





Data-Driven Learning Control Algorithms for Unachievable Tracking Problems

Zeyi Zhang , Hao Jiang , Dong Shen , *Senior Member, IEEE*, and Samer S. Saab , *Senior Member, IEEE*

Abstract—For unachievable tracking problems, where the system output cannot precisely track a given reference, achieving the best possible approximation for the reference trajectory becomes the objective. This study aims to investigate solutions using the P-type learning control scheme. Initially, we demonstrate the necessity of gradient information for achieving the best approximation. Subsequently, we propose an input-output-driven learning gain design to handle the imprecise gradients of a class of uncertain systems. However, it is discovered that the desired performance may not be attainable when faced with incomplete information. To address this issue, an extended iterative learning control scheme is introduced. In this scheme, the tracking errors are modified through output data sampling, which incorporates low-memory footprints and offers flexibility in learning gain design. The input sequence is shown to converge towards the desired input, resulting in an output that is closest to the given reference in the least square sense. Numerical simulations are provided to validate the theoretical findings.

Index Terms—Data-driven algorithms, incomplete information, iterative learning control, gradient information, unachievable problems.

I. INTRODUCTION

ITERATIVE learning control (ILC) is a powerful approach for addressing tracking problems in repetitive processes [1], [2]. It leverages historical information obtained during repeated iterations to improve performance in subsequent iterations [3], similar to how humans learn through experience. In ILC, an iteration refers to a system operating over a finite time interval, and input and output data from previous iterations are used to compute the input for the current iteration [4], [5]. Among various ILC schemes, the P-type learning control algorithm has been widely studied and applied due to its simple structure, where the tracking error is linearly mapped to the input space using a learning gain matrix [6]. However, designing the learning gain matrix requires knowledge of the

system, which can be challenging when the system model is unknown or uncertain and the available data is incomplete.

Many studies have highlighted the challenges posed by unknown/uncertain system models and incomplete data in the context of ILC [7]–[12]. Robust and data-driven control methods have been developed to address these issues. Incomplete data scenarios often involve data dropouts [13], [14], communication delays [15], [16], and varying operation lengths [17], [18]. In such scenarios, the unit does not receive all the information, but instead, it obtains input and output fragments at certain time instants and dimensions. The incompleteness is typically modeled using a 0-1 distribution to indicate whether the data is obtained.

However, the P-type learning control algorithm can achieve high-precision tracking performance despite the challenges posed by unknown/uncertain models and incomplete data [19]. This is predicated on the assumption that the reference trajectory is realizable [20]–[22], meaning that there exists an input that can produce an output equal to the reference trajectory over the entire operation interval. Achieving perfect tracking, in this case, requires zero initial tracking error at each iteration [23], although precise initialization is often difficult to guarantee in practice. Various techniques have been developed to relax this condition [24], [25].

However, what happens when ILC encounters an unrealizable reference trajectory? An unrealizable reference trajectory refers to a situation where the tracking errors cannot be simultaneously reduced to zero for all time instants, regardless of the input used. This problem frequently arises in underactuated systems, where the dimension of the inputs is less than that of the output [26]. In such cases, neither ILC nor other control methods can ensure that the system output matches the desired reference. This is known as the unachievable tracking problem, which has received limited attention in previous ILC studies.

One approach to address the unachievable tracking problem is to minimize the accumulated error index over the entire operation interval. In an earlier work [27], the authors proposed a P-type ILC method to minimize the Euclidean norm of tracking errors, leveraging the system model to achieve the best possible tracking performance. However, the information necessary to achieve optimal performance remains unspecified. A preliminary work [28] addressed the unachievable tracking problem under random data dropouts; however, it required system information, and the learning control scheme was limited to specific inputs. Thus, a more rigorous formulation is needed to tackle the unachievable tracking problem.

Manuscript received June 4, 2023; revised June 26, 2023; accepted July 6, 2023. This work was supported by the National Natural Science Foundation of China (62173333, 12271522), Beijing Natural Science Foundation (Z210 002), and the Research Fund of Renmin University of China (2021030187). Recommended by Associate Editor Nian Zhang. (*Corresponding author: Dong Shen.*)

Citation: Z. Zhang, H. Jiang, D. Shen, and S. Saab, “Data-driven learning control algorithms for unachievable tracking problems,” *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 1, pp. 205–218, Jan. 2024.

Z. Zhang, H. Jiang, and D. Shen are with the School of Mathematics, Renmin University of China, Beijing 100872, China (e-mail: zhzy@ruc.edu.cn; jiangh@ruc.edu.cn; dshen@ruc.edu.cn).

S. Saab is with the School of Engineering, Lebanese American University, Byblos 2038, Lebanon (e-mail: ssaab@lau.edu.lb).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2023.123756

For the achievable tracking problem, ILC can ensure zero-error tracking performance over the entire time interval, and most existing ILC studies have focused on this problem. However, for the unachievable tracking problem, it is inherently impossible to achieve perfect zero-error tracking performance for a given reference over the entire time interval, which is jointly determined by the system model and the desired reference. Hence, it is not merely a technical control issue but rather an inherent property. In this case, the goal is to approximate the desired reference as closely as possible in a certain sense. Motivated by this idea, we use an optimization objective to model the best achievable tracking performance, describe the necessary information to solve the optimization objective, and propose data-driven ILC algorithms for practical applications. In summary, our work aims to address the following issues related to the unachievable tracking problem:

1) Can the widely used P-type learning control scheme solve the optimization objective, and what information is necessary for the design?

2) How can a data-driven P-type learning control scheme be established to handle uncertain/unknown system models?

3) What is the essential effect of incomplete tracking data, and how can this difficulty be overcome?

To tackle these problems, this study makes the following contributions:

1) We characterize the concept of the unachievable tracking problem and demonstrate the necessity of the true gradient information in achieving the control objective using the conventional P-type ILC (Proposition 2 and Claim 1).

2) To handle uncertain/unknown system models, we design a new learning gain matrix using an input and output (I/O) sampling strategy (Algorithm 1). This strategy extracts sufficient gradient information from the available input and output data and achieves the optimization objective in the mean square sense without relying on the system matrix (Theorem 1).

3) Incomplete input and output data can affect the learning process and lead to drift in the gradient information (Proposition 3). To overcome this challenge, we propose an extended ILC scheme that combines the sampling strategy with an error compensation strategy (Algorithm 2). The extended scheme incorporates error correction and achieves convergence of the input error to zero in the mean square sense, thereby achieving the optimization objective (Theorem 2).

Given the inherent difficulties introduced by the unachievable tracking problem, this paper presents two novel data-driven schemes of the P-type ILC that aim to achieve an optimization objective. In comparison to [27], we clarify the necessity of gradient information and highlight the impact of incomplete data on the convergence of the unachievable problem, which does not occur when the reference is realizable. Previous methods such as [9], [10], [29], [30] identify a dynamic system model, whereas our proposed sampling scheme constructs the learning gain matrix directly from samples, eliminating the need for system identification. The compensation method in [11] accelerates robust ILC by compensating for the input, and [12] presents a real-time data-based compensation method to address external disturbances. In

contrast, our second scheme aims to compensate for the tracking error of the unrealizable reference. Compared to the earlier attempt in [28], this study defines the unachievable tracking problem, reveals the necessity of gradient information for learning, and proposes novel data-driven learning control algorithms. Furthermore, the algorithm in [28] can be seen as a special case of Algorithm 2 presented in this study. Overall, the existing ILC literature rarely addresses the unachievable tracking problem, and this work represents the first comprehensive investigation of this topic.

Organization: Section II presents the problem formulation. Section III demonstrates the necessity of the gradient. Section IV proposes an ILC scheme with a sampling strategy. Section V elaborates the effect of incomplete data, which then is overcome in Section VI by proposing an extended ILC scheme. In Section VII, numerical simulations verify the theoretical analysis. Section VIII concludes the paper.

Notations: $\mathbb{E}[\cdot]$ denotes the mathematical expectation of a random variable. $\text{Range}(M)$ and $\text{Null}(M)$ denote the column space and null spaces of a matrix M , respectively. S^\perp is the complementary orthogonal space of S . $\|\cdot\|$ and $\|\cdot\|_F$ denote the 2-norm and the Frobenius norm of a matrix, respectively.

II. PROBLEM FORMULATION

Consider the discrete time-varying linear system,

$$\begin{aligned} x_k(t+1) &= A_t x_k(t) + B_t u_k(t) \\ y_k(t) &= C_t x_k(t) \end{aligned} \quad (1)$$

where t denotes time instants, $t = 0, 1, \dots, N$, and k denotes the iteration label. $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}^p$ and $y_k(t) \in \mathbb{R}^q$ denote the system state, input, and output, respectively. $A_t \in \mathbb{R}^{n \times n}$, $B_t \in \mathbb{R}^{n \times p}$, and $C_t \in \mathbb{R}^{q \times n}$ are the system matrices with appropriate dimensions.

The desired reference for tracking is denoted by $y_r(t)$. The objective is to generate an input signal such that the output $y_k(t)$ tracks the reference $y_r(t)$ well.

The following assumptions are given for analysis.

Assumption 1: The input-output coupling matrix $C_{t+1}B_t$ is of full column rank $\forall t = 0, \dots, N-1$.

Remark 1: Assumption 1 is necessary to ensure the uniqueness of the input solution based on the minimum performance index. Relaxing the requirement of the full column rank does not affect the essence of the analysis but leads to more complex derivations. Additionally, Assumption 1 does not consider the case of full row rank because the latter implies that any given reference can be perfectly tracked.

Assumption 2: The initial state $x_k(0)$ is reset to an unknown invariant value x_0 for all iterations.

Remark 2: To emphasize our main contributions, Assumption 2 is used as an initialization condition. This condition is more practical than the commonly used identical initialization condition, where the initial state is required to be reset to the desired state $x_r(0)$ satisfying $y_r(0) = C_0 x_r(0)$. The latter condition ensures that the desired reference is achievable, as defined below. Assumption 2 only requires the initial state to be reset to a fixed point for each iteration, which is easy to implement in various applications.

Generally, learning tracking problems are classified into two

categories: achievable and unachievable tracking problems.

1) The *achievable tracking problem* indicates that the desired reference $y_r(t)$ can be precisely tracked by the system output, i.e., a $u_r(t)$ (unnecessarily unique) exists such that

$$\begin{aligned} x_r(t+1) &= A_t x_r(t) + B_t u_r(t) \\ y_r(t) &= C_t x_r(t) \end{aligned} \quad (2)$$

where the initial state $x_r(0)$ satisfies $y_r(0) = C_0 x_r(0)$. Here, the desired reference $y_r(t)$ is called *realizable*. The tracking problem is widely investigated in the existing literature, where the existence and uniqueness of $u_r(t)$ are usually required as an assumption. The convergence of the output sequence to the desired reference can be verified by showing that the input sequence converges to $u_r(t)$ for all time instants.

2) The *unachievable tracking problem* refers to the case that the desired reference $y_r(t)$ is unrealizable, i.e., no input exists such that (2) is satisfied. In this case, the system cannot achieve zero-error tracking simultaneously for all time instants. Instead, the control objective becomes to approximate the desired reference as closely as possible in a certain sense.

For the unachievable tracking problem, this study considers the tracking performance index,

$$\mathcal{J} = \sum_{t=1}^N \|y(t) - y_r(t)\|^2 \quad (3)$$

where $y(t)$ is output generated by the system (1). Our objective is to iteratively generate an input sequence converging to a limit that minimizes the index \mathcal{J} by designing suitable ILC algorithms without knowing the system information.

Remark 3: The tracking performance at the initial time instant, i.e., $\|y_k(0) - y_r(0)\|^2$, is not included in the index (3) for the following reasons: First, the output at the initial time instant $y_k(0)$ is solely determined by the initial state $x_k(0)$ and cannot be improved by any input signal due to the system's relative degree being one, based on Assumption 1. For systems with a higher relative degree, the performance index could be modified accordingly, but the subsequent derivations remain valid with slight modifications. Therefore, adding this term does not affect finding the minimum index. If the initial state is available, the tracking reference for the remaining time instants can be adjusted accordingly, such that the performance index adequately determines the learning control problem. Several initial state learning mechanisms proposed in [31] can be combined with the learning control. However, investigating these mechanisms is beyond the scope of this study.

We employ the lifting formulation of the system dynamics [1], integrate the time domain dynamics into super-vectors, and highlight the iteration dynamics, thus making the derivations uncomplicated. The super-vectors are defined as follows:

$$\begin{aligned} Y_k &= [y_k^T(1), y_k^T(2), \dots, y_k^T(N)]^T \in \mathbb{R}^{qN} \\ U_k &= [u_k^T(0), u_k^T(1), \dots, u_k^T(N-1)]^T \in \mathbb{R}^{pN} \\ Y_r &= [y_r^T(1), y_r^T(2), \dots, y_r^T(N)]^T \in \mathbb{R}^{qN}. \end{aligned}$$

Then, the system matrix \mathbf{G} is given by

$$\begin{bmatrix} C_1 B_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ C_2 A_1 B_0 & C_2 B_1 & \mathbf{0} & \cdots & \mathbf{0} \\ C_3 A_1^2 B_0 & C_3 A_2 B_1 & C_3 B_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_N A_1^{N-1} B_0 & C_N A_2^{N-1} B_1 & C_N A_3^{N-1} B_2 & \cdots & C_N B_{N-1} \end{bmatrix}$$

where $A_j^i \triangleq A_i A_{i-1} \cdots A_j$, $i \geq j$. As a result, the system (1) is identical to

$$Y_k = \mathbf{G} U_k + \mathbf{M} x_0 \quad (4)$$

where $\mathbf{M} = [(C_1 A_0)^T, (C_2 A_0^1)^T, \dots, (C_N A_0^{N-1})^T]^T$.

Consequently, the tracking performance index (3) becomes

$$\mathcal{J} = \|Y - Y_r\|^2 \quad \text{s.t.} \quad Y = \mathbf{G} U + \mathbf{M} x_0. \quad (5)$$

Here, Y and U are vectors with the same dimensions as Y_k and U_k , respectively. From Assumption 1, \mathbf{G} is of full column rank; therefore, minimizing (5) leads to the unique desired input, denoted by $U_d = [u_d^T(0), u_d^T(1), \dots, u_d^T(N-1)]^T$. Theoretically, the desired input is

$$U_d = \arg \min_{U \in \mathbb{R}^{Np}} \|\mathbf{G} U + \mathbf{M} x_0 - Y_r\|^2. \quad (6)$$

Replacing the input U_k in (4) with the desired input U_d , we obtain the *best achievable reference* Y_d , given by $Y_d = \mathbf{G} U_d + \mathbf{M} x_0$. Hence, the control objective can be summarized as finding U_d through iterative learning of the input using input/output data. Moreover, the vector $Y_d - \mathbf{M} x_0 \in \text{Range}(\mathbf{G})$ corresponds to the unique vector closest to $Y_r - \mathbf{M} x_0$ in the least square sense. Consequently, $Y_r - Y_d$ is orthogonal to $\text{Range}(\mathbf{G})$, and the minimum performance index (5) is obtained as follows:

$$\mathcal{J}_{\min} = \|Y_r - Y_d\|^2. \quad (7)$$

For the sake of brevity, we denote $\bar{Y}_r = Y_r - \mathbf{M} x_0$ and $\bar{Y}_d = Y_d - \mathbf{M} x_0$ as the *modified reference* and *modified best achievable reference*, respectively.

Next, we introduce the conventional P-type ILC algorithm. Given an arbitrary initial input U_0 , the update law is defined as follows:

$$U_{k+1} = U_k - \alpha \mathbf{L} E_k \quad (8)$$

where $E_k \triangleq Y_k - Y_r$ represents the tracking error in the lifting form, $\alpha > 0$ is the step size, and $\mathbf{L} = \text{diag}\{\mathbf{L}_0, \mathbf{L}_1, \dots, \mathbf{L}_{N-1}\}$ is the learning gain matrix with $\mathbf{L}_l \in \mathbb{R}^{p \times q}$ as diagonal blocks.

Remark 4: The conventional P-type ILC algorithm, known for its simplicity, has been widely utilized in the literature. The learning gain matrix \mathbf{L} plays a crucial role in determining the update direction during the learning process. Furthermore, the block diagonal structure of the matrix \mathbf{L} allows for independent implementation of the algorithm (8) at each time instant.

To describe the control objective for the unachievable tracking problem, we formulate an optimization problem. In the subsequent sections, we will elaborate on how the P-type ILC algorithm achieves the control objective by addressing the following aspects:

- 1) We demonstrate the necessity of precise gradient for the algorithm (8) to solve the problem (3) (see Section III).
- 2) We propose a data-driven sampling strategy to obtain an

alternative gradient and analyze the convergence of the associated ILC algorithm (see Section IV).

3) We delve into the profound impact of incomplete data on the P-type ILC (see Section V).

4) We design an extended ILC scheme to mitigate the effects of incomplete data (see Section VI).

III. NECESSITY OF GRADIENT FOR BEST TRACKING PERFORMANCE

Since it is evident from (8) that $U_k \in U_0 + \text{Range}(\mathbf{L})$ by the recursion, the learning gain matrix \mathbf{L} determines the space for the input renewal process. Hence, to ensure effective learning, $\rho(\mathbf{I} - \alpha\mathbf{L}\mathbf{G}) < 1$ is widely required in ILC [1], where $\rho(\cdot)$ denotes the spectral radius. Based on this condition, we can obtain an in-depth understanding of the tracking ability of the P-type ILC scheme for the unachievable tracking problem.

Proposition 1: If the spectral radius of $\mathbf{I} - \alpha\mathbf{L}\mathbf{G}$ is less than one, then $\mathbb{R}^{qN} = \text{Range}(\mathbf{G}) \oplus \text{Null}(\mathbf{L})$.

Proof: First, $\mathbf{L}\mathbf{G}$ should be of full rank; otherwise, $\mathbf{I} - \alpha\mathbf{L}\mathbf{G}$ must have an eigenvalue equal to one. Thus, \mathbf{L} is of full row rank, implying that $\dim \text{Null}(\mathbf{L}) = qN - pN$, and then $\dim \text{Range}(\mathbf{G}) + \dim \text{Null}(\mathbf{L}) = qN$. It is sufficient to show $\text{Range}(\mathbf{G}) \cap \text{Null}(\mathbf{L}) = \{\mathbf{0}\}$. Supposing that a nonzero vector $y \neq \mathbf{0}$ exists, such as $y \in \text{Range}(\mathbf{G}) \cap \text{Null}(\mathbf{L})$, a vector $x \neq \mathbf{0}$ can be produced such that $y = \mathbf{G}x$, resulting in $\mathbf{L}\mathbf{G}x = \mathbf{L}y = \mathbf{0}$. This finding contradicts the full rank property of $\mathbf{L}\mathbf{G}$. ■

According to the above proposition, we decompose the modified reference \bar{Y}_r into two parts,

$$\bar{Y}_r = Y_G + Y_L \quad (9)$$

where $Y_G \in \text{Range}(\mathbf{G})$ and $Y_L \in \text{Null}(\mathbf{L})$ are unique as per Proposition 1. Subsequently, a unique input U_G , satisfying $Y_G = \mathbf{G}U_G$, is produced because \mathbf{G} is of full column rank; theoretically, $U_G = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T Y_G$.

Subtracting U_G from both sides of (8) leads to the expression shown below:

$$\begin{aligned} U_{k+1} - U_G &= U_k - U_G - \alpha\mathbf{L}(\mathbf{G}U_k + \mathbf{M}x_0 - Y_r) \\ &= U_k - U_G - \alpha\mathbf{L}(\mathbf{G}U_k - Y_G - Y_L) \\ &= (\mathbf{I} - \alpha\mathbf{L}\mathbf{G})(U_k - U_G) \end{aligned}$$

where $\mathbf{L}Y_L = \mathbf{0}$ is applied. Based on the conventional contraction mapping principle, it is evident that $U_k - U_G \rightarrow \mathbf{0}$ if the spectral radius of $(\mathbf{I} - \alpha\mathbf{L}\mathbf{G})$ is less than one.

Remark 5: For the achievable tracking problem, $Y_L = \mathbf{0}$ naturally holds. Thus, for any \mathbf{L} satisfying $\rho(\mathbf{I} - \alpha\mathbf{L}\mathbf{G}) < 1$, U_k has a unique limit U_G meaning that $\mathbf{G}U_k$ converges to $Y_G = \bar{Y}_r = Y_r - \mathbf{M}x_0$, i.e., $Y_k \rightarrow Y_r$. Therefore, for the achievable tracking problem, \mathbf{L} has a relatively wide design range implying that the reference can be perfectly tracked even if the model is uncertain. However, for the unachievable tracking problem, different \mathbf{L} would lead to different $Y_L (\neq \mathbf{0})$ and then to different convergence limits Y_G .

Proposition 2: Consider the system (1). Let Assumptions 1 and 2 hold, all eigenvalues of $L_t C_{t+1} B_t$ be positive real numbers for all time instants, and α satisfy $0 < \alpha < 2/\rho(\mathbf{L}\mathbf{G})$. Then, for any desired reference Y_r , the input sequence $\{U_k\}$ generated by (8) converges to $U_G = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T Y_G$, where Y_G is determined by the decomposition (9).

Proof: Note that \mathbf{G} is a block lower triangular matrix with $C_{t+1} B_t$ being its diagonal block. By Assumption 1, \mathbf{G} is of full column rank; hence, there must exist \mathbf{L} such that $\mathbf{L}\mathbf{G}$ has positive eigenvalues. Then, $\mathbf{L}\mathbf{G}$ is nonsingular. Hence, the selection of $0 < \alpha < 2/\rho(\mathbf{L}\mathbf{G})$ ensures that the spectral radius of $\mathbf{I} - \alpha\mathbf{L}\mathbf{G}$ is less than one. ■

From Proposition 2, the system output Y_k converges to $Y_G + \mathbf{M}x_0$, where Y_G is uniquely determined by the reference Y_r according to Proposition 1 and the direct sum decomposition (9). However, we emphasize that the limit $Y_G + \mathbf{M}x_0$ does not necessarily achieve the minimum of the tracking performance index (5). Indeed, the tracking performance using the learning gain matrix \mathbf{L} is characterized by

$$\mathcal{J}_L = \|Y_G + \mathbf{M}x_0 - Y_r\|^2 = \|Y_L\|^2 \geq \mathcal{J}_{\min}.$$

This is because Y_L is never learned since the updated route is restricted by the learning gain matrix \mathbf{L} . Mathematically speaking, the linear space $\text{Null}(\mathbf{L})$ containing Y_L is not naturally identical to $\text{Range}^\perp(\mathbf{G})$. Therefore, $\mathcal{J}_L = \mathcal{J}_{\min}$ if and only if $\text{Null}(\mathbf{L}) = \text{Range}^\perp(\mathbf{G})$. This observation leads to the following claim.

Claim 1: A necessary condition for minimizing (5) is that all columns of \mathbf{L}^T belong to the space spanned by \mathbf{G} .

If we know the precise system matrix \mathbf{G} , a simple choice for \mathbf{L} is $\mathbf{L} = \mathbf{G}^T$, which leads to the standard gradient algorithm,

$$U_{k+1} = U_k - \alpha \mathbf{G}^T E_k.$$

This algorithm generates an input sequence that converges to U_d defined by (6) [27]. Hence, \mathbf{G}^T is referred to as the standard gradient. A more general design that satisfies Claim 1 is $\mathbf{L} = \mathbf{\Lambda} \mathbf{G}^T$, where $\mathbf{\Lambda}$ is a nonsingular matrix such that all eigenvalues of $\mathbf{\Lambda} \mathbf{G}^T \mathbf{G}$ are positive. Such an \mathbf{L} is not equal to the standard gradient \mathbf{G}^T , but it contains the necessary gradient information to achieve the control objective.

Remark 6: The necessity of the gradient information does not imply that the full system matrix \mathbf{G} must be known for the design. For example, if $\text{rank}(C_{t+1} B_t) = \text{rank}(C_{t+1})$, it is evident that the selection of $L_t = (C_{t+1} B_t)^T$ and $\mathbf{L} = \text{diag}(L_0, L_1, \dots, L_{N-1})$ satisfies Claim 1. This selection does not require full system information. A sampling strategy is proposed in Section IV to generate a sufficient learning gain matrix \mathbf{L} .

A typical design for providing the necessary gradient information is to choose $\mathbf{\Lambda}$ as a symmetric positive-definite (SPD) matrix. In this case, $\mathbf{\Lambda} \mathbf{G}^T$ is considered a subgradient. We collectively refer to \mathbf{G}^T and $\mathbf{\Lambda} \mathbf{G}^T$ as the gradient.

Overall, Claim 1 demonstrates the necessity of the gradient in determining \mathbf{L} to minimize the unlearned part \mathcal{J}_L and achieve the best approximation. However, the system matrix \mathbf{G} is often unknown or uncertain in many applications. Consequently, the associated learning gain matrix \mathbf{L} may be incorrect, defective, or unavailable. To address this problem, the next section presents a sampling strategy.

IV. ILC WITH SAMPLING STRATEGY

This section addresses learning control for the deterministic system (4) without any data incompleteness. We use prior-sampled I/O data to acquire system information and resolve the unachievable tracking problem without system information. In particular, these data are used to form the randomized

learning gain matrix, avoiding the direct use of the system matrix; thus, this mechanism is data-driven.

Definition 1 (Sample): A sample refers to a pair of I/O data $\{U_k, Y_k\}$ generated by the system (4).

Section III discloses that the learning gain matrix must detect the gradient to ensure the best tracking performance. It is difficult to regulate the control direction without system matrix \mathbf{G} . From a mathematical perspective, the gradient algorithm requires the row space of the learning gain matrix to be equal to $\text{Range}(\mathbf{G})$. Fortunately, the difference between outputs naturally belongs to the space $\text{Range}(\mathbf{G})$, which motivates us to employ the sampled I/O data reflecting the system information to regulate the learning control direction. To this end, we set a group of $\{U_i^\circ\}_{i=0,1,\dots,l}$ as the sample inputs. These inputs generate the outputs $\{Y_i^\circ\}_{i=0,1,\dots,l}$ as per the dynamics (4). Subsequently, we compute $Z_i = Y_i^\circ - Y_0^\circ$ and $X_i = U_i^\circ - U_0^\circ$ for $i \in \mathcal{I}$, where $\mathcal{I} \triangleq \{1, 2, \dots, l\}$ is the index set. Consequently, the sampling set $\{(X_i, Z_i)\}_{i \in \mathcal{I}}$ satisfies $Z_{\mathcal{I}}^T = X_{\mathcal{I}}^T \mathbf{G}^T$ for any subset $\mathcal{I}_s = \{i_1, \dots, i_{l_s}\} \subset \mathcal{I}$, where $Z_{\mathcal{I}_s} = [Z_{i_1}, \dots, Z_{i_{l_s}}]$ and $X_{\mathcal{I}_s} = [X_{i_1}, \dots, X_{i_{l_s}}]$ are matrices constructed by column vectors. Therefore, we obtain an implicit gradient to design the learning gain matrix. An ILC scheme with the sampling strategy is presented in Algorithm 1.

Algorithm 1 ILC With a Random Sampling Strategy

```

1: Determine  $\{(X_i, Z_i)\}_{i \in \mathcal{I}}$  and division  $\mathcal{D} = \{\mathcal{I}_1, \dots, \mathcal{I}_\tau\}$ .
2: Initialize arbitrary  $U_0, k = 0$ .
3: while  $k \leq K$  do
4:   Select  $\mathcal{I}_s$  with probability  $p_s$ , construct  $Z_{\mathcal{I}_s}$  and  $X_{\mathcal{I}_s}$ .
5:    $U_{k+1} = U_k - \alpha X_{\mathcal{I}_s} Z_{\mathcal{I}_s}^T E_k$ .
6:    $k = k + 1$ .
7: end while
8: return  $U_k$ 

```

Remark 7: The predetermined division \mathcal{D} is a set of subsets of \mathcal{I} . We use \mathcal{D} just to make the algorithm more general for various practical cases. Hence, \mathcal{D} does not have to be explicitly generated and stored in practice. The sampling index set $\mathcal{I}_s \in \mathcal{D}$ is selected with a probability $p_s > 0$, $s = 1, \dots, \tau$ during each iteration. Then, for the full utilization of all samples, we require $\mathcal{I} = \cup_s \mathcal{I}_s$. In this way, the division is ergodic over $\{(X_i, Z_i)\}_{i \in \mathcal{I}}$. Consequently, the division can be arbitrary considering computational capacity and memory size. A special case is the use of all samples in each iteration, indicating that the gain matrix is $X_{\mathcal{I}} Z_{\mathcal{I}}^T$, which is deterministic. Another special case is that the gain matrix $X_i Z_i^T$ uses only one data pair with low-memory footprints.

Since the deterministic P-type form is obtained by taking mathematical expectations to the recursion, the input updates defined in Step 5 in Algorithm 1 can be interpreted as a randomized version of the conventional P-type learning control algorithm (8), which results in the following expression:

$$\begin{aligned} \mathbb{E}[U_{k+1}] &= \mathbb{E}[U_k] - \alpha \mathbb{E}[X_{\mathcal{I}_s} Z_{\mathcal{I}_s}^T E_k] \\ &= \mathbb{E}[U_k] - \alpha \mathbb{E}[X_{\mathcal{I}_s} Z_{\mathcal{I}_s}^T] (\mathbf{G} \mathbb{E}[U_k] - \bar{Y}_r). \end{aligned}$$

Note that $\mathbb{E}[X_{\mathcal{I}_s} Z_{\mathcal{I}_s}^T] = \mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T] \mathbf{G}^T$ satisfies the gradient requirement, i.e., $\text{Range}(\mathbf{G} \mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T]) \subset \text{Range}(\mathbf{G})$ in Claim 1.

Moreover, this selection leads to the following Lemma 1.

Lemma 1: If $\{X_i\}_{i \in \mathcal{I}}$ contains a basis of the linear space \mathbb{R}^{pN} and $\mathcal{I} = \cup_s \mathcal{I}_s$, each sampling subset \mathcal{I}_s can be selected with a positive probability, then $\mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T]$ is SPD.

Proof: Note that $X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T$ is symmetric semi-positive definite and $X_{\mathcal{I}} X_{\mathcal{I}}^T$ is SPD. $\forall v \in \mathbb{R}^{pN}$, then $\mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T] v = \sum_{s=1}^\tau p_s X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T v = 0 \Leftrightarrow X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T v = 0, \forall s \Leftrightarrow X_i X_i^T v = 0, \forall i \in \mathcal{I} \Leftrightarrow X_{\mathcal{I}} X_{\mathcal{I}}^T v = 0$. Thus, $\text{Null}(\mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T]) = \text{Null}(X_{\mathcal{I}} X_{\mathcal{I}}^T)$, and then $\mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T]$ is SPD. ■

Lemma 2 ([32]): Suppose that M_1 and M_2 are real SPD matrices. Then, all eigenvalues of $M_1 M_2$ are positive real numbers.

Remark 8: In a single iteration, $X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T \mathbf{G}^T$ may not yield a sufficient gradient since $X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T$ may be singular. However, it provides an unbiased estimation of the gradient, and Algorithm 1 obtains the gradient in a randomized manner. From a matrix perspective, the more columns $X_{\mathcal{I}_s}$ contains, the more likely non-singular $X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T$ is, resulting in a more accurate gradient estimation. From an estimation perspective, a larger l_s in $X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T \mathbf{G}^T = \sum_{i=1}^{l_s} X_i X_i^T \mathbf{G}^T$ corresponds to more samples and leads to a more accurate estimation of the gradient. In summary, using more samples benefits the learning process by providing a more accurate estimation of the gradient, but it also involves more computational effort.

We now present the convergence of Algorithm 1 for the system (1).

Theorem 1: Consider system (1) with Assumptions 1 and 2 holding. Let α be a sufficiently small number and $\mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T]$ be nonsingular. Then, for any unachievable reference Y_r , the input U_k generated by Algorithm 1 converges to the desired U_d in the mean square sense.

Proof: Substituting $Z_{\mathcal{I}_s} = \mathbf{G} X_{\mathcal{I}_s}$ into Step 5 in Algorithm 1 and denoting $\delta_k = U_k - U_d$, we have

$$\begin{aligned} \delta_{k+1} &= U_k - U_d - \alpha X_{\mathcal{I}_s} (\mathbf{G} X_{\mathcal{I}_s})^T E_k \\ &= \delta_k - \alpha X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T \mathbf{G}^T (\mathbf{G} U_k - \mathbf{G} U_d + \bar{Y}_d - \bar{Y}_r) \\ &= \delta_k - \alpha X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T \mathbf{G}^T \mathbf{G} \delta_k \end{aligned} \quad (10)$$

where the orthogonality of $\bar{Y}_d - \bar{Y}_r$ and $\text{Range}(\mathbf{G})$ is applied. For the SPD matrices $\mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T]$ and $\mathbf{G}^T \mathbf{G}$, all eigenvalues of the product $\mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T] \mathbf{G}^T \mathbf{G}$ are positive real numbers by Lemma 2. Based on the Lyapunov stability theory, a unique SPD matrix \mathbf{Q} satisfying $\mathbf{Q} \mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T] \mathbf{G}^T \mathbf{G} + \mathbf{G}^T \mathbf{G} \mathbb{E}[X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T] \mathbf{Q} = \mathbf{I}_{pN}$ is obtained. Next, we introduce a weighted norm of δ_{k+1} with respect to \mathbf{Q} : $\|\delta_{k+1}\|_{\mathbf{Q}}^2 \triangleq \delta_{k+1}^T \mathbf{Q} \delta_{k+1}$. By substituting (10) into the weighted norm, the expression below is obtained:

$$\begin{aligned} \|\delta_{k+1}\|_{\mathbf{Q}}^2 &= \|\delta_k\|_{\mathbf{Q}}^2 + \alpha^2 \|X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T \mathbf{G}^T \mathbf{G} \delta_k\|_{\mathbf{Q}}^2 \\ &\quad - \alpha \delta_k^T (\mathbf{Q} X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T \mathbf{G}^T \mathbf{G} + \mathbf{G}^T \mathbf{G} X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T \mathbf{Q}) \delta_k. \end{aligned}$$

Taking mathematical expectation on this equality, we have

$$\mathbb{E}\|\delta_{k+1}\|_{\mathbf{Q}}^2 = \mathbb{E}\|\delta_k\|_{\mathbf{Q}}^2 - \alpha \mathbb{E}\|\delta_k\|^2 + \alpha^2 \mathbb{E}\|X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T \mathbf{G}^T \mathbf{G} \delta_k\|_{\mathbf{Q}}^2.$$

There exist suitable constants c_1 and c_2 satisfying that $\mathbb{E}\|X_{\mathcal{I}_s} X_{\mathcal{I}_s}^T \mathbf{G}^T \mathbf{G} \delta_k\|_{\mathbf{Q}}^2 \leq c_1 \mathbb{E}\|\delta_k\|_{\mathbf{Q}}^2$ and $\mathbb{E}\|\delta_k\|^2 \geq c_2 \mathbb{E}\|\delta_k\|_{\mathbf{Q}}^2$. Thus,

$\mathbb{E}\|\delta_{k+1}\|_Q^2 \leq (1 + \alpha^2 c_1 - \alpha c_2) \mathbb{E}\|\delta_k\|_Q^2$. If α is sufficiently small such that $\alpha < c_2/c_1$, then $0 < 1 + \alpha^2 c_1 - \alpha c_2 < 1$. Consequently, $\mathbb{E}\|\delta_k\|_Q^2$ must converge to zero by the contraction principle. ■

Theorem 1 reveals that the random sampling strategy can probe the inherent gradient, removing the requirement of exact system model matrices for the conventional P-type ILC algorithms. Because the random sampling strategy is a data-driven strategy without providing an estimate of \mathbf{G} , it differs from the estimation-based methodology. Nevertheless, it offers an explicit candidate for the learning gain matrix \mathbf{L} , differing from the conventional P-type ILC algorithms which give a design condition of \mathbf{L} .

Remark 9: The condition of the matrix $\mathbb{E}[X_{I_s} X_{I_s}^T]$ being nonsingular holds if and only if $\{X_i\}_{i \in \mathcal{I}}$ contains a basis of the linear space \mathbb{R}^{pN} and an ergodic sampling over the basis (see Lemma 1). For example, $\{X_i\}_{i \in \mathcal{I}}$ traverses all columns of the identity matrix \mathbf{I}_{pN} . Notably, it is easy to ensure this condition by selecting suitable inputs.

Sections III and IV clarify the necessity of gradients for unachievable tracking and how to obtain them without an exact model. In the next section, we will investigate how these findings are undermined if the received data is randomly incomplete.

V. GRADIENT DRIFT BY INCOMPLETE DATA

This section shows that algorithms in the form (8) cannot achieve the best tracking performance in incomplete data scenarios even if a precise system matrix \mathbf{G} is employed.

To this end, we first present a general model of the incomplete data environment. In this study, we use $\Gamma_k E_k$ to represent the available error information for the input updating. Data incompleteness is modeled by a random matrix $\Gamma_k \in \mathbb{R}^{qN \times qN}$ multiplying the corresponding data vector, where each entry of Γ_k is subject to 0-1 distribution: the entry being equal to 1 if the associated information is available and 0 if the associated information is lost. Therefore, Γ_k is not directly available to the controller but is indicated by the obtained data. For further analysis, we assume that Γ_k is independent of the iteration k and $\bar{\Gamma} \triangleq \mathbb{E}[\Gamma_k]$ is nonsingular. The following incomplete data scenarios are covered by this model.

1) *Random Data Dropouts:* The outputs are randomly lost during the information exchange through an unreliable communication channel. We compensate for the output with the desired reference signal if the output is lost. Then, a variable $\gamma_k(t)$ is employed to denote the transmission effect on the output at time instant t , where $\gamma_k(t) = 1$ and $\gamma_k(t) = 0$ correspond to successful transmission and data dropout, respectively. Finally, $\Gamma_k = \text{diag}\{\gamma_k(1), \dots, \gamma_k(N)\} \otimes \mathbf{I}_q$ is defined to denote the virtual error for the learning control algorithm by $\Gamma_k E_k$.

2) *Randomly Varying Lengths:* The operation process for each iteration randomly ends before the desired length N . The iteration length is modeled by N_k , randomly varying in the candidate set $\{\underline{N}, \underline{N} + 1, \dots, N\}$. In this scenario, we also use $\gamma_k(t)$ to denote the occurrence of the output at time instant t . Differing from the previous scenario, the random variables $\gamma_k(t)$ are not mutually dependent, resulting in $\Gamma_k = \text{diag}\{\mathbf{I}_{N_k}, \mathbf{0}_{N-N_k}\} \otimes \mathbf{I}_q$.

Remark 10: Bernoulli random variables are most popular in

modeling various randomness such as data dropouts, varying trial lengths, and communication delays [33]. These are covered in our setting where Γ_k is independent regarding the iteration number. More complex settings such as the Markov chain and random sequence model can be considered; however, these models are omitted in this study to highlight the main novelty.

We provide a proposition to show the convergence of the P-type ILC algorithm (8) employing the precise gradient $\mathbf{L} = \mathbf{G}^T$ under incomplete data. In Section III, $\mathbf{L} = \mathbf{G}^T$ guarantees that $\lim_{k \rightarrow \infty} U_k = U_d$ if the received data are complete. However, this conclusion may fail to hold if the received tracking error information is incomplete.

Proposition 3: Consider the system (1). Let Assumptions 1 and 2 hold, $\mathbf{L} = \mathbf{G}^T$, and α be smaller than the inverse of the maximum eigenvalue of $\mathbf{G}^T \mathbf{G}$. Applying the P-type learning control algorithm (8) using the incomplete data $\Gamma_k E_k$, the update process becomes

$$U_{k+1} = U_k - \alpha \mathbf{G}^T \Gamma_k E_k. \quad (11)$$

Then, U_k converges to the drifted input U_{drift} in the expectation sense, where U_{drift} is specified in the proof.

Proof: Taking expectation on both sides of (11), the following expression is obtained:

$$\begin{aligned} \mathbb{E}[U_{k+1}] &= \mathbb{E}[U_k] - \alpha \mathbf{G}^T \mathbb{E}[\Gamma_k] (\mathbf{G} \mathbb{E}[U_k] - \bar{Y}_r) \\ &= \mathbb{E}[U_k] - \alpha \mathbf{G}^T \bar{\Gamma} (\mathbf{G} \mathbb{E}[U_k] - \bar{Y}_r). \end{aligned} \quad (12)$$

This recursion can be treated similarly to Proposition 2, where $\mathbf{G}^T \bar{\Gamma}$ corresponds to \mathbf{L} . Therefore, the sequence of $\mathbb{E}[U_k]$ converges to a limit denoted by U_{drift} and determined by an alternative decomposition of \bar{Y}_r . Particularly, $\mathbb{R}^{qN} = \text{Range}(\mathbf{G}) \oplus \text{Null}(\mathbf{G}^T \bar{\Gamma})$ results in the unique decomposition: $\bar{Y}_r = Y_{\text{drift}} + Y_{\text{drift}}^+$, where $Y_{\text{drift}} \in \text{Range}(\mathbf{G})$ and $Y_{\text{drift}}^+ \in \text{Null}(\mathbf{G}^T \bar{\Gamma})$. As a result, a unique U_{drift} satisfying $Y_{\text{drift}} = \mathbf{G} U_{\text{drift}}$ is produced. ■

Several observations are obtained from Proposition 3. First, Proposition 3 reveals that the convergence limit of the generated input sequence may drift from the desired input due to the data incompleteness even though the standard gradient $\mathbf{L} = \mathbf{G}^T$ is used. We refer to this phenomenon as *limit drift*. That is, the convergence limit is drifted by incomplete data. Indeed, the limit of the generated input sequence is jointly determined by the learning gain matrix and the received data. Moreover, the tracking performance under incomplete data environments is characterized by $\mathcal{J}_{\text{drift}} = \|\bar{Y}_r - Y_{\text{drift}}\|^2 = \|Y_{\text{drift}}^+\|^2$ in the expectation sense. Note that $\text{Null}(\mathbf{G}^T \bar{\Gamma})$ is non-identical to $\text{Null}(\mathbf{G}^T)$ unless $\bar{\Gamma}$ is a scalar matrix. Therefore, $\mathcal{J}_{\text{drift}} \geq \mathcal{J}_{\min}$, and the equality holds if and only if $\bar{\Gamma}$ is a scalar matrix. Therefore, even if \mathbf{G} is precisely known, the best tracking performance \mathcal{J}_{\min} is difficult to achieve.

If $\bar{\Gamma}$ has a decomposition $\bar{\Gamma} = \mathbf{W}^T \mathbf{W}$, the recursion (12) can be rewritten as follows:

$$\mathbb{E}[U_{k+1}] = \mathbb{E}[U_k] - \alpha (\mathbf{W} \mathbf{G})^T (\mathbf{W} \mathbf{G} \mathbb{E}[U_k] - \mathbf{W} \bar{Y}_r). \quad (13)$$

Thus, the limit of $\mathbb{E}U_k$ solves the optimization problem,

$$\min_{U \in \mathbb{R}^{Np}} \|\mathbf{W} \mathbf{G} U - \mathbf{W} \bar{Y}_r\|^2 = \min_{U \in \mathbb{R}^{Np}} \|\mathbf{G} U - \bar{Y}_r\|_{\bar{\Gamma}}^2 \quad (14)$$

which is a weighted optimization problem depending on $\bar{\Gamma}$.

This result provides an alternative understanding of the effect of incomplete data. That is, solving the problem (14) requires the gradient to be $\mathbf{G}^T \bar{\Gamma}$; however, the gradient required for minimizing \mathcal{J} in (3) is \mathbf{G}^T . Thus, in an expectation view, the limit drift is caused by the drift of the gradient from \mathbf{G}^T to $\mathbf{G}^T \bar{\Gamma}$. In other words, the gradient for the learning process has drifted because of incomplete data. We refer to this phenomenon as *gradient drift* due to incomplete data.

Remark 11: When the tracking problem is achievable, i.e., $\mathbf{G}\mathbf{U} = \bar{\mathbf{Y}}_r$ holds, the solution of (14) is the unique \mathbf{U}_d , which is not affected by $\bar{\Gamma}$. This verifies that the P-type ILC is robust to incomplete data for the achievable tracking problem.

Next, we provide an example to show that the best tracking performance (7) cannot be achieved by the P-type learning control algorithm (8) without a precise gradient.

Example 1: First, we consider the system (4) and algorithm (8) with $\mathbf{x}_0 = \mathbf{0}$, where \mathbf{G} and \mathbf{L} are given by

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and $\mathbf{Y}_r = [1, 0, 1, 0, 1, 0, 1, 0, 1]^T$. Notably, the minimum of the performance index is $\mathcal{J}_{\min} = 1.5811$ achieved by $\mathbf{U}_d = [1, 0.5, -1, 0, 1, -0.5]^T$, and the best achievable reference is $\mathbf{Y}_d = [1, 0.5, 0, 0, 0.5, 0, 1, 0, 0]^T$. Using algorithm (8) with the imprecise gradient \mathbf{L} and $\mathbf{U}_0 = \mathbf{0}$, limits are identified as $\lim_{k \rightarrow \infty} \mathbf{U}_k = [1, 1, -1, 0, 1, -1]^T$ and $\lim_{k \rightarrow \infty} \mathbf{Y}_k = [1, 1, 0, 0, 1, 0, 1, 0, 0]^T$. The minimum for this case is $\mathcal{J}_L = 1.7321$, verifying Proposition 2. Then, data incompleteness is modeled by Γ_k with $\bar{\Gamma} = \text{diag}\{1, 1, 1, 0.8, 0.1, 1, 0.1, 0.2, 1\}$. Using algorithm (11), where $\mathbf{G}^T \Gamma_k \mathbf{E}_k$ is employed for updating, we have the limits $\lim_{k \rightarrow \infty} \mathbb{E} \mathbf{U}_k = [1, 0.0909, -1, 0, 1, -0.0909]^T$ and $\lim_{k \rightarrow \infty} \mathbb{E} \mathbf{Y}_k = [1, 0.0909, 0, 0, 0.0909, 0, 1, 0, 0]^T$. The expected minimum for this case is $\mathcal{J}_{\text{drift}} = 1.6837 > \mathcal{J}_{\min}$, verifying Proposition 3. ■

Furthermore, while Algorithm 1 offers a practical methodology to address the issue of unknown systems, this algorithm cannot be applied to incomplete data environments. In the case of incomplete data, Algorithm 1 yields a revision to Step 5: $\mathbf{U}_{k+1} = \mathbf{U}_k - \alpha \mathbf{X}_{I_s} \mathbf{Z}_{I_s}^T \Gamma_k \mathbf{E}_k$. Hence, taking mathematical expectation leads to

$$\mathbb{E}[\mathbf{U}_{k+1}] = \mathbb{E}[\mathbf{U}_k] - \alpha \mathbb{E}[\mathbf{X}_{I_s} \mathbf{X}_{I_s}^T] \mathbf{G}^T \bar{\Gamma} (\mathbf{G} \mathbb{E}[\mathbf{U}_k] - \bar{\mathbf{Y}}_r).$$

By observing that $\text{Null}(\mathbb{E}[\mathbf{X}_{I_s} \mathbf{X}_{I_s}^T] \mathbf{G}^T \bar{\Gamma}) = \text{Null}(\mathbf{G}^T \bar{\Gamma})$, the limit of the input sequence in expectation is also $\mathbf{U}_{\text{drift}}$.

Although the matrix Γ_k is indicated by the received data, its expectation $\bar{\Gamma}$ indicating the statistics of incomplete data environments is generally unknown. One may design an estimation $\hat{\Gamma}_k = \frac{1}{k} \sum_{i=1}^k \Gamma_i$ satisfying $\mathbb{E}[\hat{\Gamma}_k] = \bar{\Gamma}$ and use $\hat{\Gamma}_k^{-1}$ to elimi-

nate $\bar{\Gamma}$ in (11). However, $\mathbb{E}[\hat{\Gamma}_k^{-1}] \neq \bar{\Gamma}^{-1}$ indicates that $\hat{\Gamma}_k^{-1}$ is a biased estimation. Thus, this idea is not applicable.

In the learning control algorithm (8), making learning gain matrix \mathbf{L} provide a gradient requires precise system matrices and statistical information of incomplete data. This is very difficult in real world applications. Thus, in the following section, we offer a new idea to avoid providing gradients of \mathbf{L} .

VI. EXTENDED ILC SCHEME

While considering the unachievable tracking problem under incomplete data environments, we observe a random gradient drift. As there is no input ensuring the simultaneous perfect tracking at all time instants, it is required that the control direction (reflected by the learning gain matrix) must be the gradient direction, either implicitly or explicitly, to achieve the best tracking performance. Note that this requirement is unnecessary for the achievable tracking problem (see Remarks 5 and 11). Therefore, we aim to resolve the unachievable tracking problem under incomplete data environments from a novel perspective significantly differing from the sampling strategy in Section IV. In particular, we refine the tracking error by adding a correction term defined by the sampling data such that the modified reference signals for updating are asymptotically achievable as the iteration number increases. Then, any conventional P-type learning control scheme can be used to solve the unachievable tracking problem using the refined tracking errors. The integrated framework is called the extended ILC scheme.

Given $\{\mathbf{U}_i^\circ\}_{i \in \mathcal{I}}$ in which $\mathbf{U}_1^\circ = \mathbf{0}$, we obtain complete output $\mathbf{Y}_i^\circ = \mathbf{G} \mathbf{U}_i^\circ + \mathbf{M} \mathbf{x}_0$ as a sample, $i \in \mathcal{I}$. Then, we compute $\{\mathbf{X}_i, \mathbf{Z}_i\}_{i \in \mathcal{I}}$ according to the sampling strategy. The extended ILC scheme is given in Algorithm 2, where the conventional P-type scheme (8) is integrated with a sampling-based error-refining mechanism.

Algorithm 2 Extended ILC Algorithm

- 1: Determine $\{\mathbf{X}_i, \mathbf{Z}_i\}_{i \in \mathcal{I}}$ and division $\mathcal{D} = \{\mathcal{I}_1, \dots, \mathcal{I}_\tau\}$.
- 2: Initialize arbitrary $\mathbf{U}_0, \mathbf{z}_0 = \mathbf{Y}_r - \mathbf{Y}_1^\circ, k = 0$.
- 3: **while** $k \leq K$ **do**
- 4: Select \mathcal{I}_s with probability p_s , construct \mathbf{Z}_{I_s} .
- 5: $\mathbf{z}_{k+1} = \mathbf{z}_k - \frac{\mathbf{Z}_{I_s} \mathbf{Z}_{I_s}^T}{\|\mathbf{Z}_{I_s}\|_F^2} \mathbf{z}_k$.
- 6: $\mathbf{U}_{k+1} = \mathbf{U}_k - \alpha \mathbf{L} (\Gamma_k \mathbf{E}_k + \Gamma_k \mathbf{z}_{k+1})$.
- 7: $k = k + 1$.
- 8: **end while**
- 9: **return** \mathbf{U}_k

The extended computation \mathbf{z}_k is to actively learn the unachievable part $\mathbf{Y}_r - \mathbf{Y}_d$. In this way, the modified reference $\mathbf{Y}_r - \mathbf{z}_k$ becomes realizable asymptotically (see Lemma 3), and we use the modified reference instead of the original reference for input updating. The following lemma shows the convergence of \mathbf{z}_k , whose proof is put in the Appendix.

Lemma 3: The sequence $\mathbf{Y}_r - \mathbf{Y}_d - \mathbf{z}_k$ converges monotonically to zero in the mean square sense.

Consequently, we have the following convergence result for Algorithm 2, whose proof is put in the Appendix.

Theorem 2: Consider the system (1). Let Assumptions 1 and 2 hold, α be sufficiently small, and all eigenvalues of $\mathbf{L}^T \bar{\Gamma} \mathbf{G}$ be

positive real numbers. Then, for any unachievable reference Y_r , the input sequence U_k generated by Algorithm 2 converges to the desired U_d defined by (6) in the mean square sense.

Remark 12: From a mathematical perspective, $\text{Range}(\bar{\Gamma}G)$ and $\text{Range}(G)$ may be unequal. In other words, as the actual control direction changes from G to $\bar{\Gamma}G$, the learning gain matrix L should serve for $\bar{\Gamma}G$ rather than G . Thus, the selection of L seems to require prior information of $\bar{\Gamma}$. However, completely knowing $\bar{\Gamma}$ is unnecessary due to two facts. First, an interval selection of these parameters is sufficient in practice based on the robust design principle [34], [35]. Second, if we design $L \triangleq \text{diag}\{L_0, \dots, L_{N-1}\}$, the block diagonal matrices L and $\bar{\Gamma}$ are commutable, making the information of $\bar{\Gamma}$ no longer necessary.

The update of U_k is rewritten as $U_{k+1} = U_k - \alpha L \Gamma_k (Y_r - (Y_r - z_{k+1}))$. It is seen that the original reference trajectory Y_r is replaced with $Y_r - z_{k+1}$. As an alternative, it is beneficial to start from Y_r instead of $Y_r - z_0 = Mx_0$ for the learning process, i.e., we replace $Y_r - z_k$ with $(1 - \mu_k)Y_r + \mu_k(Y_r - z_k)$, where μ_k goes from 0 to 1 as $k \rightarrow \infty$.

Corollary 1: If Step 6 of Algorithm 2 is modified as $U_{k+1} = U_k - \alpha L \Gamma_k (E_k + \mu_k z_{k+1})$, Theorem 2 still holds.

The proof of this corollary is put in the Appendix.

We emphasize that the central idea of Algorithm 2 is to introduce an active reference refinement mechanism such that the modified reference is asymptotically achievable. Then, any learning control method for the achievable tracking problem can be integrated into the proposed scheme to resolve the unachievable tracking problem. The primary advantage of this scheme is to relax the requirement for precise gradient information, which might be unavailable due to various conditions. Particularly, the sampling mechanism given in Algorithm 1 can be embedded into the input update step of Algorithm 2. In this case, the entire framework is completely data-driven. The convergence is summarized in the following corollary, whose proof is omitted for brevity.

Corollary 2: In Step 6 of Algorithm 2, if the learning gain matrix L is replaced with $X_{I_s} Z_{I_s}^T \Gamma_k^T$, the convergence results of Theorem 2 still hold.

Note that $\mathbb{E}[X_{I_s} Z_{I_s}^T \Gamma_k^T \Gamma_k G] = \mathbb{E}[X_{I_s} X_{I_s}^T] G^T \bar{\Gamma}^T \bar{\Gamma} G$ is applicable to Lyapunov equation theory based on Lemma 2.

Table I summarizes the information required for the algorithms. Here, PILC, SILC, and EILC represent the P-type ILC, ILC with sampling strategy, and extended ILC, respectively.

TABLE I
CONVERGENCE REQUIREMENTS OF THE ALGORITHMS

Algorithms	Precise system model	Complete data
PILC	yes	yes
SILC	no	yes
EILC	no	no

VII. ILLUSTRATIVE SIMULATIONS

In this section, we validate the theoretical results through numerical simulations conducted on a toy example and a chemical batch reactor system. We evaluate the performance

of gradient-based ILC (GILC), PILC with block diagonal gain, SILC, and EILC under both complete and incomplete data environments. GILC, which utilizes the precise gradient, serves as the benchmark for comparison. The relationships between the simulation outcomes and theoretical results are summarized in Table II.

TABLE II
CORRESPONDENCE BETWEEN FIGURES AND THEORETICAL RESULTS

Fig. 1	Proposition 2 and Claim 1
Fig. 3	Proposition 3
Fig. 1	Theorem 1
Fig. 4	Theorem 2

In practical implementations, ILC is often deployed within a networked control structure, where information and data are transmitted through communication networks. However, network congestion and limited bandwidth can lead to data packet loss during transmission [13]. Therefore, data dropout is a typical scenario in incomplete data environments, which is adopted in this section to illustrate the challenges posed by incomplete data.

A. Numerical Case

We consider the system represented by (A_t, B_t, C_t) ,

$$A_t = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 \sin(t-5) & 0 \\ 0 & 0 & 0.7e^{t-13} \end{bmatrix}, \quad B_t = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$C_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1-0.1t & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Let the reference trajectory be

$$y_r(t) = \begin{bmatrix} 0.18t^2 - 4.8t - 14.8 \\ -0.007t^3 + 0.235t^2 - 4.125t + 11.625 \\ 6 \sin^2(t-5) - 0.4t - 1.8 \end{bmatrix}.$$

The following settings are considered: the time interval is set to $[0, 10]$; the time instants calculated in (3) are $1, 2, \dots, 10$; and the initial input and state are zero vectors.

1) *Complete Data Environment:* For GILC, we set $L = G^T$. For PILC and EILC, we set $L_t^c = (C_{t+1} B_t)^T$. The step size α is set to 0.25 for all schemes. To implement the sampling strategy for SILC and EILC, we first randomly generate $pN + 1$ vectors denoted by U_i° ($i = 0, 1, \dots, pN$) such that $U_i^\circ - U_0^\circ$ are linearly independent. Then, Y_i° ($i = 0, 1, \dots, pN$) is produced by the system accordingly. This prior given data constitutes the set of sample pairs (X_i, Z_i) , where $X_i = U_i^\circ - U_0^\circ$ and $Z_i = Y_i^\circ - Y_0^\circ$ for $i \in I' \triangleq 1, 2, \dots, pN$. The rule for selecting samples is as follows: in each iteration, ten sample pairs are randomly and uniformly extracted from the set $\{(X_i, Z_i)\}_{i \in I'}$ to obtain a sufficient gradient estimate without causing heavy computation, and the selected pairs are used to construct X_{I_s} and Z_{I_s} . This avoids an explicit expression of \mathcal{D} .

To assess the performance of each scheme, all results are normalized by the error at the first iteration, ensuring that all

lines in the corresponding figures start from 1^0 .

Fig. 1 shows the decrease in the input error $\|\delta_k\|^2$ for the four schemes: GILC, PILC, SILC, and EILC. The y-axis scale is logarithmic. Based on the 30th iteration as the dividing point, we analyze the performance of the different schemes. For the first stage ($k < 30$), GILC, which utilizes full system information, achieves the fastest convergence and is considered the benchmark (solid line). PILC (dotted line) and EILC (dashed line) utilize only the control and measurement matrix knowledge, making them slower than GILC. SILC (dashed-dotted line) is the slowest among the four schemes, as its search space is at most ten-dimensional according to Proposition 1, whereas the other schemes have a 20-dimensional search space. Additionally, EILC exhibits slower convergence than PILC because it uses a revised reference $Y_r - z_k$, which can be far from the best achievable reference in the early stage. For the second stage, when the learning gain L_t^c does not provide the precise gradient, PILC does not converge to the desired input, even in the case of complete data. This behavior is explained by Proposition 2 and Claim 1. It can be observed that GILC, EILC, and SILC show zero convergence tendencies, demonstrating their effectiveness, with SILC being particularly guaranteed by Theorem 1.

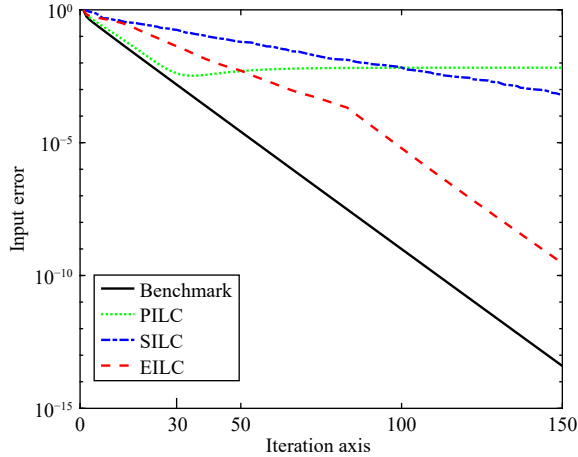


Fig. 1. Input error profiles for GILC, PILC, SILC, and EILC, where GILC is labeled as a benchmark.

Furthermore, Fig. 2 depicts the decrease in the tracking error, providing further support for the observations described above. In the subplot, we observe that GILC, EILC, and SILC achieve the best approximation performance, while PILC significantly deviates from the other three methods in terms of approximation performance.

2) *Incomplete Data Environments*: We discard PILC as it is invalid even under complete data and maintain the benchmark performance under complete data for comparison. Following [13], we model the random data dropout using $\gamma_k(t)$. Here, $\gamma_k(t)$ is generated according to the Bernoulli distribution with $\mathbb{E}[\gamma_k(t)] = \gamma(t)$, where $\gamma(t)$ is predetermined and unknown to any algorithm. The affected tracking error $\gamma_k(t)(y_k(t) - y_r(t))$ is used in the schemes for updating. Other settings remain the same as in the previous subsection.

Fig. 3 depicts the input error decrease in GILC (dotted line),

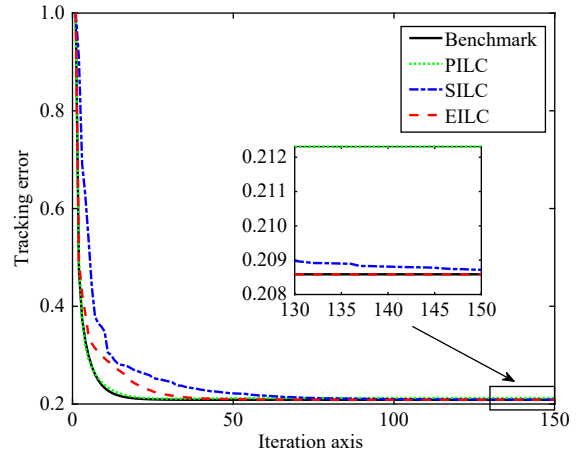


Fig. 2. Tracking error profiles for GILC, PILC, SILC, and EILC.

SILC, and EILC using incomplete data. As indicated by our analysis in Proposition 3, GILC and SILC lose their efficacy due to gradient drift. In particular, GILC and SILC cannot handle random missing data and consequently exhibit oscillatory behavior without achieving convergence. Only EILC demonstrates a continuous decreasing trend, confirming Theorem 2. As shown in Fig. 4, while EILC reaches the benchmark performance, GILC and SILC fail to converge to the benchmark and exhibit oscillatory behavior. This highlights the effectiveness of the EILC scheme.

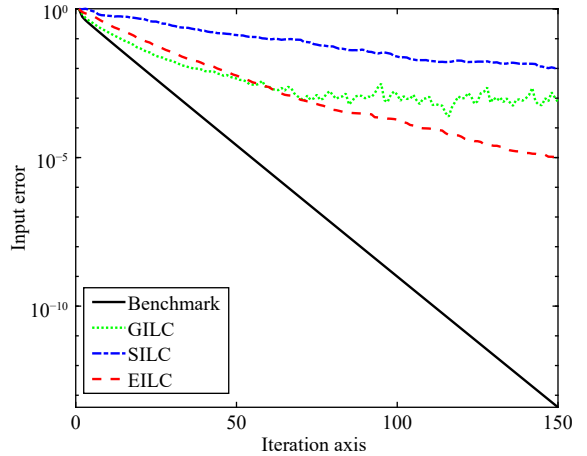


Fig. 3. Input error profiles for GILC, SILC, and EILC.

B. Chemical Batch Reactor Simulation

ILC research has considered a nonlinear chemical batch reactor, which exhibits a second-order exothermic reaction $A \rightarrow B$ [30], [36]. In this reactor, the temperature of the cooling jacket is directly manipulated, while the objective is to track a reference temperature trajectory. The dynamics of the reactor can be described by the following continuous equations:

$$\begin{aligned} \dot{T} &= -\frac{UA}{MC_p}(T - T_j) + \frac{(-\Delta H)V}{MC_p}k_0 e^{-E/RT} C_A^2 \\ \dot{C}_A &= -k_0 e^{-E/RT} C_A^2 \end{aligned} \quad (15)$$

where T , C_A , and T_j represent the reaction temperature, con-

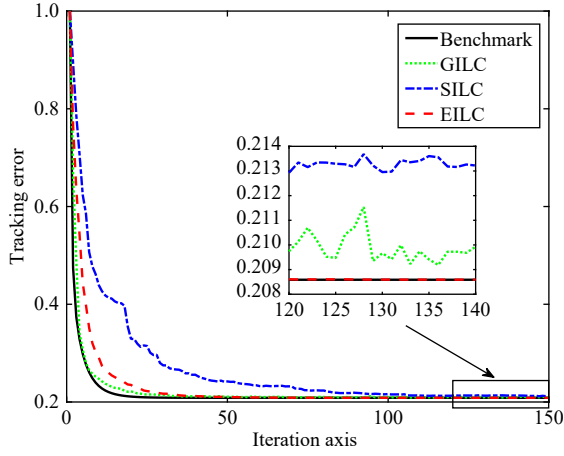


Fig. 4. Tracking error profiles for GILC, SILC, and EILC.

centration of reactant A , and temperature of the coolant stream, respectively. The parameters used in the model are as follows [36]:

$$\frac{UA}{MC_p} = 0.09 \text{ (1/min)}, \quad \frac{(-\Delta H)V}{MC_p} = 1.64 \text{ (Kl/mol)}$$

$$k_0 = 2.53 \times 10^{19} \text{ (l/mol min)}, \quad \frac{E}{R} = 13550 \text{ (K)}.$$

We define the state variables as $x_k(t) = [T, C_A]^T$ and the input as $u_k(t) = T_j$. Additionally, we introduce the following notations:

$$f(x_k(t)) = \begin{bmatrix} -\frac{UA}{MC_p}T + \frac{(-\Delta H)V}{MC_p}k_0e^{-E/RT}C_A^2 \\ -k_0e^{-E/RT}C_A^2 \end{bmatrix}$$

$$g_t = \begin{bmatrix} \frac{UA}{MC_p} \\ 0 \end{bmatrix}.$$

With these definitions, we can rewrite (15) as a state-space model,

$$\dot{x}_k(t) = f(x_k(t)) + g_t u_k(t) \quad (16)$$

where the initial state is $[T_0, C_{A0}]^T = [25^\circ \text{C}, 0.9 \text{ mol/l}]^T$ for all iterations. It is worth noting that the system (16) has two outputs and one input. There may exist reference trajectories that are not achievable, making the ILC tracking problem infeasible. Let the reference trajectory be defined as follows:

$$T(t) = \begin{cases} \frac{7}{9}t + \frac{200}{9}, & 0 \leq t < 10 \\ \frac{7}{30}t + \frac{62}{3}, & 10 \leq t < 40 \\ \frac{3}{10}t + 15, & 40 \leq t < 50 \\ 33, & 50 \leq t < 60 \\ -\frac{4}{5}t + 81, & 60 \leq t < 70 \\ 25, & 70 \leq t < 80 \end{cases}$$

$$C_A(t) = 0.1734e^{-1.927t} + 0.009325.$$

For the system described by (15), a linear model-based

design approach can be employed [36]. Although the algorithms in this study are designed for linear systems, it is important to acknowledge the existence of nonlinear systems such as (15). Consequently, we discretize the system (16) using inputs $\{U_i^\circ\}_{i \in \mathcal{I}}$ and obtain the corresponding outputs $\{Y_i^\circ\}_{i \in \mathcal{I}}$, where the time interval is from 0 to 80 min with a sampling time of 1 min. We then preprocess this data using Algorithm 1: $Z_i = Y_i^\circ - Y_1^\circ$, $X_i = U_i^\circ - U_1^\circ$, $i \in \mathcal{I}$. Subsequently, we construct the learning gain L for P-type ILC by employing the least squares principle,

$$L^T = \arg \min_{L \in \mathbb{R}^{160 \times 80}} \sum_{i \in \mathcal{I}} \|Z_i - LX_i\|^2.$$

The sampling data required for Algorithms 1 and 2 are constructed from the available data, and additional operations are not necessary for subsequent iterations. Each iteration involves a random selection of 10 samples. Notably, we do not present an input error analysis due to the inability to analytically solve for the desired input trajectory in the presence of system nonlinearity. Therefore, our performance analysis focuses on tracking errors.

1) *Complete Data Environment*: In the complete data environment, PILC, SILC, and EILC are applied to the reactor tracking problem using the same learning gain matrix L . The step size is set to $\alpha = 1$, and the initial input is $u_0(t) = 26$ for $t = 0, \dots, 80$. The actual tracking error profiles of PILC, SILC, and EILC are shown in Fig. 5.

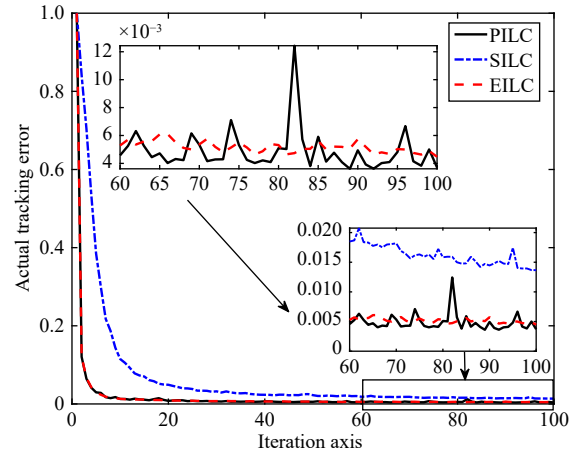


Fig. 5. Actual tracking error profiles for PILC, SILC, and EILC.

Fig. 5 demonstrates that the actual tracking error decreases as the iterations progress, indicating that all three methods improve their performance over time. The y-axis values in the figure are normalized by the first iteration error. SILC (dashed-dotted line) shows slower convergence compared to PILC (solid line) and EILC (dashed line). PILC and EILC exhibit similar performance levels since they use the same learning gain matrix. However, from the subplot, it can be observed that EILC yields a more steady convergence.

2) *Incomplete Data Environment*: Incomplete data environments are simulated by introducing data dropouts, where the tracking error at each time instant is multiplied by a Bernoulli random variable. PILC, SILC, and EILC are tested on the nonlinear system (16) using the same learning gain matrix L

for PILC and EILC. Due to the presence of data dropouts, which can reduce the convergence rate, we perform 150 iterations to observe the convergence trends.

Fig. 6 shows the actual tracking errors of PILC, SILC, and EILC under data dropouts. It can be observed that the performance comparison of the algorithms is similar to that of the complete data case. However, the random dropouts induce larger oscillations, making it difficult to conclude that EILC performs significantly better than PILC. This indicates the need for improved application methods for nonlinear systems, as the sampling data of nonlinear systems may not fully reflect the global system properties.

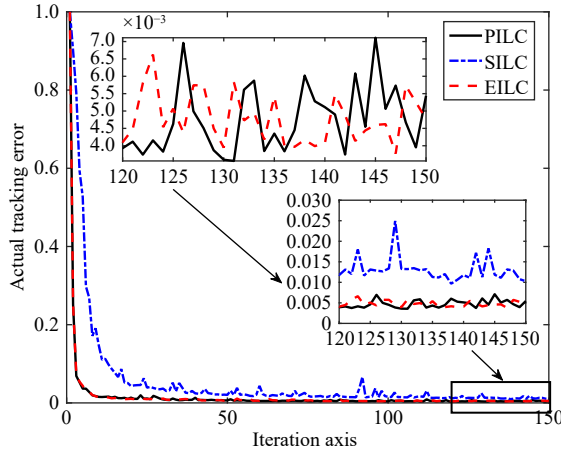


Fig. 6. Actual tracking errors profiles of PILC, SILC, and EILC.

In Fig. 7, the 1st, 20th, 40th, 60th, and 80th iterations of the reaction temperature generated by EILC under incomplete data are presented. Since temperature control is the main objective, only the reaction temperature is plotted (dotted red line). The figure shows that the reference trajectory (solid black line) is gradually tracked. Due to the fixed initial state of 25 instead of the reference value 22.2, the reaction temperature in the first few minutes shows a tendency to approach the reference but does not perfectly reach it. Overall, the output achieves satisfactory tracking performance within 20 iterations. This indicates that EILC can provide benefits for a class

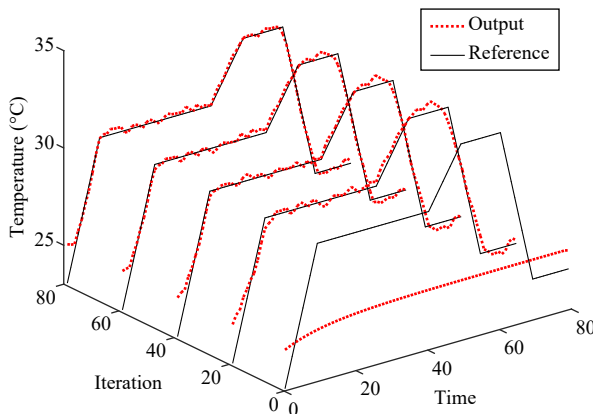


Fig. 7. Reaction temperature at the 1st, 20th, 40th, 60th, and 80th iterations for EILC under incomplete data.

of nonlinear systems.

VIII. CONCLUSIONS

This study has successfully formulated and characterized the unachievable tracking problem, which arises when the reference cannot be precisely achieved at all time instants. In response to this problem, we have provided and analyzed different ILC solutions based on specific requirements. Notably, we extensively clarified the necessity of precise gradients in the conventional P-type ILC. Moreover, when the system information is unknown, preventing the establishment of gradients, we identified the capabilities of the P-type ILC for linear time-varying systems and devised a data-driven approach. Additionally, to address systems operating under incomplete data environments, we proposed an extended ILC scheme to mitigate the gradient drift effect. Through investigations, we demonstrated that the proposed learning control algorithms can achieve the best approximation performance of the unrealizable reference in a mean square sense. For future research, it is imperative to explore the integration of offline and online data to enhance these methods against disturbances and noises.

APPENDIX

Lemma 4 ([23]): Let a real non-negative number sequence $\{x_k\}$ satisfy $x_{k+1} = (1 - a_k)x_k + b_k$, $k \geq 0$, where $a_k \in [0, 1)$ and $\sum_{k=1}^{\infty} b_k < \infty$. Then, $\lim_{k \rightarrow \infty} x_k = 0$ if and only if $\sum_{k=1}^{\infty} a_k = \infty$, $\forall x_0 \neq 0$.

Proof of Lemma 3: We denote $r_k \triangleq \bar{Y}_r - \bar{Y}_d - z_k = Y_r - Y_d - z_k$ and remind that $\mathbf{G}^T(\bar{Y}_r - \bar{Y}_d) = 0$. For $k \geq 0$, by Step 5 in Algorithm 2,

$$\begin{aligned} r_{k+1} &= \bar{Y}_r - \bar{Y}_d - z_k + \frac{Z_{I_s} Z_{I_s}^T}{\|Z_{I_s}\|_F^2} z_k \\ &= r_k + \mathbf{G} \frac{X_{I_s} X_{I_s}^T}{\|Z_{I_s}\|_F^2} \mathbf{G}^T (z_k - (\bar{Y}_r - \bar{Y}_d)) \\ &= r_k - \mathbf{G} \frac{X_{I_s} X_{I_s}^T}{\|Z_{I_s}\|_F^2} \mathbf{G}^T r_k. \end{aligned} \quad (17)$$

Taking the Euclidean norm and conditional expectation gives

$$\mathbb{E}[\|r_{k+1}\|^2 | z_k] = r_k^T \mathbb{E} \left[\left(I_{q_N} - \mathbf{G} \frac{X_{I_s} X_{I_s}^T}{\|Z_{I_s}\|_F^2} \mathbf{G}^T \right)^2 \right] r_k. \quad (18)$$

Next, we specify the contraction of r_k , i.e., $\mathbb{E}[\|r_{k+1}\|^2] < \mathbb{E}[\|r_k\|^2]$. The key point here is to prove that r_k is orthogonal to the 1-characteristic subspace of $\mathbb{E}[(I_{q_N} - \mathbf{G} \frac{X_{I_s} X_{I_s}^T}{\|Z_{I_s}\|_F^2} \mathbf{G}^T)^2]$. The following are worth noting:

$$\begin{aligned} &\mathbb{E} \left[\left(I_{q_N} - \mathbf{G} \frac{X_{I_s} X_{I_s}^T}{\|Z_{I_s}\|_F^2} \mathbf{G}^T \right)^2 \right] \\ &= I_{q_N} + \mathbb{E} \left[\mathbf{G} \frac{X_{I_s} X_{I_s}^T}{\|Z_{I_s}\|_F^2} \mathbf{G}^T \right]^2 - 2 \mathbb{E} \left[\mathbf{G} \frac{X_{I_s} X_{I_s}^T}{\|Z_{I_s}\|_F^2} \mathbf{G}^T \right]. \end{aligned} \quad (19)$$

Based on the above, we claim that

$$\begin{aligned}
\text{Null}(\mathbf{G}\mathbf{G}^T) &= \text{Null}(\mathbb{E}[\mathbf{G} \frac{\mathbf{X}_{I_s} \mathbf{X}_{I_s}^T}{\|\mathbf{Z}_{I_s}\|_F^2} \mathbf{G}^T]) \\
&= \text{Null}(\mathbb{E}(\mathbf{G} \frac{\mathbf{X}_{I_s} \mathbf{X}_{I_s}^T}{\|\mathbf{Z}_{I_s}\|_F^2} \mathbf{G}^T)^2). \quad (20)
\end{aligned}$$

The first equality can be proved similarly to Lemma 1 by verifying the SPD $\mathbb{E}[\frac{\mathbf{X}_{I_s} \mathbf{X}_{I_s}^T}{\|\mathbf{Z}_{I_s}\|_F^2}]$. Regarding the second equality, we note that $\forall \mathbf{v} \neq \mathbf{0}$, $(\mathbf{G} \frac{\mathbf{X}_{I_s} \mathbf{X}_{I_s}^T}{\|\mathbf{Z}_{I_s}\|_F^2} \mathbf{G}^T)^2 \mathbf{v} = \mathbf{0} \Leftrightarrow \mathbf{G} \frac{\mathbf{X}_{I_s} \mathbf{X}_{I_s}^T}{\|\mathbf{Z}_{I_s}\|_F^2} \mathbf{G}^T \mathbf{v} = \mathbf{0}$. Therefore, (20) must hold.

We claim that $r_k \in \text{Range}(\mathbf{G})$, as proven by the induction principle. For $k=0$, $r_0 = \bar{Y}_r - \bar{Y}_d - z_0 = -\bar{Y}_d \in \text{Range}(\mathbf{G})$; for $k \geq 0$, $r_{k+1} = r_k - \mathbf{G} \mathbf{X}_{I_s} \mathbf{X}_{I_s}^T \mathbf{G}^T r_k / \|\mathbf{Z}_{I_s}\|_F^2 \in \text{Range}(\mathbf{G})$. Moreover, $Y_r - z_k = r_k + Y_d$ can be a realizable reference.

For a matrix M , $\text{Range}(M) = \text{Null}^\perp(M^T)$. Thus, r_k is completely contained in the orthogonal complementary space of $\text{Null}(\mathbf{G}^T) = \text{Null}(\mathbf{G}\mathbf{G}^T)$. It is true that r_k is orthogonal to the 1-characteristic subspace of the left-hand side of (19), using (20).

On the other hand, the largest eigenvalue of the left-hand side of (19) is not greater than one because this term is equal to

$$\mathbb{E} \left[\left(\mathbf{I}_{qN} - \frac{\mathbf{G} \mathbf{X}_{I_s} (\mathbf{G} \mathbf{X}_{I_s})^T}{\text{trace}((\mathbf{G} \mathbf{X}_{I_s})^T (\mathbf{G} \mathbf{X}_{I_s}))} \right)^2 \right]$$

in which the sum of the eigenvalues of a symmetric matrix is equal to its trace.

In short, r_k is completely within the contraction space of (19). Hence, we denote $0 \leq \rho_2 < 1$ as the largest eigenvalue of (19) that is less than 1, then (18) drives $\mathbb{E}[\|r_{k+1}\|^2 | z_k] \leq \rho_2 \|r_k\|^2$. Retaking expectation gives $\mathbb{E}\|r_{k+1}\|^2 \leq \rho_2 \mathbb{E}\|r_k\|^2$, consequently proving the monotonic exponential convergence of r_k in the mean square sense. ■

Proof of Theorem 2: Recall the definition $\delta_k = U_k - U_d$. Subsequently, we can derive Step 6 of Algorithm 2 to obtain

$$\begin{aligned}
\delta_{k+1} &= U_k - U_d - \alpha \mathbf{L} \Gamma_k (E_k + z_{k+1}) \\
&= \delta_k - \alpha \mathbf{L} \Gamma_k (\mathbf{G} U_k - \mathbf{G} U_d + \bar{Y}_d - \bar{Y}_r + z_{k+1}) \\
&= (\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k + \alpha \mathbf{L} \Gamma_k r_{k+1}.
\end{aligned}$$

Next, we claim the contraction of $(\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k$ in the expectation sense, noting that all eigenvalues of $\mathbf{L} \bar{\Gamma} \mathbf{G}$ are positive real numbers. By the Lyapunov stability theory, a unique SPD \mathbf{Q} satisfying $(\mathbf{L} \bar{\Gamma} \mathbf{G})^T \mathbf{Q} + \mathbf{Q} \mathbf{L} \bar{\Gamma} \mathbf{G} = \mathbf{I}_{pN}$ is observed. Still, we check the following derivations:

$$\begin{aligned}
&\delta_k^T (\mathbf{I}_{pN} - \alpha \mathbf{G}^T \Gamma_k \mathbf{L}^T) \mathbf{Q} (\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k \\
&= \delta_k^T \mathbf{Q} \delta_k + \alpha^2 \delta_k^T \mathbf{G}^T \Gamma_k \mathbf{L}^T \mathbf{Q} \mathbf{L} \Gamma_k \mathbf{G} \delta_k \\
&\quad - \alpha \delta_k^T (\mathbf{G}^T \Gamma_k \mathbf{L}^T \mathbf{Q} + \mathbf{Q} \mathbf{L} \Gamma_k \mathbf{G}) \delta_k.
\end{aligned}$$

Noting the independence of Γ_k with respect to the iteration label k , we have $\mathbb{E}[\mathbf{G}^T \Gamma_k \mathbf{L}^T \mathbf{Q} + \mathbf{Q} \mathbf{L} \Gamma_k \mathbf{G}] = (\mathbf{L} \bar{\Gamma} \mathbf{G})^T \mathbf{Q} + \mathbf{Q} \mathbf{L} \bar{\Gamma} \mathbf{G} = \mathbf{I}_{pN}$. Therefore, there are suitable constants c_3 and c_4 , such that the inequalities $\delta_k^T \mathbb{E}[\mathbf{G}^T \Gamma_k \mathbf{L}^T \mathbf{Q} \mathbf{L} \Gamma_k \mathbf{G}] \delta_k \leq c_3 \delta_k^T \mathbf{Q} \delta_k$ and $\delta_k^T \mathbb{E}[\mathbf{G}^T \Gamma_k \mathbf{L}^T \mathbf{Q} + \mathbf{Q} \mathbf{L} \Gamma_k \mathbf{G}] \delta_k \geq c_4 \delta_k^T \mathbf{Q} \delta_k$ hold. As a result, we have $\mathbb{E}[\delta_k^T (\mathbf{I}_{pN} - \alpha \mathbf{G}^T \Gamma_k \mathbf{L}^T) \mathbf{Q} (\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k | x_k] \leq (1 +$

$\alpha^2 c_3 - \alpha c_4) \delta_k^T \mathbf{Q} \delta_k$. Provided that the stepsize is sufficiently small, satisfying $\alpha < c_4/c_3$, we have $0 < c_0 \triangleq 1 + \alpha^2 c_3 - \alpha c_4 < 1$.

Then, taking a weighted norm on δ_{k+1} with respect to \mathbf{Q} , i.e., $\|\delta_{k+1}\|_{\mathbf{Q}}^2$, and applying the Cauchy-Schwartz inequality and the Young's inequality, we have, for $\forall \epsilon > 0$,

$$\begin{aligned}
\|\delta_{k+1}\|_{\mathbf{Q}}^2 &= \delta_{k+1}^T \mathbf{Q} \delta_{k+1} \\
&= \|(\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k\|_{\mathbf{Q}}^2 + \|\alpha \mathbf{L} \Gamma_k r_{k+1}\|_{\mathbf{Q}}^2 \\
&\quad + 2\alpha r_{k+1}^T \Gamma_k \mathbf{L}^T \mathbf{Q} (\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k \\
&\leq \|(\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k\|_{\mathbf{Q}}^2 + \|\alpha \mathbf{L} \Gamma_k r_{k+1}\|_{\mathbf{Q}}^2 \\
&\quad + 2\|\alpha \mathbf{L} \Gamma_k r_{k+1}\|_{\mathbf{Q}} \|(\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k\|_{\mathbf{Q}} \\
&\leq (1 + \epsilon) \|(\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k\|_{\mathbf{Q}}^2 \\
&\quad + \frac{1 + \epsilon}{\epsilon} \|\alpha \mathbf{L} \Gamma_k r_{k+1}\|_{\mathbf{Q}}^2.
\end{aligned}$$

Subsequently, taking conditional expectation gives

$$\begin{aligned}
&\mathbb{E}[\|\delta_{k+1}\|_{\mathbf{Q}}^2 | x_k, z_k] \\
&\leq (1 + \epsilon) \mathbb{E}[\|(\mathbf{I}_{pN} - \alpha \mathbf{L} \Gamma_k \mathbf{G}) \delta_k\|_{\mathbf{Q}}^2 | x_k, z_k] \\
&\quad + \frac{1 + \epsilon}{\epsilon} \mathbb{E}[\|\alpha \mathbf{L} \Gamma_k r_{k+1}\|_{\mathbf{Q}}^2 | x_k, z_k] \\
&\leq (1 + \epsilon) c_0 \|\delta_k\|_{\mathbf{Q}}^2 + \frac{1 + \epsilon}{\epsilon} \alpha^2 c_5 \mathbb{E}[\|r_{k+1}\|^2 | x_k, z_k] \\
&\leq (1 + \epsilon) c_0 \|\delta_k\|_{\mathbf{Q}}^2 + \frac{1 + \epsilon}{\epsilon} \alpha^2 c_5 \rho_2 \|r_k\|^2
\end{aligned}$$

where c_5 is a positive constant. Retaking expectation gives

$$\mathbb{E}[\|\delta_{k+1}\|_{\mathbf{Q}}^2] \leq (1 + \epsilon) c_0 \mathbb{E}[\|\delta_k\|_{\mathbf{Q}}^2] + \frac{1 + \epsilon}{\epsilon} \alpha^2 c_5 \rho_2 \mathbb{E}[\|r_k\|^2].$$

As a result, we fix appropriate α and ϵ in place, such that $(1 + \epsilon) c_0 = 1 - \rho_1 < 1$. Then, we can apply Lemma 4 on

$$\mathbb{E}[\|\delta_{k+1}\|_{\mathbf{Q}}^2] \leq (1 - \rho_1) \mathbb{E}[\|\delta_k\|_{\mathbf{Q}}^2] + \frac{1 + \epsilon}{\epsilon} \alpha^2 c_5 \rho_2 \mathbb{E}[\|r_k\|^2].$$

Note that

$$\begin{aligned}
\sum_{k=1}^{\infty} \frac{1 + \epsilon}{\epsilon} \alpha^2 c_5 \rho_2 \mathbb{E}[\|r_k\|^2] &= \frac{1 + \epsilon}{\epsilon} \alpha^2 c_5 \rho_2 \sum_{k=1}^{\infty} \mathbb{E}[\|r_k\|^2] \\
&\leq \frac{1 + \epsilon}{\epsilon} \alpha^2 c_5 \|r_0\|^2 \sum_{k=1}^{\infty} \rho_2^{k+1}
\end{aligned}$$

is convergent because of $\rho_2 < 1$ by Lemma 3. Thus, $\mathbb{E}[\|\delta_{k+1}\|_{\mathbf{Q}}^2] \rightarrow 0$ by Lemma 4 holds. ■

Proof of Corollary 1: First, we redefine $r_k = \bar{Y}_r - \bar{Y}_d - \mu_k z_k$, which is still convergent to zero in the mean square sense. Next, the expression below is produced using the Cauchy-Schwartz and Young's inequalities:

$$\begin{aligned}
\|r_k\|^2 &= \|\bar{Y}_r - \bar{Y}_d - z_k\|^2 + (1 - \mu_k)^2 \|z_k\|_2^2 \\
&\quad + 2(1 - \mu_k) z_k^T (\bar{Y}_r - \bar{Y}_d - z_k) \\
&\leq 2\|\bar{Y}_r - \bar{Y}_d - z_k\|^2 + 2(1 - \mu_k)^2 \|z_k\|_2^2
\end{aligned}$$

and then,

$$\mathbb{E}[\|r_k\|^2] \leq 2\mathbb{E}[\|\bar{Y}_r - \bar{Y}_d - z_k\|^2] + 2(1 - \mu_k)^2 \mathbb{E}[\|z_k\|_2^2].$$

We observed that the two terms on the right-hand side of the above inequality tend to zero because $\mathbb{E}[\|z_k\|_2^2]$ is bounded. Then, the rest of the proof can be completed similar to Theorem 2. ■

REFERENCES

- [1] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, 2006.
- [2] S. R. Nekoo, J. Á. Acosta, G. Heredia, and A. Ollero, "A PD-type state-dependent riccati equation with iterative learning augmentation for mechanical systems," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 8, pp. 1499–1511, 2022.
- [3] D. Shen and C. Zhang, "Zero-error tracking control under unified quantized iterative learning framework via encoding/decoding method," *IEEE Trans. Cybernetics*, vol. 52, no. 4, pp. 1979–1991, 2022.
- [4] D. Shen, N. Huo, and S. S. Saab, "A probabilistically quantized learning control framework for networked linear systems," *IEEE Trans. Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7559–7573, 2022.
- [5] D. Shen and Y. Xu, "Iterative learning control for discrete-time stochastic systems with quantized information," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 1, pp. 59–67, 2016.
- [6] S. S. Saab, "Stochastic P-type/D-type iterative learning control algorithms," *Int. Journal of Control*, vol. 76, no. 2, pp. 139–148, 2003.
- [7] X. Dai, S. Tian, Y. Peng, and W. Luo, "Closed-loop P-type iterative learning control of uncertain linear distributed parameter systems," *IEEE/CAA J. Autom. Sinica*, vol. 1, no. 3, pp. 267–273, 2014.
- [8] D. Meng and K. L. Moore, "Robust iterative learning control for nonrepetitive uncertain systems," *IEEE Trans. Automatic Control*, vol. 62, no. 2, pp. 907–913, 2017.
- [9] C. Liu and X. Ruan, "Input-output-driven gain-adaptive iterative learning control for linear discrete-time-invariant systems," *Int. J. Robust and Nonlinear Control*, vol. 31, no. 17, pp. 8551–8568, 2021.
- [10] R. Chi, H. Zhang, B. Huang, and Z. Hou, "Quantitative data-driven adaptive iterative learning control: From trajectory tracking to point-to-point tracking," *IEEE Trans. Cybernetics*, vol. 52, no. 6, pp. 4859–4873, 2022.
- [11] S. He, W. Chen, D. Li, Y. Xi, Y. Xu, and P. Zheng, "Iterative learning control with data-driven-based compensation," *IEEE Trans. Cybernetics*, vol. 52, no. 8, pp. 7492–7503, 2022.
- [12] C. Hu, R. Zhou, Z. Wang, Y. Zhu, and M. Tomizuka, "Real-time iterative compensation framework for precision mechatronic motion control systems," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 7, pp. 1218–1232, 2022.
- [13] X. Bu, Z. Hou, S. Jin, and R. Chi, "An iterative learning control design approach for networked control systems with data dropouts," *Int. J. Robust and Nonlinear Control*, vol. 26, no. 1, pp. 91–109, 2016.
- [14] J. Chen, C. Hua, and X. Guan, "Iterative learning model-free control for networked systems with dual-direction data dropouts and actuator faults," *IEEE Trans. Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 5232–5240, 2021.
- [15] H.-S. Ahn, K. L. Moore, and Y. Chen, "Stability of discrete-time iterative learning control with random data dropouts and delayed controlled signals in networked control systems," in *Proc. 10th Int. Conf. Control, Automation, Robotics and Vision*, 2008, pp. 757–762.
- [16] J. Liu and X. Ruan, "Networked iterative learning control design for discrete-time systems with stochastic communication delay in input and output channels," *Int. J. Systems Science*, vol. 48, no. 9, pp. 1844–1855, 2017.
- [17] X. Li, J.-X. Xu, and D. Huang, "An iterative learning control approach for linear systems with randomly varying trial lengths," *IEEE Trans. Automatic Control*, vol. 59, no. 7, pp. 1954–1960, 2014.
- [18] D. Shen and S. S. Saab, "Noisy-output-based direct learning tracking control with Markov nonuniform trial lengths using adaptive gains," *IEEE Trans. Automatic Control*, vol. 67, no. 8, pp. 4123–4130, 2022.
- [19] X. Li, K. Wang, and D. Liu, "An improved result of multiple model iterative learning control," *IEEE/CAA J. Autom. Sinica*, vol. 1, no. 3, pp. 315–322, 2014.
- [20] D. Shen, G. Qu, and X. Yu, "Averaging techniques for balancing learning and tracking abilities over fading channels," *IEEE Trans. Automatic Control*, vol. 66, no. 6, pp. 2636–2651, 2021.
- [21] S. Zhu, X. Wang, and H. Liu, "Observer-based iterative and repetitive learning control for a class of nonlinear systems," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 5, pp. 990–998, 2018.
- [22] G. Qu and D. Shen, "Stochastic iterative learning control with faded signals," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 5, pp. 1196–1208, 2019.
- [23] D. Shen and X. Yu, "Learning tracking over unknown fading channels based on iterative estimation," *IEEE Trans. Neural Networks and Learning Systems*, vol. 33, no. 1, pp. 48–60, 2022.
- [24] Y. Chen, C. Wen, Z. Gong, and M. Sun, "An iterative learning controller with initial state learning," *IEEE Trans. Automatic Control*, vol. 44, no. 2, pp. 371–376, 1999.
- [25] Y. Hui, R. Chi, B. Huang, and Z. Hou, "Extended state observer-based data-driven iterative learning control for permanent magnet linear motor with initial shifts and disturbances," *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol. 51, no. 3, pp. 1881–1891, 2021.
- [26] X. He, Z. Sun, Z. Geng, and A. Robertsson, "Exponential set-point stabilization of underactuated vehicles moving in three-dimensional space," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 2, pp. 270–282, 2022.
- [27] D. Shen, C. Liu, L. Wang, and X. Yu, "Iterative learning tracking for multisensor systems: A weighted optimization approach," *IEEE Trans. Cybernetics*, vol. 51, no. 3, pp. 1286–1299, 2021.
- [28] Z. Zhang, H. Jiang, and D. Shen, "Extended iterative learning control for inconsistent tracking problems with random dropouts," in *Proc. IEEE 11th Data Driven Control and Learning Systems Conf.*, 2022, pp. 935–940.
- [29] R. Chi, Z. Hou, B. Huang, and S. Jin, "A unified data-driven design framework of optimality-based generalized iterative learning control," *Computers & Chemical Engineering*, vol. 77, pp. 10–23, 2015.
- [30] L. Ma, X. Liu, X. Kong, and K. Y. Lee, "Iterative learning model predictive control based on iterative data-driven modeling," *IEEE Trans. Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3377–3390, 2021.
- [31] S. S. Saab, D. Shen, M. Orabi, D. Kors, and R. H. Jaafar, "Iterative learning control: Practical implementation and automation," *IEEE Trans. Industrial Electronics*, vol. 69, no. 2, pp. 1858–1866, 2022.
- [32] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.
- [33] D. Shen, "Iterative learning control with incomplete information: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 5, pp. 885–901, 2018.
- [34] H.-S. Ahn, K. Moore, and Y. Chen, "Monotonic convergent iterative learning controller design based on interval model conversion," *IEEE Trans. Automatic Control*, vol. 51, no. 2, pp. 366–371, 2006.
- [35] H.-S. Ahn, K. L. Moore, and Y. Chen, "Stability analysis of discrete-time iterative learning control systems with interval uncertainty," *Automatica*, vol. 43, no. 5, pp. 892–902, 2007.
- [36] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, pp. 641–657, 2000.



Zeyi Zhang received the B. S. degree in mathematics from Qingdao University in 2020. He is currently working toward the Ph.D. degree with the School of Mathematics, Renmin University of China. His research interests include learning control and iterative methods.



Hao Jiang received the B. S. degree from the Harbin Institute of Technology in 2009, and the Ph.D. degree from the University of Hong Kong, Hong Kong, China, in 2013. She is an Associate Professor with the School of Mathematics, Renmin University of China. Her research interests include learning-based modeling, optimization, and control of complex systems.



Dong Shen (Senior Member, IEEE) received the B. S. degree in mathematics from School of Mathematics, Shandong University, in 2005, and the Ph.D. degree in mathematics from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS), in 2010. From 2010 to 2012, he was a Post-Doctoral Fellow with the Institute of Automation, CAS. From 2012 to 2019, he was with the College of Information Science and Technology, Beijing University of Chemical Technology. In 2016 and 2019, he was a Visiting Scholar with the National University of Singapore, Singapore, and RMIT University, Australia, respectively. Since 2020,

he has been a Professor with the School of Mathematics, Renmin University of China. His research interests include iterative learning control, stochastic optimization, stochastic systems, and distributed artificial intelligence. He is currently serving on the Editorial Board of *International Journal of Robust and Nonlinear Control* and *IET Cyber-Systems and Robotics*, and Early Career Advisory Board of *IEEE/CAA Journal of Automatica Sinica*.



Samer S. Saab (Senior Member, IEEE) is a Professor of electrical engineering at the Lebanese American University in Byblos, Lebanon. He joined the faculty in 1996 and has since made significant contributions to the university. Dr. Saab served as the Chairperson of the Department of Electrical and Computer Engineering from 2005 to 2012 and later held the position of Associate Dean in the School of Engineering from 2012 to 2013. Since October 2017, he has been fulfilling the role of Dean of Graduate Studies and Research, overseeing the university's commitment to excellence in advanced education and research endeavors.

Prior to his academic career, Dr. Saab gained valuable industry experience at Union Switch and Signal and ABB Daimler-Benz Transportation in Pittsburgh, PA from 1993 to 1996. He holds a B.S., M.S., and Ph.D. in electrical engineering, received in 1988, 1989, and 1992, respectively, from the University of Pittsburgh, USA. Additionally, he obtained an M.A. degree in applied mathematics in 1990 from the same institution. Dr. Saab's research interests encompass various areas, including optimal stochastic control, navigational positioning systems, control of robot manipulators, and managerial policy design.

Throughout his career, Dr. Saab has actively contributed to the academic community. He served on the Editorial Board of the *IEEE Transactions on Control Systems Technology* from 2005 to 2011, the Editorial Board of the IEEE Control Systems Society-Conference from 2005 to 2009, and the Editorial Board of the *IEEE Transactions on Automatic Control* from 2015 to 2022.