DRL compensator, as shown in Fig. 1. Firstly, based on 2D system theory²⁶, the two-dimensional iterative learning controller is designed to ensure the primary control performance for the batch processes. Meanwhile, the two-dimensional DRL compensator based on the soft actor-critic (SAC) algorithm²⁷ is introduced to counteract the negative impact of batch-varying parameters. Finally, the practical implementation plan of the 2D ILC-RL control scheme is given.

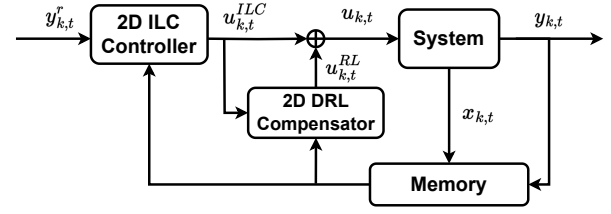


FIGURE 1 The block diagram of the two-dimensional iterative reinforcement learning control scheme.

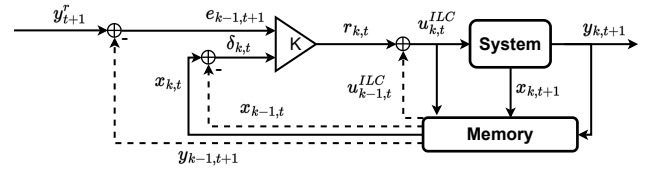


FIGURE 2 The block diagram of the two-dimensional iterative learning control system.

3.1 | Two-dimensional (2D) Iterative Learning Controller

As shown in the non-repetitive batch system (1), there are two independent axes: t time and k batch. Thus, the system (1) should be considered a 2D system. So, naturally, it is more suitable to be controlled by a 2D iterative learning controller. Therefore, for the construction of the 2D state space model, the following general iterative learning control law is considered:

$$\begin{cases} u_{k,t} = u_{k-1,t} + r_{k,t} \\ u_{0,t} = 0 \end{cases} \quad (7)$$

where $r_{k,t}$ is update law at time t of the k^{th} batch; $u_{0,t}$ indicates that the initial control signal is 0.

By defining a new state variable:

$$\delta_{k,t} = x_{k,t} - x_{k-1,t} \quad (8)$$

and substituting equation (7) into equation (1), a 2D state space model is obtained:

$$\begin{bmatrix} \delta_{k,t+1} \\ e_{k,t+1} \end{bmatrix} = (\bar{A} + \Delta \bar{A}_t) \begin{bmatrix} \delta_{k,t} \\ e_{k-1,t+1} \end{bmatrix} + (\bar{B} + \Delta \bar{B}_t) r_{k,t} + \bar{D} w_{k,t} \quad (9)$$

with subject to

$$\begin{aligned} \bar{A} &= \begin{bmatrix} A & 0 \\ -CA & I \end{bmatrix}, \bar{B} = \begin{bmatrix} B \\ -CB \end{bmatrix}, \bar{D} = \begin{bmatrix} I \\ -C \end{bmatrix} \\ \Delta \bar{A}_t &= \bar{E}_1 \Delta_t^A \bar{F}_1, \Delta \bar{B}_t = \bar{E}_2 \Delta_t^B \bar{F}_2 \\ \bar{E}_1 &= \begin{bmatrix} E_1 \\ -CE_1 \end{bmatrix}, \bar{F}_1 = \begin{bmatrix} F_1 & 0 \end{bmatrix}, \bar{E}_2 = \begin{bmatrix} E_2 \\ -CE_2 \end{bmatrix}, \bar{F}_2 = F_2 \\ w_{k,t} &= f(x_{k,t}, u_{k,t}, \varepsilon_t, \zeta_k) - f(x_{k-1,t}, u_{k-1,t}, \varepsilon_t, \zeta_{k-1}) \end{aligned}$$

where $\begin{bmatrix} \delta_{k,t} \\ e_{k-1,t+1} \end{bmatrix}$, $r_{k,t}$, $w_{k,t}$ represent the state, input and external disturbance of the 2D system, respectively; $\bar{A}, \bar{B}, \bar{D}$ are the new nominal system matrices.

Remark 2. If the available system dynamic is exact, and the system parameters are iteration-invariant, the external disturbance $w_{k,t}$ is equal to 0.

For the 2D batch system (9), the 2D ILC update law is designed as follows:

$$r_{k,t} = K \begin{bmatrix} \delta_{k,t} \\ e_{k-1,t+1} \end{bmatrix} \quad (10)$$

where the feedback gain K is designed with robust H_∞ performance by solving linear matrix inequality (LMI) according to **Theorem 1**. The overall structure of the 2D ILC controller is illustrated in Fig. 2, where the solid lines and the dashed lines represent the current batch feedback information and the previous batch information, respectively.

Theorem 1. The closed-loop 2D batch system has robust H_∞ performance γ with the robust convergence index ρ ($0 < \rho < 1$) if there exist positive block diagonal matrix $Q = \text{diag}\{Q_\delta, Q_e\}$, and positive scalars $\varepsilon > 0$, $\eta > 0$ such that the following LMI condition is satisfied:

$$\underset{Q, K, \varepsilon, \eta}{\text{Minimize}} \quad \gamma \quad (11)$$

with subject to

$$\begin{bmatrix} -\rho Q & QK^T \bar{F}_2^T & Q\bar{F}_1^T & Q\bar{A}^T & Q\bar{C}^T & 0 \\ * & -Q + \bar{H} & 0 & 0 & 0 & 0 \\ * & * & -\varepsilon I & 0 & 0 & 0 \\ * & * & * & -\eta I & 0 & \bar{D} \\ * & * & * & * & -\gamma I & 0 \\ * & * & * & * & * & -\gamma I \end{bmatrix} < 0$$

The proof is found in **Appendix A**

3.2 | Two-dimensional (2D) Deep Reinforcement Learning (DRL) Compensator

From the 2D batch model (9), the impact of external disturbance is not eliminated when the non-repetitive disturbance or unknown iteration-wise parameters variation exists. The unavailable external disturbance deteriorates the control performance, leading to weak robustness. Compared with the conventional control scheme, DRL is a data-driven machine learning method that finds the optimal policy for the complex dynamic process through meaningful interaction with the environment²⁸. Therefore, as a compensator, the DRL agent can constantly interact with the controlled batch system to find the optimal compensation signal for restraining the negative influence of the non-repetitive characteristic in the 2D ILC systems.

3.2.1 | Soft Actor-Critic (SAC)

The basic architecture of the DRL algorithm, as shown in Fig. 3, consists of two components: an agent and an environment. The agent is the self-learning decision-maker which completes a specific task through meaningful interactions with the environment. More precisely, the agent collects the state feedback from the environment and gives action according to its policy. Simultaneously, the agent can also optimize its policy according to the environment's performance assessment (reward), leading to optimal decisions gradually. From the control theory's viewpoint, the DRL algorithm's structure is similar to the closed-loop feedback control. The environment is equivalent to the controlled plant, while the agent mainly plays the role of state feedback controller. Math-

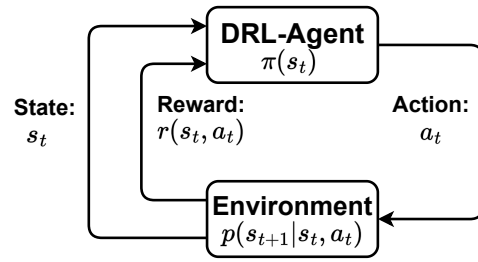


FIGURE 3 The architecture of deep reinforcement learning.

ematically, the Markov decision process (MDP) is introduced to describe the DRL system by a 5-tuple (S, A, p, R, ν) :

- State $s_t \in S$: The feedback state information from the environment to the agent. S notes the admissible sets of the state. The states of the batch process consist of observable information, such as temperatures, pressures, flow rates, etc.
- Action $a_t = \pi(s_t) \in A$: The DRL policy $\pi(s_t)$ gives the action according to the state information. A represents the entire space of the action.
- State transition function $p(s_{t+1} | s_t, a_t)$: Describing the dynamic characteristics of the environment. The next state s_{t+1} is based on the current state s_t and the action a_t . In this paper, the state transition function is equal to the non-repetitive batch system (1).
- Reward $r_t = r(s_t, a_t) \in R$: The instant reward is obtained from the real-time interaction. Rewards are generated based on a well-designed reward function $r(s_t, a_t)$, similar to the cost function in the control field.
- Discount factor ν : Apart from the instant reward, the following state-action value function $Q(s_t, a_t)$ is defined to estimate the cumulative discount reward obtained in the subsequent after time t .

$$Q(s_t, a_t) = E_{(s_t, a_t) \sim p\pi} \left[\sum_{i=0}^{\infty} \nu^i r_{t+i} \right] \quad (12)$$

where $E_{(s_t, a_t) \sim p\pi}$ represents the expected value subject to the state transition function p and current policy π .

The fundamental objective of the DRL is to find the optimal policy π_* , which maximizes the state-action value function $Q(s_t, a_t)$.

$$\begin{aligned}\pi_*(s_t) &= \operatorname{argmax}_{a_t \in A} Q(s_t, a_t) \\ &= \operatorname{argmax}_{\pi} E_{(s_t, a_t) \sim p\pi} \left[\sum_{i=0}^{\infty} \nu^i r_{t+i} \right]\end{aligned}\quad (13)$$

Lemma 1. ²⁸ (Optimal Bellman Equation) The optimal policy π_* and the optimal state-action value function $Q_*(s_t, a_t)$ satisfy the following equation:

$$Q_*(s_t, a_t) = E_{(s_{t+1}, a_{t+1}) \sim p\pi_*} [r_t + \nu Q_*(s_{t+1}, a_{t+1})] \quad (14)$$

After decades of development, many different DRL algorithms have been proposed in different industrial application scenes²⁹. In the DRL community, the soft actor-critic (SAC) is an off-policy actor-critic DRL algorithm based on the maximum entropy, outperforming prior on-policy and off-policy methods²⁷. The on-policy learning strategy, such as TRPO³⁰, PPO³¹, and A3C³², requires new samples for each or several gradient steps. Compared with them, the SAC is one off-policy learning strategy, which reuses the previous experience saved in the experience buffer to improve the sample efficiency. Moreover, maximum entropy policies substantially improve exploration efficiency and robustness in the face of model mismatches and estimation errors³³. In addition, the separation of the target network and online network, a commonly used method in DDPG³⁴, is adopted by SAC for sample-efficient learning and stable training. Hence, in this paper, the SAC algorithm is considered the original framework for constructing the 2D DRL compensator.

Typically, in the SAC algorithm, the state-action value function $Q(s_t, a_t)$ is approximated by deep neural networks. Meanwhile, the policy $\pi(s_t)$ is modeled as a Gaussian Distribution by deep neural networks.

$$\begin{aligned}Q(s_t, a_t) &= Q_{\theta}(s_t, a_t) \\ \pi(s_t) &= \pi_{\phi}(s_t) = \mathcal{N}_{\phi}(\mu, \sigma^2 | s_t)\end{aligned}\quad (15)$$

where the θ and ϕ indicate the neural network weight of the state-action value function and the policy, respectively; \mathcal{N} is the normal distribution with the mean μ and standard deviation σ , which fits the random gaussian noise to increase exploration in the environment. In addition, different from the optimization objective of the conventional DRL (13), SAC maximizes the cumulative discount reward while maximizing entropy. Thus, the optimal policy of the SAC is defined as follows:

$$\pi_*(s_t) = \operatorname{argmax}_{\pi} E_{(s_t, a_t) \sim p\pi} \left[\sum_{i=0}^{\infty} \nu^i \left(r_{t+i} + \alpha \mathcal{H}(\pi(\cdot | s_{t+i})) \right) \right] \quad (16)$$

where $\mathcal{H}(\pi(\cdot | s_t))$ is the entropy term of the policy π in the current state s_t ; α is the entropy factor, which determines the relative importance of the entropy term. Firstly, according to Lemma 1, the

state-action value function is trained to minimize the following squared residual error:

$$J_Q(\theta) = E_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}} \left[\frac{1}{2} (Q_{\theta}(s_t, a_t) - \hat{Q}(s_t, a_t, s_{t+1}))^2 \right] \quad (17)$$

with

$$\begin{aligned}\hat{Q}(s_t, a_t, s_{t+1}) &= r(s_t, a_t) + \nu E_{a_{t+1} \sim \pi} [Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) \\ &\quad - \alpha \log \pi_{\phi}(a_{t+1} | s_{t+1})]\end{aligned}\quad (18)$$

where \mathcal{D} is the distribution of the sampled states, actions, and next states in the experience buffer; $\bar{\theta}$ is the target network parameters of the state-action value function; $-\log \pi(\cdot | s_t)$ is the concrete entropy term $\mathcal{H}(\pi(\cdot | s_t))$. After the optimization of the state-action value function, based on the equation (16), the policy is trained to maximize the cumulative discount reward and entropy. Therefore, the policy parameters are trained to minimize the residual equation:

$$J_{\pi}(\phi) = E_{s_t \sim \mathcal{D}, a_t \sim \pi_{\phi}} [\alpha \log \pi_{\phi}(a_t | s_t) - Q_{\theta}(s_t, a_t)] \quad (19)$$

where choosing a suitable entropy factor α is vital for the policy. However, it is difficult to set the entropy factor in different tasks. Therefore, instead of the fixed entropy factor, the following loss function is given to adjust the entropy factor automatically.

$$J_{\alpha}(\alpha) = E_{s_t \sim \mathcal{D}, a_t \sim \pi_{\phi}} [-\alpha \log \pi_{\phi}(a_t | s_t) - \alpha \bar{\mathcal{H}}] \quad (20)$$

where $\bar{\mathcal{H}}$ is the target entropy. Moreover, the SAC algorithm applies the two state-action value functions to mitigate positive bias. Thus, the minimum of both state-action functions is used in equation (18) and equation (19). The pseudocode of the SAC algorithm is shown in Appendix B.

3.2.2 | Two-dimensional (2D) DRL Compensator based on the modified SAC algorithm

Compared with the original SAC algorithm, which adopts the target structure only for the state-action value function, we modified the SAC algorithm by incorporating the target structure in the policy. Consequently, the policy target networks improve the sample efficiency and stabilize the training while the computation complexity is slightly increased³⁴. For the modified SAC algorithm, the equation (18) is redefined as:

$$\begin{aligned}\hat{Q}(s_t, a_t, s_{t+1}) &= r(s_t, a_t) + \nu E_{a_{t+1} \sim \pi} [Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) \\ &\quad - \alpha \log \pi_{\bar{\phi}}(a_{t+1} | s_{t+1})]\end{aligned}$$

where $\bar{\phi}$ is the target network parameters of the policy; Based on the modified SAC algorithm, this section illustrates the two-dimensional DRL compensator by designing the proper state space, policy and reward function. The detailed process framework of the 2D DRL compensator is shown in Fig. 4.

• design of the two-dimensional state space

As shown in the equation (1), the unknown non-repetitive nature $f(\cdot, \cdot, \cdot, \cdot)$ are both time-dependent and batch-dependent. Therefore,

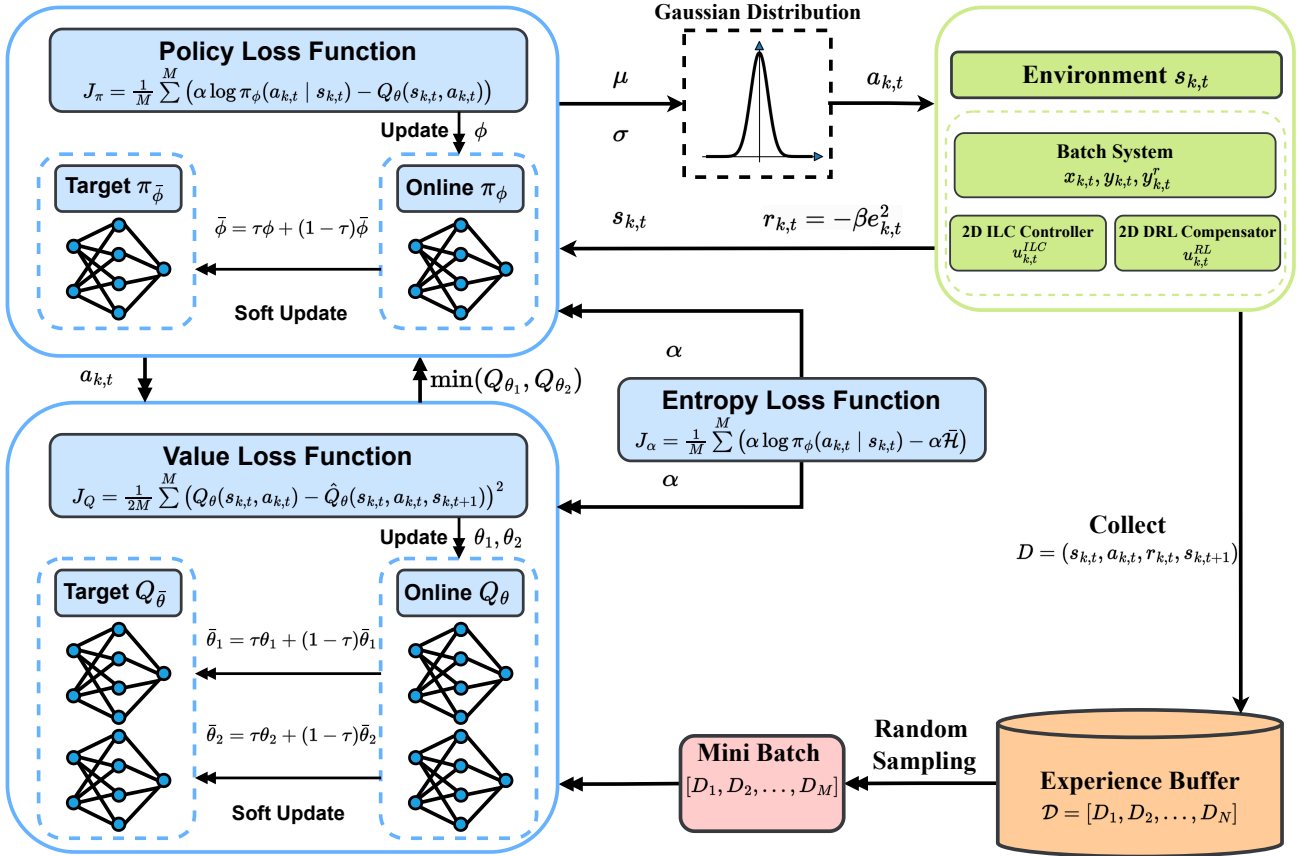


FIGURE 4 The block diagram of the 2D DRL compensator based on the modified SAC algorithm.

not only the time-wise information but also the batch-wise information is utilized to compensate for the negative impact of the system mismatch and non-repetitive nature. More precisely, the state space covers sufficient information about the current batch and the experience of the previous batch, namely a two-dimensional state space (2D state space), which is defined as follows:

$$s_{k,t} = \begin{bmatrix} \cdots & q_{k,t-1} & q_{k,t} \\ \cdots & q_{k-1,t-1} & q_{k-1,t} & q_{k-1,t+1} & \cdots \end{bmatrix} \quad (21)$$

with

$$q_{k,t} \in \{x_{k,t}, y_{k,t}, y_{k,t}^r, u_{k,t}^{ILC}, u_{k,t}^{RL}\} \quad (22)$$

where $q_{k,t}$ indicates any observable information at time t of the k^{th} batch. The first row of $s_{k,t}$ represents the observable state information of the current batch; the second row indicates the experience from the previous batch.

• design of the compensation signal

The compensation signal of the 2D DRL compensator is equivalent to the action of the policy:

$$u_{k,t}^{RL} = \pi_\phi(s_{k,t}) \quad (23)$$

where $u_{k,t}^{RL}$ indicates the compensation control signal, which is given by the policy based on the 2D state space $s_{k,t}$.

• design of the reward function

The reward function in the DRL community is similar to the cost function in control theory. However, the cost function is minimized to get the optimal control signal while maximizing the cumulative reward function to acquire the optimal action. Thus, the instant reward function is designed in reference to the cost function as follows:

$$r_{k,t} = -\beta e_{k,t}^2 \quad (24)$$

where β is the positive coefficient.

3.2.3 | Practical Algorithm of the 2D ILC-RL Control Scheme

When the 2D ILC-RL control scheme, as shown in Fig. 1, is directly applied to the batch systems to achieve the control object, there are some problems to be considered during the training of the 2D DRL compensator. Firstly, in the early training phase, although the 2D ILC controller guarantees basic control performance, the random exploration of the 2D DRL compensator may generate an unsuitable compensation for the

2D ILC controller, worsening control performance. Furthermore, online training on practical batch systems is low-efficiency and cost-expensive. To overcome the mentioned problems, we propose a practical implementation plan, which is divided into three phases.

- **Phase 1: Training in the known nominal system.**

Firstly, the control law K of the 2D ILC controller is calculated according to **Theorem 1**. Based on the known nominal system, the virtually simulated environment is constructed. Finally, the 2D DRL compensator is trained through interaction with the simulated environment until the training objective is achieved. To assess whether the 2D DRL compensator is completely trained, the average reward per batch is used as the training objective.

$$r_k^{av} = \frac{1}{T} \sum_{t=1}^T r_{k,t} \quad (25)$$

- **Phase 2: Training in the practical system.**

After the 2D DRL compensator is trained in the simulated environment, the 2D ILC-RL controller is directly transferred to the actual batch processes (1). The 2D ILC-RL controller controls the batch system, while the 2D DRL compensator is continuously trained without random exploration ($\sigma = 0$ in the equation (15)). The Root Mean Squared Error (RMSE) is utilized as the criterion to assess whether the DRL compensator is completely trained.

$$l_k = \sqrt{\frac{1}{T} \sum_{t=1}^T (y_{k,t} - y_{k,t}^*)^2} \quad (26)$$

If the control performance of the proposed method meets the requirements, the training for the 2D DRL compensator is terminated, and the online policy of the 2D DRL compensator is saved as the final policy.

- **Phase 3: The entirely 2D ILC-RL controller.**

Finally, the designed and trained 2D ILC-RL controller is implemented in the actual batch system.

The detailed process framework of the practical algorithm is illustrated in **Algorithm 2**.

4 | NUMERICAL SIMULATIONS

To demonstrate the effectiveness and the control performance of the proposed 2D ILC-RL control scheme, the linear injection molding process³⁵ and the nonlinear batch reactor³⁶ are considered and simulated in the experimental study.

4.1 | The Linear Injection Molding Process

The injection molding process is the typical batch process in the chemical industry. A numerical injection molding batch process with system

Algorithm 2 Practical 2D ILC-RL Control Scheme

- 1: Design the 2D iterative learning controller based on the scheme in section 3.1.
- 2: Initial all relevant parameters for the 2D DRL compensator.
- 3: Phase 1:
- 4: for maximal training batches N_1 or the average reward per batch $r_{av} > \epsilon_1$ do
- 5: Sample the action:

$$u_{k,t} = u_{k,t}^{ILC} + \mathcal{N}_\phi(\mu, \sigma^2 | s_{k,t}) \quad (27)$$

- 6: Interaction with the virtual environment.
- 7: Optimization of the 2D DRL compensator based on the modified SAC algorithm in section 3.2.
- 8: end for
- 9: Phase 2:
- 10: for maximal training batches N_2 or $l_k < \epsilon_2$ do
- 11: Sample the action:

$$u_{k,t} = u_{k,t}^{ILC} + \mathcal{N}_\phi(\mu, 0 | s_{k,t}) \quad (28)$$

- 12: Interaction with the practical batch system.
 - 13: Optimization of the 2D DRL compensator based on the modified SAC algorithm in section 3.2.
 - 14: end for
 - 15: Phase 3:
 - 16: Finally, the trained 2D ILC-RL controller acts as the final controller of the batch process.
-

model mismatch and non-repetitive nature is considered and modeled by:

$$\begin{cases} x_{k,t+1} = (A + E_1 \Delta_t^1) x_{k,t} + (B + E_2 \Delta_t^2) u_{k,t} + f_{k,t} \\ y_{k,t} = C x_{k,t} \end{cases} \quad (29)$$

with subject to

$$\Delta_t^1 = \begin{bmatrix} \delta_t^1 \\ \delta_t^2 \\ \delta_t^3 \end{bmatrix}, \Delta_t^2 = \delta_t^4, f_{k,t} = E_1 \zeta_k x_{k,t} + E_2 \zeta_k u_{k,t}$$

with

$$A = \begin{bmatrix} 1.607 & -0.608 & -0.928 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1.239 \\ 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$E_1 = \begin{bmatrix} 0.0804 & -0.0304 & -0.0464 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, E_2 = \begin{bmatrix} 0.062 \\ 0 \\ 0 \end{bmatrix}$$

$$\delta_t^i = 0.5 * \sin(0.1t) \text{ for } i \in \{1, 2, 3, 4\}, \zeta_k = 0.5 * \sin(\frac{\pi * k}{5})$$

where A, B, C, E_1, E_2 are the known nominal system matrices; the function $f_{k,t}$ illustrates the unknown iteration-varying dynamics; δ_t^i indicates

the system mismatch. Meanwhile, ζ_k represents the non-repetitive nature. The fixed time duration per batch T is 200. In accordance with the proposed control scheme, firstly, the control law K of the 2D iterative learning controller is calculated by LMI (11) as follows:

$$K = [-1.40838, 0.57543, 0.87757, 0.71898] \quad (30)$$

For the 2D DRL compensator, the 2D state space and the reward function are defined as follows:

$$s_{k,t} = \begin{bmatrix} u_{k,t-1}^{RL} & u_{k,t-1}^{ILC} & y_{k,t-1} & y_{k,t-1}^r & y_{k,t}^r \\ u_{k-1,t}^{RL} & u_{k-1,t}^{ILC} & y_{k-1,t} & y_{k-1,t}^r & u_{k,t}^{ILC} \\ u_{k-1,t-1}^{RL} & u_{k-1,t-1}^{ILC} & y_{k-1,t-1} & y_{k-1,t-1}^r & \end{bmatrix} \quad (31)$$

$$r_{k,t} = -10e_{k,t}^2 \quad (32)$$

The other hyperparameters of the 2D DRL compensator are listed in Table C1.

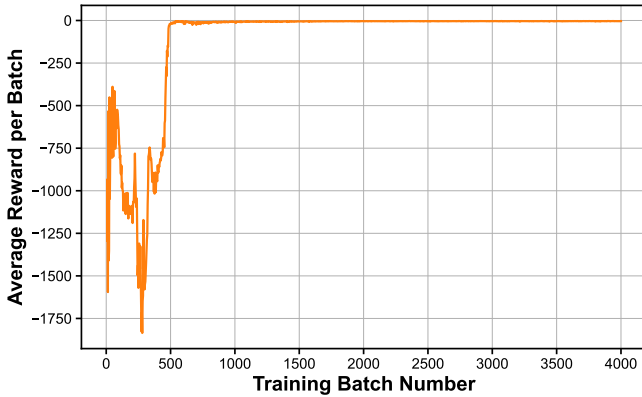


FIGURE 5 The training curve with the nominal virtual environment.

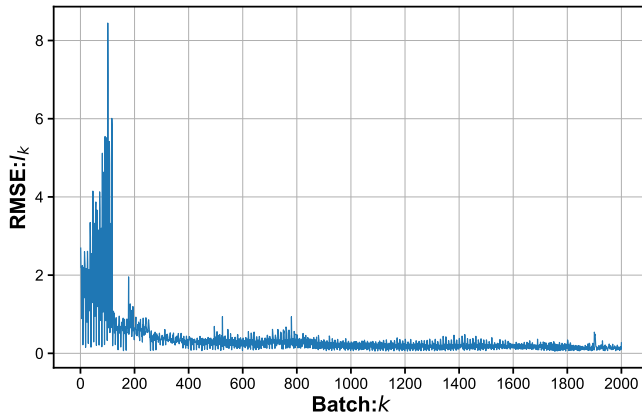


FIGURE 6 The control performance while training with the injection molding process.

In the first phase of the proposed algorithm, the 2D DRL compensator is

trained through exploration in the simulated nominal model of the injection molding system. As shown in Fig. 5, the average reward r_k^{av} gradually increases with the training batch and converges to nearly 0, which indicates that the 2D DRL compensator finds the suitable compensation signal for the nominal model systems. Fig. 6 represents the control performance of the 2D ILC-RL control scheme while training with the natural injection molding system in the second phase. The tracking error I_k decreases with the training batch, demonstrating that the control performance of the system is continuously enhanced through the on-line training of phase 2. Furthermore, the maximal tracking error is only 8.446 (Table 1) in the whole training process with the real system. The small tracking error ensures the safety of the controlled injection molding process to avoid system instability and operation in dangerous areas.

TABLE 1 The maximal RMSE I_k in training with the practical system.

Batch system	Maximal RMSE I_k
Linear Injection Molding Process	8.446
Nonlinear Batch Reactor	5.424

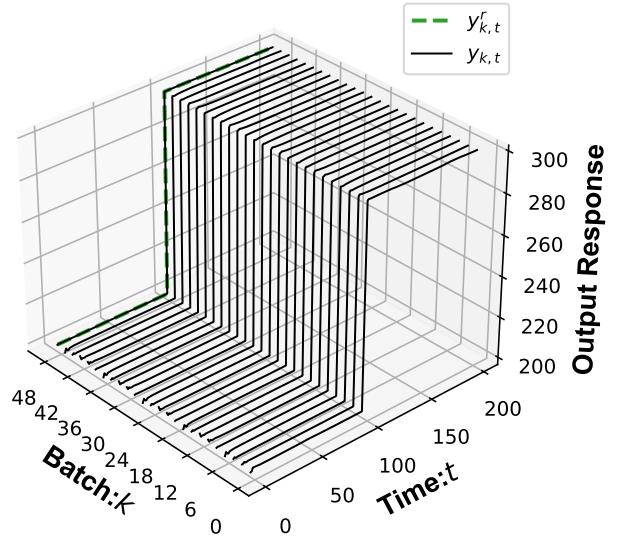


FIGURE 7 The output response of the injection molding process.

Finally, the fully trained 2D ILC-RL controller is applied to the injection molding system (29). Fig. 7 shows the output responses of the injection molding batch process, which demonstrates that the proposed control scheme has good tracking performance. The control signal of the 2D ILC controller and the compensation signal of the 2D DRL compensator are shown in Fig. E1 in Appendix E. To further illustrate the control performance, we compare the RMSE I_k of the proposed control scheme and the sole 2D ILC controller. As shown in Fig. 8, although the convergence of the sole 2D ILC scheme is good, the iteration-wise

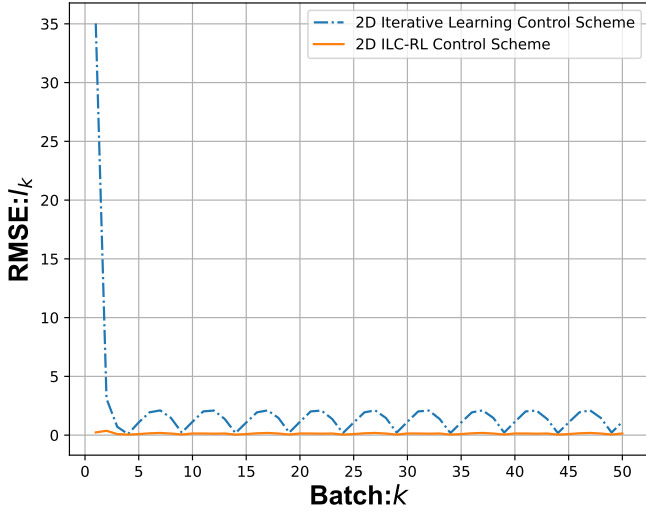


FIGURE 8 The control performance comparison of the pure 2D ILC controller and the 2D ILC-RL control scheme.

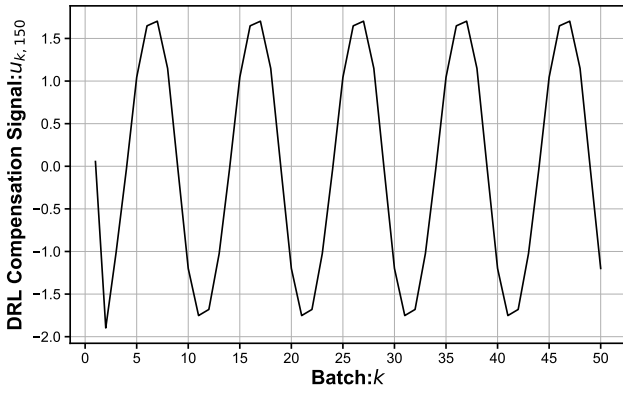


FIGURE 9 The 2D DRL compensation signal at $t = 150$ for all batches.

tracking error caused by the sinusoidal non-repetitive nature $\zeta_k = 0.5 * \sin(\frac{\pi * k}{5})$ is not well suppressed. Nevertheless, as shown in Fig. 9, the 2D DRL compensator of the proposed control scheme gives the non-repetitive compensation signal to restrain the negative impact of the non-repetitive nature. Therefore, the tracking error of the 2D ILC-RL control scheme is smaller than that of the sole 2D ILC scheme. To further accurately compare the control performance of the two schemes, the tracking error of the 47th batch, 49th batch, and the mean value of the tracking error for all batches are illustrated in Table 2. It is seen that the control performance of the 2D ILC-RL control scheme is superior to that of the sole 2D ILC controller. The simulation results illustrate the higher effectiveness and better control performance of the proposed control method for the batch processes with non-repetitive nature and model mismatch.

TABLE 2 The RMSE l_k in 47th batch and 49th batch, and mean value for all batches.

Algorithm	47 th batch	49 th batch	Mean
Sole 2D ILC Scheme	2.1	0.22	2.02
2D ILC-RL Control Scheme	0.15	0.05	0.12

4.2 | The Nonlinear Batch Reactor

The batch reactor is a typical nonlinear batch process that is widely used in the chemical and biochemical industries. The reaction temperature of the batch reactor is precisely controlled to synthesize different polymers. To further demonstrate the applicability of the proposed method, the following nonlinear continuous stirred tank reactor system³⁶ with non-repetitive nature and model mismatch is considered:

$$\begin{cases} \dot{x}_1 = -(\frac{F}{V} + k_0 e^{\frac{-E}{R x_2}})x_1 + \frac{F}{V}C_{A_0} \\ \dot{x}_2 = -\frac{\Delta H k_0}{\rho C_p} e^{\frac{-E}{R x_2}} x_1 - \frac{F}{V}x_2 + \frac{F}{V}(T_{A_0} + \sigma_{k,t}) + \frac{u}{\rho C_p V} \\ y = x_2 \end{cases} \quad (33)$$

with subject to

$$\sigma_{k,t} = T_{A_0} * \sin(0.1\pi t) + T_{A_0} * \sin(\frac{\pi * k}{10}) \quad (34)$$

where x_1 and x_2 are the reactant concentration in the reactor and reactor temperature, respectively; u is the heating or cooling control variables; The first term of $\sigma_{k,t}$ indicates the unknown model mismatch and the second term of $\sigma_{k,t}$ represents the unknown non-repetitive nature. F indicates the flow rate of the reactant; V represents the reactor volume; k_0 is a pre-exponential constant; E and ΔH is the activation energy and the reaction enthalpy, respectively; R is the fixed coefficient; C_{A_0} is the initial reactant concentration; ρ denotes the fluid density; C_p shows the heat capacity; T_{A_0} is the temperature of the fluid; The initial state is $x_{k,0} = [0.5 \ 310]$. The running time per batch is fixed to be 3 minutes with a sampling time of 0.01 and $T = 300$. All parameters' values are listed in Table D3. Firstly, the nonlinear system is linearized and discretized in the equilibrium point $x_{eq} = [0.48632751 \ 350]$, $u_{eq} = 890.6456$ to acquire the following nominal model matrices:

$$A = \begin{bmatrix} 0.9898 & -1.1061 \times 10^{-5} \\ 0.0557 & 0.9923 \end{bmatrix} \quad B = \begin{bmatrix} -2.321 \times 10^{-9} \\ 4.1679 \times 10^{-4} \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (35)$$

The upper boundary of the system uncertainty E_1, E_2, F_1, F_2 are set to 80% of the nominal model matrices:

$$\begin{aligned} E_1 &= \begin{bmatrix} 1 & -1.1 \times 10^{-5} \\ 0.056 & 1 \end{bmatrix} & F_1 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ E_2 &= \begin{bmatrix} -2.32 \times 10^{-9} \\ 4.17 \times 10^{-4} \end{bmatrix} & F_2 &= 1 \end{aligned} \quad (36)$$

Remark 3. The model mismatch of the linearization and the non-repetitive nature modeled by (33) are all considered in the unknown nonlinear system dynamics $f(\cdot, \cdot, \cdot, \cdot)$ of the equation (1).

Based on the practical implementation of the 2D ILC-RL control scheme, the control law K of the 2D iterative learning controller is calculated by LMI (11):

$$K = [1185.5240, -2045.1603, 1017.7915] \quad (37)$$

For the design of the 2D DRL compensator, the 2D state space and the reward function are defined as follows:

$$s_{k,t} = \begin{bmatrix} u_{k,t-1}^{RL} & u_{k,t-1}^{ILC} & y_{k,t-1} & y_{k,t-1}^r & y_{k,t}^r \\ u_{k-1,t}^{RL} & u_{k-1,t}^{ILC} & y_{k-1,t} & y_{k-1,t}^r & u_{k,t}^{ILC} \\ u_{k-1,t-1}^{RL} & u_{k-1,t-1}^{ILC} & y_{k-1,t-1} & y_{k-1,t-1}^r & \end{bmatrix} \quad (38)$$

$$r_{k,t} = -40e_{k,t}^2 \quad (39)$$

The other hyperparameters and the structure of neural networks are given in Table C2.



FIGURE 10 The training curve with the simulated environment.

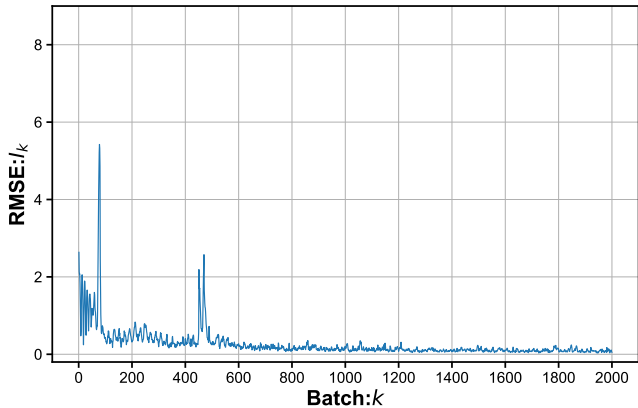


FIGURE 11 The control performance while training with the batch reactor.

Fig. 10 illustrates that the average reward r_k^{av} gradually rises because the 2D DRL compensator understands the system characteristic and

gives an effective compensation signal by exploring the simulated nonlinear batch reactor. After the training in the first phase, the 2D DRL compensator continues to be trained through exploration with the practical nonlinear batch reactor. As indicated in Fig. 11, the tracking error I_k decreases with the batch index, which demonstrates that the control performance of the proposed control scheme is improved. Moreover, the maximal tracking error is less than 5.424 (Table 1), which guarantees the safe operation of the batch reactor in the whole training process.

Finally, the fully trained 2D ILC-RL controller is utilized to control the batch reactor. As shown in Fig. 12, the green dashed line and solid black lines represent the reference trajectory and the output response of the reactor temperature, respectively. It is noted that the output response is highly close to the desired trajectory, which proves that the proposed control scheme has good tracking performance for the nonlinear batch reactor with non-repetitive nature and model mismatch. The control signal of the 2D ILC controller and the compensation signal of the 2D DRL compensator are indicated in Fig.E2 in Appendix E. In addition, the pure 2D ILC controller is the baseline to verify the control performance further. Fig. 13 and Table 3 indicate that the control performance of the proposed method is superior to that of the 2D ILC scheme. Similar to the injection molding process, for the unknown non-repetitive nature, the 2D DRL compensator provides the sinusoidal compensation signal along batch index (Fig. 14) to counteract the negative impact of the sinusoidal non-repetitive nature.

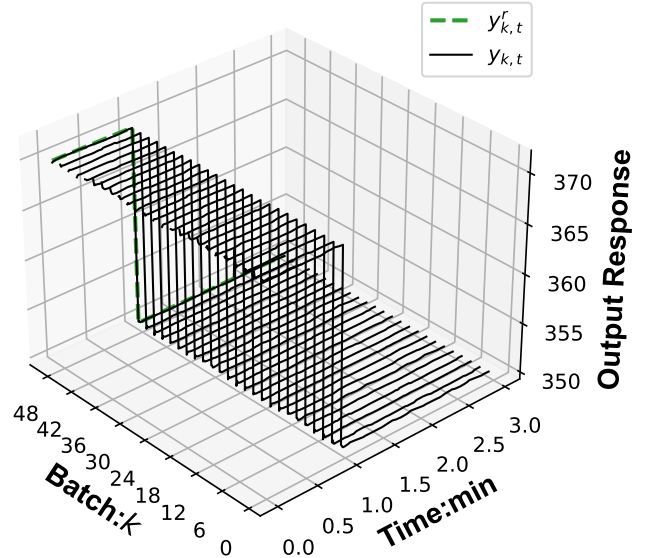


FIGURE 12 The output response of the nonlinear batch reactor.

In summary, the linear injection molding and the nonlinear batch reactor simulation results demonstrate that the proposed 2D ILC-RL control scheme is a universal design framework significantly performing the non-repetitive batch process with system model mismatch. Furthermore, for the batch processes with the time-varying, iteration-varying

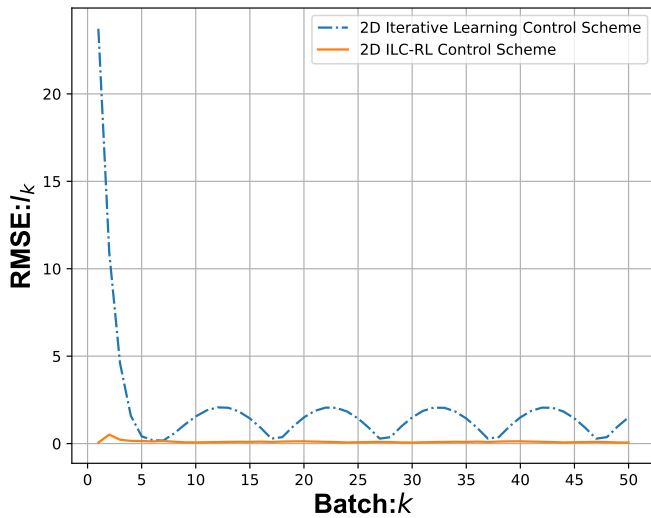


FIGURE 13 The control performance comparison of the pure 2D ILC controller and the 2D ILC-RL control scheme.

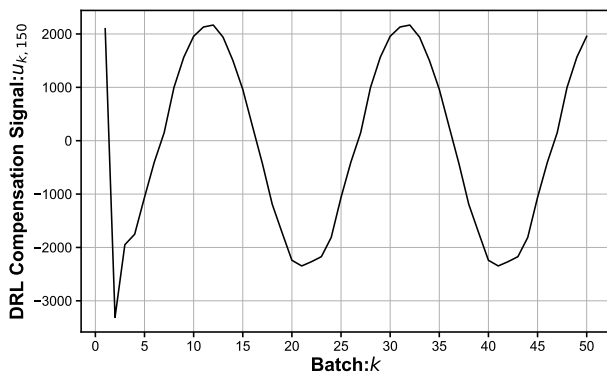


FIGURE 14 The 2D DRL compensation signal at $t = 150$ for all batches.

TABLE 3 The RMSE I_k in 47th batch and 50th batch, and mean value for all batches.

Algorithm	47 th batch	50 th batch	Mean
Sole 2D ILC Scheme	0.28	1.49	1.96
2D ILC-RL Control Scheme	0.09	0.06	0.1

nature, and model mismatch, the combination of the 2D ILC and 2D DRL compensator illustrates the superior control performance than that of the sole 2D ILC scheme.

5 | DATA AVAILABILITY AND REPRODUCIBILITY STATEMENT

The numerical data from Figures 5-9 and 10-14 are tabulated and available as a .zip file in the Supplementary Material. The numerical data

from Figures E1 and E2 in Appendix are available in the Supplementary Material. The Supplementary Material also includes execution files from the trained 2D ILC-RL controller simulation that are described in Figures 7-9 and 12-14. The other data is available on request from the authors.

6 | CONCLUSIONS

For the batch processes with system mismatch and non-repetitive nature, we proposed a novel 2D ILC-RL control scheme in this paper. Based on the 2D system theory, the 2D iterative learning controller is designed to guarantee essential control performance. Meanwhile, the stability and convergence analyses are presented by robust stability theory. Furthermore, the 2D DRL compensator based on the SAC algorithm is constructed and combined with the 2D ILC scheme to improve the entire control performance by counteracting the negative influence of the model mismatch and non-repetitive nature. In addition, to ensure the safety of the online training, we proposed a practical implementation plan based on the known nominal model in which the 2D DRL compensator is first trained in the simulated environment. And then, the trained compensator is continued to be trained in the real controlled system. Finally, the simulation results in the linear injection molding and the nonlinear batch reactor demonstrate the proposed method's superior tracking performance and strong robustness against non-repeatable variation and system mismatch.

AUTHOR CONTRIBUTIONS

Jianan Liu: Conceptualization (lead); data curation (lead); formal analysis (lead); methodology (lead); writing – original draft (lead); writing-review and editing (supporting); **Zike Zhou:** Data curation (equal); writing-review and editing (supporting); **Wenjing Hong:** Funding acquisition (lead); supervision (lead); writing-review and editing (supporting); **Jia Shi:** Funding acquisition (lead); methodology (equal); supervision (lead); writing-review and editing (equal);

ACKNOWLEDGMENTS

We thank the associate editor and Anonymous reviewers.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

ORCID

Jianan Liu <https://orcid.org/0009-0002-7179-0129>

Zike Zhou <https://orcid.org/0009-0005-9644-8444>

Wenjing Hong <https://orcid.org/0000-0003-4080-6175>

Jia Shi <https://orcid.org/0000-0001-5833-7798>

REFERENCES

- Shi J, Gao F, Wu TJ. From two-dimensional linear quadratic optimal control to iterative learning control. Paper 1. Two-dimensional linear quadratic optimal controls and system analysis. *Industrial & engineering chemistry research*. 2006;45(13):4603–4616.

2. Singh V, Kodamana H. Reinforcement learning based control of batch polymerisation processes. *IFAC-PapersOnLine*. 2020;53(1):667–672.
3. De Roover D, Bosgra OH. Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system. *International Journal of Control*. 2000;73(10):968–979.
4. Liu X, Kong X. Nonlinear fuzzy model predictive iterative learning control for drum-type boiler-turbine system. *Journal of Process Control*. 2013;23(8):1023–1040.
5. Pane YP, Nagesh Rao SP, Kober J, Babuška R. Reinforcement learning based compensation methods for robot manipulators. *Engineering Applications of Artificial Intelligence*. 2019;78:236–247.
6. Jiang G, Hou Z. Iterative learning model predictive control approaches for trajectory based aircraft operation with controlled time of arrival. *International Journal of Control, Automation and Systems*. 2020;18(10):2641–2649.
7. Wang L, Freeman CT, Rogers E. Predictive iterative learning control with experimental validation. *Control Engineering Practice*. 2016;53:24–34.
8. Shi J, Wen K, Xu X. Design of Nonlinear Iterative Learning Control Based on Deep Reinforcement Learning Algorithm. In: . 3. IEEE. 2021:722–727.
9. Arimoto S, Kawamura S, Miyazaki F. Bettering operation of robots by learning. *Journal of Robotic systems*. 1984;1(2):123–140.
10. Nagy ZK, al e. Model based robust control approach for batch crystallization product design. *Computers and Chemical Engineering*. 2009;33(10):1685–1691.
11. Chen Y, Moore KL. Harnessing the nonrepetitiveness in iterative learning control. In: . 3. IEEE. 2002:3350–3355.
12. Butcher M, Karimi A. Linear parameter-varying iterative learning control with application to a linear motor system. *IEEE/ASME transactions on mechatronics*. 2009;15(3):412–420.
13. Zhang R, Gao F. Two-dimensional iterative learning model predictive control for batch processes: A new state space model compensation approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2018;51(2):833–841.
14. Liu T, Gao F. Robust two-dimensional iterative learning control for batch processes with state delay and time-varying uncertainties. *Chemical Engineering Science*. 2010;65(23):6134–6144.
15. Yu M, Chai S. Iteration-dependent high-order internal model based iterative learning control for discrete-time nonlinear systems with time-iteration-varying parameter. *IFAC-PapersOnLine*. 2020;53(2):1658–1663.
16. Yu M, Chai S. A survey on high-order internal model based iterative learning control. *IEEE Access*. 2019;7:127024–127031.
17. Yin C, Xu JX, Hou Z. A high-order internal model based iterative learning control scheme for nonlinear systems with time-iteration-varying parameters. *IEEE Transactions on Automatic Control*. 2010;55(11):2665–2670.
18. Zhou W, Yu M, Huang DQ. A high-order internal model based iterative learning control scheme for discrete linear time-varying systems. *International Journal of Automation and Computing*. 2015;12(3):330–336.
19. Tan C, Wang S, Wang J. Robust iterative learning control for iteration-and time-varying disturbance rejection. *International Journal of Systems Science*. 2020;51(3):461–472.
20. Mnih V, Kavukcuoglu K, Silver ea. Human-level control through deep reinforcement learning. *nature*. 2015;518(7540):529–533.
21. Silver D, Huang A, Maddison ea. Mastering the game of Go with deep neural networks and tree search. *nature*. 2016;529(7587):484–489.
22. Zhao X, Xia L, Zhang L, Ding Z, Yin D, Tang J. Deep Reinforcement Learning for Page-Wise Recommendations. In: RecSys '18. Association for Computing Machinery. 2018; New York, NY, USA:95–103.
23. Ma Y, Zhu W, Benton MG, Romagnoli J. Continuous control of a polymerization system with deep reinforcement learning. *Journal of Process Control*. 2019;75:40–47.
24. Joshi T, Makker S, Kodamana H, Kandath H. Twin actor twin delayed deep deterministic policy gradient (TATD3) learning for batch process control. *Computers & Chemical Engineering*. 2021;155:107527.
25. Wang Y, Gao F, Doyle III FJ. Survey on iterative learning control, repetitive control, and run-to-run control. *Journal of Process Control*. 2009;19(10):1589–1600.
26. Kaczorek T. *Two-dimensional linear systems*. Springer, 1985.
27. Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: PMLR. 2018:1861–1870.
28. Sutton RS, Barto AG. *Reinforcement learning: An introduction*. MIT press, 2018.
29. Nian R, Liu J, Huang B. A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*. 2020;139:106886.
30. Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. Trust region policy optimization. In: PMLR. 2015:1889–1897.
31. Liu B, Cai Q, Yang Z, Wang Z. Neural trust region/proximal policy optimization attains globally optimal policy. *Advances in neural information processing systems*. 2019;32.
32. Mnih V, Badia AP, Mirza M, et al. Asynchronous methods for deep reinforcement learning. In: PMLR. 2016:1928–1937.
33. Haarnoja T, Tang H, Abbeel P, Levine S. Reinforcement learning with deep energy-based policies. In: PMLR. 2017:1352–1361.
34. Chou PW, Maturana D, Scherer S. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In: PMLR. 2017:834–843.
35. Gao F, Yang Y, Shao C. Robust iterative learning control with applications to injection molding process. *Chemical Engineering Science*. 2001;56(24):7025–7034.
36. Chi R, Huang B, Hou Z, Jin S. Data-driven high-order terminal iterative learning control with a faster convergence speed. *International Journal of Robust and Nonlinear Control*. 2018;28(1):103–119.
37. Yue D, Han QL. Delay-dependent robust H_∞ controller design for uncertain descriptor systems with time-varying discrete and distributed delays. *IEE Proceedings-Control Theory and Applications*. 2005;152(6):628–638.
38. Boyd S, El Ghaoui L, Feron E, Balakrishnan V. *Linear matrix inequalities in system and control theory*. SIAM, 1994.
39. Shi J, Gao F, Wu TJ. Robust design of integrated feedback and iterative learning control of a batch process based on a 2D Roesser system. *Journal of Process Control*. 2005;15(8):907–924.
40. Du C, Xie L, Zhang C. H_∞ control and robust stabilization of two-dimensional systems in Roesser models. *Automatica*. 2001;37(2):205–211.
41. Lewis FL, Vrabie D, Vamvoudakis KG. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*. 2012;32(6):76–105.

SUPPORTING INFORMATION

Additional supporting information can be found in the online version of the article.

How to cite this article: Jianan Liu, Zike Zhou, Wenjing Hong, and Jia Shi. 2D Iterative Learning Control with Deep Reinforcement Learning Compensation for the Non-repetitive Batch Processes *AIChE J.* 2021;00(00):1–18.

APPENDIX

A PROOF OF THEOREM 1

Definition 1. For the 2D batch system (9) without external disturbance ($w_{k,t} \equiv 0$), the 2D batch system is robust asymptotical stable if the unforced response satisfies the following:

$$\lim_{k,t \rightarrow \infty} \begin{bmatrix} \delta_{k,t} \\ e_{k,t} \end{bmatrix} \rightarrow 0$$

Definition 2. For a given scalar $\gamma > 0$, the 2D batch system has robust H_∞ performance γ if the following conditions are satisfied:

- The 2D batch system is robust asymptotical stable (Definition 1).
- For any external disturbance $w_{k,t}$,

$$\|z\| < \gamma \|w_{k,t}\|$$

Lemma 2.³⁷ X, Y are the matrices or vectors with appropriate dimensions. For any scalar $\varepsilon > 0$, if the matrix Δ satisfies $\Delta \Delta^T \leq I$, the following inequality holds:

$$X \Delta Y + Y^T \Delta^T X^T \leq \varepsilon X X^T + \varepsilon^{-1} Y^T Y \quad (A1)$$

Lemma 3.³⁸ (Schur Complement) S_{11} and S_{22} are positive definite symmetric matrices, and S_{21} is the given appropriate dimensional matrix. The following matrix inequality holds:

$$S_{21}^T S_{22} S_{21} - S_{11} < 0 \quad (A2)$$

if and only if

$$\begin{bmatrix} -S_{11} & S_{21}^T \\ S_{21} & -S_{22}^{-1} \end{bmatrix} < 0 \quad (A3)$$

Lemma 4.³⁹ For any initial boundary condition, the 2D batch system (9) is robust asymptotical stable with the robust convergence index ρ ($0 < \rho < 1$) if the positive block diagonal matrix $P = \text{diag}\{P_\delta, P_e\}$ satisfies the condition:

$$V \left(\begin{bmatrix} \delta_{k,t+1} \\ e_{k,t+1} \end{bmatrix} \right) < \rho V \left(\begin{bmatrix} \delta_{k,t} \\ e_{k-1,t+1} \end{bmatrix} \right) \quad (A4)$$

with

$$V \left(\begin{bmatrix} \delta_{k,t} \\ e_{k,t} \end{bmatrix} \right) = \begin{bmatrix} \delta_{k,t} & e_{k,t} \end{bmatrix}^T P \begin{bmatrix} \delta_{k,t} \\ e_{k,t} \end{bmatrix}$$

Lemma 5.⁴⁰ For a positive scalar γ , the unforced deterministic ($\Delta \bar{A}_t \equiv 0, r_{k,t} \equiv 0$) 2D batch system (9) has robust H_∞ performance γ if the positive block diagonal matrix $P = \text{diag}\{P_\delta, P_e\}$ satisfies the following

matrix inequality:

$$\begin{bmatrix} -P & \bar{A}^T P & \bar{C}^T & 0 \\ P \bar{A} & -P & 0 & P \bar{D} \\ \bar{C} & 0 & -\gamma I & 0 \\ 0 & \bar{D}^T P & 0 & -\gamma I \end{bmatrix} < 0 \quad (A5)$$

Remark 4. Matrix \bar{C} is the measurement matrix of the 2D batch system. To eliminate the sensitivity of the external disturbance, the approximate matrix \bar{C} is chosen as $\bar{C} = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}$.

Proof. Firstly, equation (10) is substituted into equation (9) to form the closed-loop 2D batch system:

$$\begin{bmatrix} \delta_{k,t+1} \\ e_{k,t+1} \end{bmatrix} = (\bar{A} + \bar{B}K + \Delta \bar{A}_t + \Delta \bar{B}_t K) \begin{bmatrix} \delta_{k,t} \\ e_{k-1,t+1} \end{bmatrix} + \bar{D} w_{k,t} \quad (A6)$$

Theorem 2. The closed-loop 2D batch system (A6) is robust asymptotical stable with the robust convergence index ρ ($0 < \rho < 1$) if the positive block diagonal matrix $Q = \text{diag}\{Q_\delta, Q_e\}$ and positive scalars ε, η satisfy the following matrix inequality:

$$\begin{bmatrix} -\rho Q & Q K^T \bar{F}_2^T & Q \bar{F}_1^T & Q \bar{A}^T \\ \bar{F}_2 K Q & -Q + \bar{H} & 0 & 0 \\ \bar{F}_1 Q & 0 & -\varepsilon I & 0 \\ \bar{A} Q & 0 & 0 & -\eta I \end{bmatrix} < 0 \quad (A7)$$

with

$$\bar{A} = \bar{A} + \bar{B}K, \bar{H} = \varepsilon \bar{E}_1 \bar{E}_1^T + \eta \bar{E}_2 \bar{E}_2^T$$

Proof: Theorem 2.

Based on the **Lemma 4**, the closed-loop system (A6) is robust asymptotical stable with the robust convergence index ρ , if the following inequality is given:

$$(\bar{A} + \Delta \bar{A}_t + \Delta \bar{B}_t K)^T P (\bar{A} + \Delta \bar{A}_t + \Delta \bar{B}_t K) - \rho P < 0$$

From the **Lemma 3**, the equation is transformed into:

$$\begin{bmatrix} -\rho P & (\bar{A} + \Delta \bar{A}_t + \Delta \bar{B}_t K)^T \\ \bar{A} + \Delta \bar{A}_t + \Delta \bar{B}_t K & -P^{-1} \end{bmatrix} < 0$$

According to the **Lemma 2**, the left-hand side of the inequality is transformed into:

$$\begin{aligned} & \begin{bmatrix} -\rho P & \bar{A}^T \\ \bar{A} & -P^{-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{E}_1 \end{bmatrix} \Delta_t^A \begin{bmatrix} \bar{F}_1 & 0 \end{bmatrix} + \begin{bmatrix} \bar{F}_1^T \\ 0 \end{bmatrix} (\Delta_t^A)^T \begin{bmatrix} 0 & \bar{E}_1^T \end{bmatrix} \\ & + \begin{bmatrix} 0 \\ \bar{E}_2 \end{bmatrix} \Delta_t^B \begin{bmatrix} \bar{F}_2 & 0 \end{bmatrix} + \begin{bmatrix} \bar{F}_2^T \\ 0 \end{bmatrix} (\Delta_t^B)^T \begin{bmatrix} 0 & \bar{E}_2^T \end{bmatrix} \\ & \leq \begin{bmatrix} -\rho P + \bar{H} & \bar{A}^T \\ \bar{A} & -P^{-1} + \bar{H} \end{bmatrix} < 0 \end{aligned}$$

with

$$\bar{H} = \varepsilon^{-1} \bar{F}_1^T \bar{F}_1 + \eta^{-1} K^T \bar{F}_2^T \bar{F}_2 K$$

By pre-and post-multiplying matrix $\text{diag}\{P^{-1}, I\}$ and setting $Q = P^{-1}$, the inequality is transformed into:

$$\begin{bmatrix} -\rho Q + Q\tilde{H}Q & Q\tilde{A}^T \\ \tilde{A}Q & -Q + \tilde{H} \end{bmatrix} < 0$$

Finally, according to the **Lemma 3**, the inequality (A7) is proved.

Theorem 3. The closed-loop 2D batch system (A6) has robust H_∞ performance γ if the positive block diagonal matrix $Q = \text{diag}\{Q_\delta, Q_e\}$ satisfies the following matrix inequality:

$$\begin{bmatrix} -Q & QK^T\tilde{F}_2^T & Q\tilde{F}_1^T & Q\tilde{A}^T & Q\tilde{C}^T & 0 \\ \tilde{F}_2KQ & -Q + \tilde{H} & 0 & 0 & 0 & 0 \\ \tilde{F}_1Q & 0 & -\epsilon I & 0 & 0 & 0 \\ \tilde{A}Q & 0 & 0 & -\eta I & 0 & \tilde{D} \\ \tilde{C}Q & 0 & 0 & 0 & -\gamma I & 0 \\ 0 & 0 & 0 & \tilde{D}^T & 0 & -\gamma I \end{bmatrix} < 0 \quad (\text{A8})$$

Proof: Theorem 3.

Based on the **Lemma 5**, the closed-loop 2D batch system (A6) has robust H_∞ performance γ if the positive block diagonal matrix $P = \text{diag}\{P_\delta, P_e\}$ satisfies the following matrix inequality:

$$\begin{bmatrix} -P & (\tilde{A} + \Delta\tilde{A}_t + \Delta\tilde{B}_tK)^T P & \tilde{C}^T & 0 \\ P(\tilde{A} + \Delta\tilde{A}_t + \Delta\tilde{B}_tK) & -P & 0 & P\tilde{D} \\ \tilde{C} & 0 & -\gamma I & 0 \\ 0 & \tilde{D}^T P & 0 & -\gamma I \end{bmatrix} < 0$$

Similar to the proof of **Theorem 2**, based on the **Lemma 2**, the above inequality is equivalent to:

$$\begin{bmatrix} -P + \tilde{H} & \tilde{A}^T P & \tilde{C}^T & 0 \\ P\tilde{A} & -P + P\tilde{H}P & 0 & P\tilde{D} \\ \tilde{C} & 0 & -\gamma I & 0 \\ 0 & \tilde{D}^T P & 0 & -\gamma I \end{bmatrix} < 0$$

By pre-and post-multiplying matrix $\text{diag}\{P^{-1}, P^{-1}, I, I\}$ and setting $Q = P^{-1}$, the above inequality is transformed into:

$$\begin{bmatrix} -Q + Q\tilde{H}Q & Q\tilde{A}^T & Q\tilde{C}^T & 0 \\ \tilde{A}Q & -Q + \tilde{H} & 0 & \tilde{D} \\ \tilde{C}Q & 0 & -\gamma I & 0 \\ 0 & \tilde{D}^T & 0 & -\gamma I \end{bmatrix} < 0 \quad (\text{A9})$$

Finally, according to the **Lemma 3**, the inequality (A8) is given.

Proof: Theorem 1.

Obviously, if the linear matrix inequality in **Theorem 1** is held, **Theorem 2** and **Theorem 3** are satisfied. In addition, to achieve the strong H_∞ performance γ , the minimal γ is calculated. Finally, **Theorem 1** is proved.

B THE SOFT ACTOR-CRITIC ALGORITHM

Algorithm 1 SAC algorithm

- 1: Initialize the online value function $Q_{\theta_i}(s, a)$ and the target value function $Q_{\bar{\theta}_i}(s, a)$ with the same parameters $\theta_i = \bar{\theta}_i$ for $i \in \{1, 2\}$.
- 2: Initialize the online policy $\pi_\phi(s)$ and the target policy $\pi_{\bar{\phi}}(s)$ with the same parameters $\phi = \bar{\phi}$.
- 3: Reset experience buffer \mathcal{D} with maximal data N .
- 4: Initialize the entropy factor α , the discount factor ν , and the soft factor τ , and the target entropy \bar{H} .
- 5: Reset the different learning rates λ_Q , λ_π , and λ_α .
- 6: for each episode do
- 7: Initialize the environment and get initial state s_0 .
- 8: for each environment step i do
- 9: Sample the action from the policy:

$$a_i = \pi_\phi(s_i) = \mathcal{N}_\phi(\mu, \sigma^2 | s_i) \quad (\text{B10})$$

- 10: Execute action a_i , and observe s_{i+1} , and r_i .
- 11: Store the transition (s_i, a_i, r_i, s_{i+1}) in the buffer \mathcal{D} .
- 12: end for
- 13: for each gradient step do
- 14: Sample a random mini-batch M transitions from the experience buffer \mathcal{D} .
- 15: Update the parameters of the online value function based on the defined loss function (17):

$$\theta_i = \theta_i - \lambda_Q \nabla_{\theta_i} J_Q(\theta_i) \text{ for } i \in \{1, 2\} \quad (\text{B11})$$

- 16: Update the parameters of the online policy based on the defined loss function (19):

$$\phi = \phi - \lambda_\pi \nabla_\phi J_\pi(\phi) \quad (\text{B12})$$

- 17: Update the entropy factor based on the defined loss function (20):

$$\alpha = \alpha - \lambda_\alpha \nabla_\alpha J_\alpha(\alpha) \quad (\text{B13})$$

- 18: Update the target function and target policy:

$$\begin{cases} \bar{\theta}_i = \tau\theta_i + (1 - \tau)\bar{\theta}_i & \text{for } i \in \{1, 2\} \\ \bar{\phi} = \tau\phi + (1 - \tau)\bar{\phi} \end{cases} \quad (\text{B14})$$

- 19: end for

- 20: end for

C THE HYPERPARAMETERS FOR THE 2D DRL COMPENSATOR

TABLE C1 The Hyperparameters for the Linear Injection Molding Process.

Parameter	Value
optimizer	Adam
learning rate λ_Q , λ_π , and λ_α	$3 \cdot 10^{-4}$
number of hidden layers	3
capacity of experience buffer N	$4 \cdot 10^5$
number of hidden units per layer	256
number of samples per mini-batch M	512
nonlinear function	ReLU
gradient steps	600
target update interval	1
the initial entropy factor α	1
the target entropy $\bar{\pi}$	0
the discount factor ν	0.99
the soft factor τ	0.005
the maximal training batches in Phase 1 N_1	4000
the maximal training batches in Phase 2 N_2	2000
the maximal average reward per batch ϵ_1	-3
the minimal RMSE ϵ_2	0.001

D THE NONLINEAR CONTINUOUSLY STIRRED TANKS REACTOR SYSTEM

TABLE D3 The parameter's value of the nonlinear continuously stirred tanks reactor system.

Parameter	Value
the flow rate of the reactant F	$0.1 \text{ m}^3/\text{min}$
the reactor volume V	0.1 m^3
the pre-exponential constant k_0	$72 \times 10^9 \text{ min}^{-1}$
the activation energy E	$8.314 \times 10^4 \text{ kJ/kmol}$
the reaction enthalpy ΔH	$-4.78 \times 10^4 \text{ kJ/kmol}$
the initial concentration C_{A_0}	1 kmol/m^3
the fluid density ρ	1000 kg/m^3
the heat capacity C_p	$0.239 \text{ kJ/kg} \cdot \text{K}$
the temperature of the fluid T_{A_0}	310 K
the coefficient R	$8.314 \text{ kJ/kmol} \cdot \text{K}$

TABLE C2 The Hyperparameters for the Nonlinear Batch Reactor.

Parameter	Value
optimizer	Adam
learning rate λ_Q , λ_π , and λ_α	$3 \cdot 10^{-4}$
number of hidden layers	3
capacity of experience buffer N	$6 \cdot 10^5$
number of hidden units per layer	256
number of samples per mini-batch M	512
nonlinear function	ReLU
gradient steps	3000
target update interval	1
the initial entropy factor α	1
the target entropy $\bar{\pi}$	0
the discount factor ν	0.99
the soft factor τ	0.005
the maximal training batches in Phase 1 N_1	10000
the maximal training batches in Phase 2 N_2	2000
the maximal average reward per batch ϵ_1	-5
the minimal RMSE ϵ_2	0.001

E THE INPUT SIGNAL OF THE 2D ILC-RL CONTROL SCHEME

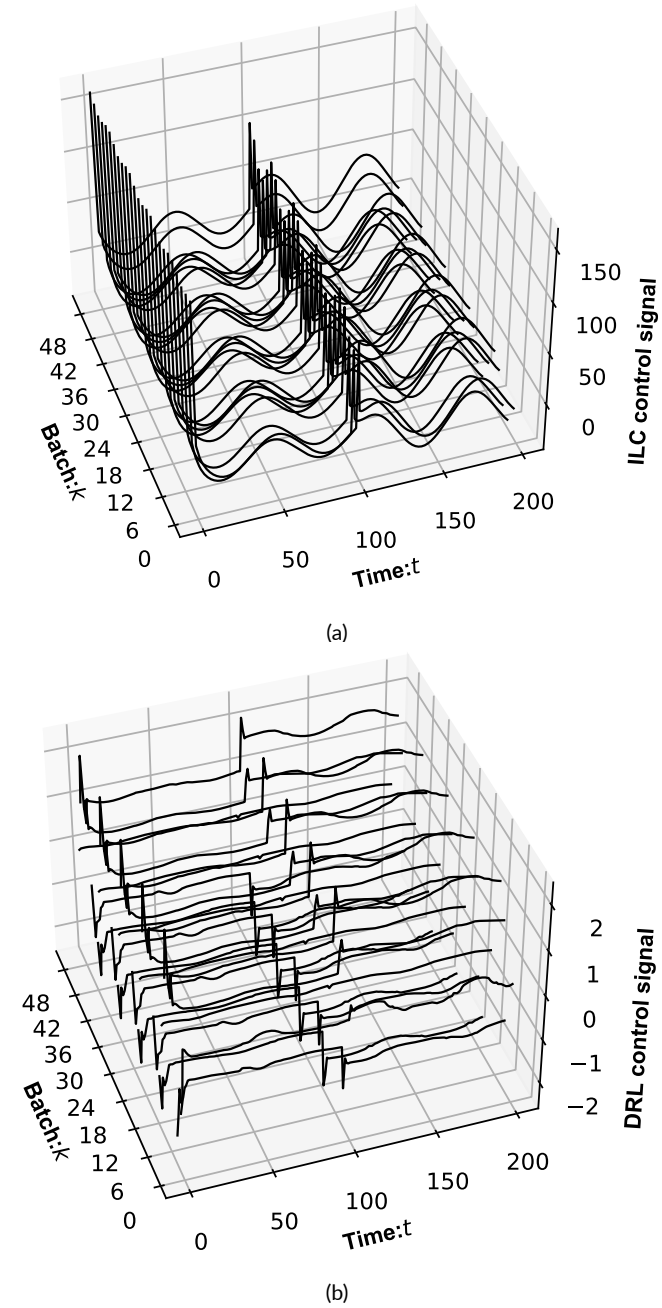


FIGURE E1 The input signal for the linear injection molding process: (a) ILC control signal. (b) DRL compensation signal.

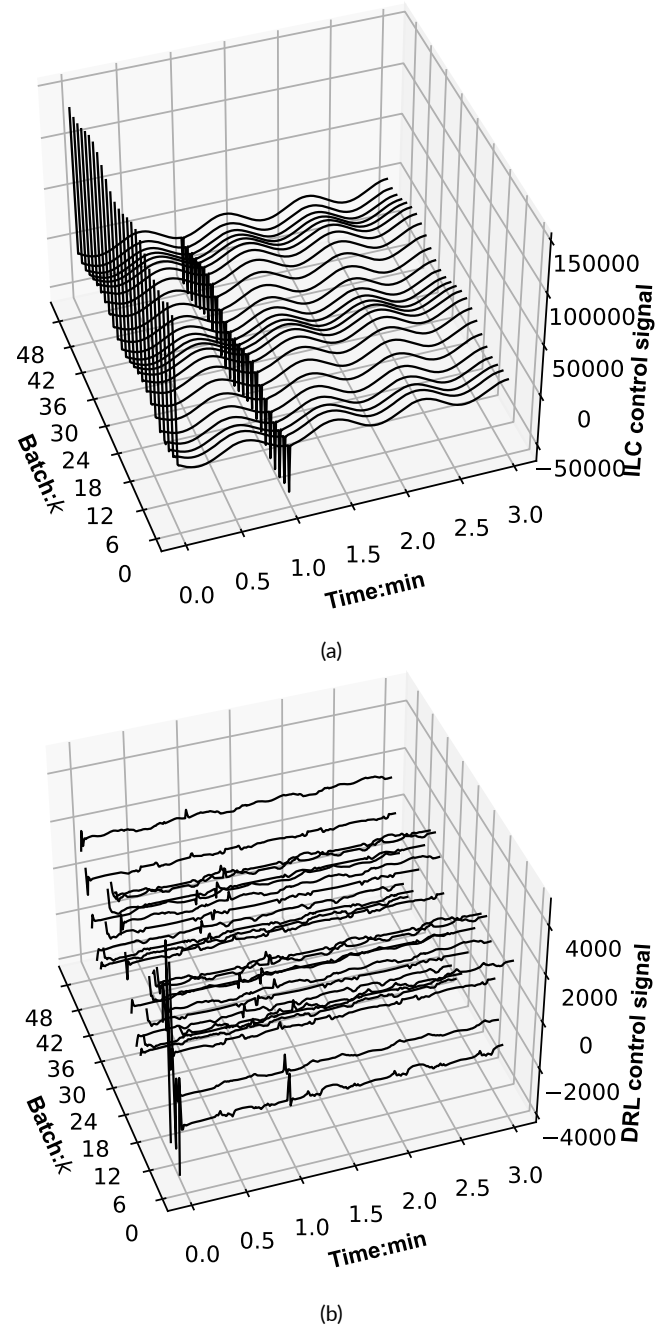


FIGURE E2 The input signal for the nonlinear batch reactor: (a) ILC control signal. (b) DRL compensation signal.