International Journal Of
Industrial Electronics Control and Optimization
Print ISSN: 3517-2645
Online ISSN: 3568-2645
Homepage: https://ieco.usb.ac.ir/

A Guaranteed Monotonically Convergent Optimal Iterative Learning Control for Multiple-Input Multiple-Output Continuous-Time Linear Time-Varying Systems

Naser Taghvamanesh | Ali Madady

Department of Electrical Engineering, Tafresh University, Tafresh 39518-79611, Iran.
Corresponding author's email: madadi@tafreshu.ac.ir

Article Info

Article type:
Research Article

Article history:
Received: 08-February-2025
Received in revised form: 26-July-2025
Accepted: 16-August-2025
Published online: 23-Sep-2026

Keywords:
Iterative learning control,
Linear time-varying systems,
Linear quadratic tracking,
Monotonic convergence,
Trajectory tracking.

ABSTRACT

This paper introduces a novel optimal iterative learning control scheme for continuous-time systems with multiple-inputs and multiple-outputs and linear time-varying dynamics. While iterative learning control has been extensively studied in the discrete-time domain, the development of optimal iterative learning control for continuous-time systems remains limited due to the lack of lifted-formulations and associated mathematical challenges. The proposed method transforms the original optimal iterative learning control problem into a linear quadratic tracking-like problem, enabling the derivation of an explicit close-loop control law that ensures both tracking performance and control effort minimization. Unlike many existing approaches that rely on learning algorithms involving derivative terms, which are often sensitive to measurement noise, the proposed design avoids such terms and remains computationally efficient. Moreover, the monotonic convergence of the tracking error and the associated cost function are proved by rigorous mathematical analysis. Theoretical results are supported by four comprehensive simulation examples, including comparisons with several existing iterative learning control methods. Quantitative evaluations confirm that the proposed optimal scheme significantly outperforms previous techniques in terms of convergence speed and error reduction rate. This contribution offers a new framework for the optimal control of continuous-time systems with multiple inputs and outputs in repetitive tasks and provides a foundation for future extensions to constrained, nonlinear, or partially measurable systems.

I. Introduction

Iterative learning control (ILC) is an effective method for managing systems that perform repetitive tasks over a finite time interval [1, 2]. The key feature of ILC is the utilization of information from previous iterations to enhance the performance of the system in subsequent executions [3, 4]. This iterative process allows the system to progressively improve from iteration to iteration, which makes ILC particularly valuable in applications requiring high precision and repeatability, such as robotics, manufacturing, and biomedical fields. Since its introduction [5], ILC has seen a variety of design approaches and has been successfully applied across a wide range of practical scenarios [6, 7, 8].

Notably, various approaches to ILC have been proposed to address different challenges in control systems. For example, in [9], an adaptive repetitive controller is introduced to reject periodic disturbances with an unknown period. In [10], two ILC methods leveraging neural iterative learning identifiers are proposed for a class of unknown time-varying nonlinear systems. The design of ILC for continuous-time linear systems is reformulated as a two-dimensional stabilization problem in [11]. Subsequently, a dynamic output feedback control strategy is developed to effectively resolve this challenge. A data-driven ILC method, integrating predictive control and point-to-point ILC, is proposed in [12] for a class of unknown, repetitive, non-affine nonlinear single-input single-output (SISO) systems. The issue of quantized ILC, which utilizes encoding and decoding mechanisms for networked control systems (NCSs) with bandwidth-constrained communication links, is addressed in [13]. Additionally, a piecewise-type ILC law is designed for linear, discrete, time-varying systems with different initial time points using segment compensation techniques in [14]. Lastly, an error-tracking-driven adaptive back-stepping ILC is proposed in [15] for a type of nonlinear parametric strict-feedback systems performing repetitive tasks over finite intervals.

On the other hand, various optimization-based control strategies, such as genetic algorithms (GA) and particle swarm optimization (PSO), have been widely used to design optimal controllers for complex systems. For instance, a novel approach for the optimal control of robotic manipulators using PSO is presented in [16]. However, these methods fundamentally differ from ILC. While GA and PSO are global optimization techniques that explore the entire parameter space to find an optimal solution, ILC improves control performance through repetitive task execution, refining the control signal based on errors from previous iterations. This makes ILC particularly well-suited for systems performing repetitive tasks, where data from past executions can be utilized to enhance performance over time.

This unique characteristic of ILC has given rise to a category that combines optimization techniques with ILC for controlling systems that perform repetitive tasks. Since an optimal ILC can significantly improve both convergence speed and overall system performance [17], there has been growing interest among researchers in developing optimization-based ILC methods. Below, some of the key studies in this area are outlined.

For instance, in [18], a model-based ILC method is introduced, generalizing ILC using a quadratic performance criterion for time-varying linear constrained systems affected by both deterministic and stochastic disturbances. In [19], a parameter optimization-based ILC approach is presented as a new paradigm for solving the ILC problem in discrete-time linear time-invariant (LTI) systems. A recursive optimal algorithm minimizing the input error covariance matrix is proposed in [20] to determine the optimal forgetting and learning gain matrices for a P-type ILC in linear discrete-time-varying systems with arbitrary relative degrees.

In [21], a feed-forward parameter-optimal high-order ILC algorithm is proposed, noted for its ease of implementation. The extension of the classical proportional-integral-derivative (PID) controller to linear repetitive discrete-time systems is addressed in [22], where an optimal design method is introduced for determining the PID learning gains. Mishra et al. [23] further developed optimization-based ILC schemes for constrained discrete-time LTI systems. In [24], two paradigms of norm-optimal ILC and parameter-optimal ILC are explored for multivariable LTI systems.

Further studies include [25], which established an extended PID-type ILC with four independent learning gains, and [26], which proposed a non-lifted norm-optimal ILC method for linear time-varying (LTV) systems. In [27] and [28], rational basis functions and a novel optimal N-parametric ILC design for discrete-time LTI systems are presented to improve extrapolation properties and design flexibility.

To enhance the convergence rate, [29] introduced an optimal approach for online tuning of PID-type ILC. A robust optimal ILC scheme for updating PID gains in linear systems affected by initial state errors and disturbances is proposed in [30]. The high-performance tracking problem for discrete-time SISO and LTI systems with unknown dynamics is tackled in [31], where two parameter-optimal ILC approaches are introduced, one based on a data-driven algorithm and the other using reinforcement learning.

Zhuang et al. [32] proposed an optimal ILC algorithm for LTI MIMO discrete-time systems with non-uniform trial lengths under input constraints. In [33] and [34], norm-optimal ILC in integer-valued control and optimal ILC for constrained systems with bounded uncertainties are examined. Furthermore, in [35] and [36], an optimal feedforward ILC for discrete-time LTI, SISO systems is developed, combining performance cost and task flexibility. This method outperforms standard norm-optimal ILC in both flexibility and performance.

Finally, in [37] and [38], gradient-like and alternating projection-based optimal ILC designs are presented for LTI continuous-time SISO and MIMO systems, respectively. Meanwhile, in [39] and [40], data-driven tracking control for LTI discrete-time ILC systems and computationally efficient iterative optimization frameworks for nonlinear batch processes are proposed, respectively.

As observed in the aforementioned studies, the majority of optimization-based ILCs have been developed for discrete-time systems. This is primarily due to the fact that, in discrete systems, time is represented as an integer-valued variable, and thus the number of time points per iteration is finite, denoted by M. This enables a compact formulation, known as the lifted-form [17, 24] (or super-vector form [41, 42]), which simplifies the system in the repetition domain by eliminating the time variable and utilizing the Markov parameters of the system. The lifted-form approach reduces the optimal ILC design for discrete-time systems to a parametric and static optimization problem [22, 28, 42], making it solvable.

Despite these advances in discrete-time systems, designing optimal ILCs for continuous-time systems remains an under-explored area. The key challenge stems from the absence of a compact lifted representation in continuous time. Since time is a continuous variable, the infinite-dimensional nature of the signal space makes it difficult to formulate the learning problem as a static optimization problem. As such, only a limited number of studies have addressed optimal ILC in continuous-time settings, and most of them are restricted to scalar systems or assume time-invariant dynamics. One possible approach to overcome this challenge could be to discretize the system and then apply the lifted-form method. However, to achieve sufficient accuracy in the discretization, a very small step size would be required. This would lead to significantly large dimensions for the vectors and matrices in the resulting lifted-form, thereby causing a substantial increase in the computational volume and complexity.

This paper addresses this gap by proposing an optimal ILC framework for continuous-time systems, specifically for MIMO linear time-varying (LTV) systems. To the best of the authors' knowledge, this approach has not been explored in previous research. The closest results to those presented in this paper can be found in Chapter 9 of [42] for time-invariant systems. However, there are fundamental differences, which are detailed in Section 3. In this work, we propose an alternative approach to prove convergence, avoiding the use of complex mathematical concepts such as the properties of self-adjoint operators in Hilbert spaces.

To present our approach more clearly, we propose a novel optimal iterative learning control (OPILC) method for continuous-time systems with MIMO dynamics and time-varying characteristics. This method reformulates the ILC problem into a linear quadratic tracking-like (LQT) problem, facilitating the derivation of an explicit closed-loop control law that guarantees both accurate tracking and minimized control effort. This approach is especially advantageous for applications where monotonic convergence of the tracking error is of paramount importance. In contrast to previously developed continuous-time ILC methods, which often rely on derivative-based learning algorithms [1], [2], [51], that may be susceptible to measurement noise, the proposed method eliminates these terms, thereby ensuring both computational efficiency and robustness. We provide theoretical guarantees for monotonic convergence through a rigorous mathematical analysis, which is validated by extensive simulation results. For a comprehensive comparison, the proposed method is analyzed in relation to existing ILC techniques, such as those presented in [14], [48], and [51], with key distinctions highlighted in terms of convergence speed and error reduction rate.

The main contributions of the proposed optimal ILC in this paper are summarized as follows:

A. Direct Addressing of Continuous-Time Dynamics:

It works directly with continuous-time system dynamics, eliminating the need to discretize the system and apply the lifted-form method. As a result, the computational workload remains low and efficient.

B. Penalizing Error and Input Variation:

To ensure a fair and comprehensive evaluation, our method considers both tracking accuracy and control smoothness. Specifically, the weighting matrices in the cost function, which penalize error and input variation, are adjusted to balance the trade-off between aggressive learning and conservative input usage. This highlights the flexibility of the proposed optimal ILC in addressing performance specifications and physical constraints.

C. Independence from Derivative-Based Learning:

The proposed approach does not rely on derivative-based learning algorithms (D-type learning). While many previously presented continuous-time ILCs are based on D-type learning, which involves differentiating tracking errors or system outputs, such algorithms are highly sensitive to noise. This sensitivity becomes a significant issue in real-world applications, where measurements are inherently noisy.

D. Applicability to Time-Varying Systems:

Unlike the method in [42], which is limited to time-invariant systems, our formulation is designed to handle general time-varying systems.

E. Guaranteed Monotonic Convergence:

We prove that the proposed optimal ILC ensures the monotonic convergence of both the tracking error and the cost function, utilizing only elementary mathematical tools, such as the concepts and properties of the transition matrix for homogeneous linear differential equations. In contrast, the method used in [42] to prove convergence, even for the time-invariant case, involves many mathematical premises, including the theory of self-adjoint operators in Hilbert spaces and the properties of these operators.

F.Computational Efficiency and Ease of Implementation:

In the proposed method, two types of learning gains exist. The first type is a symmetric $n \times n$ matrix, where n represents the order of the controlled system. This matrix is obtained prior to the first iteration by solving a Riccati matrix differential equation, using the system coefficient matrices and the cost function weighting matrices. This learning gain is then used throughout all iterations. The second type of learning gain is a vector with n components, which needs to be determined by solving a linear vector differential equation, using the tracking error obtained from the previous iteration. This calculation is done before the start of each new iteration and when the system is being reset. Thus, all learning gains are determined either before the first iteration or during the intermediate phases between iterations. In other words, no algebraic or differential equations need to be solved during the iterations themselves. This characteristic significantly enhances computational efficiency and simplifies the practical implementation of the proposed method.

G. Superior Performance:

Extensive simulation results, including comparisons with the ILC methods in [14], [48], and [51], show that the proposed optimal ILC achieves faster convergence and higher tracking accuracy, while maintaining an acceptable computational cost.

The rest of this paper is organized as follows. Section II formulates the OPILC problem for continuous-time LTV systems and defines the mathematical assumptions. Section III derives the optimal control input using LQT theory and presents the resulting OPILC structure. Section IV rigorously proves the monotonic convergence of the tracking error and cost function. Section V provides detailed simulation results and performance comparisons with existing ILC methods. Section VI concludes the paper and discusses future extensions such as incorporating input/state constraints, uncertainty, or nonlinear dynamics.

II. Problem Setup

Consider the continuous-time MIMO LTV system that performs a repetitive operation:

$$\begin{cases} \dot{x}_i(t) = A(t)x_i(t) + B(t)u_i(t) + \eta_x(t) \\ y_i(t) = C(t)x_i(t) + \eta_y(t) \\ 0 \le t \le t_f i = 0,1,\dots \end{cases}, \tag{1}$$

where t denotes the time variable, i is the operation or iteration index, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, and $y \in \mathbb{R}^q$ are the system sate, input, and output vectors, respectively. $\eta_x \in \mathbb{R}^n$ and $\eta_y \in \mathbb{R}^q$ are the unknown but bounded deterministic disturbances, or the effects of the un-modeled dynamics of the system, which impact the state and output vectors of the system, respectively. t_f is a constant and specified, indicating the time duration of iterations. $A(t)$, $B(t)$, and $C(t)$ are the real-valued coefficients with appropriate dimensions. Moreover, the superimposed dot denotes the time derivative.

Consider system (1) and make the following reasonable assumptions:

A1) Every entry of $B(t)$ is a continuously differentiable function in the interval $t \in (0, t_f)$.

A2) All entries of $A(t)$, $B(t)$, $C(t)$, and $\dot{B}(t)$ are bounded and continuous functions in the interval $t \in [0, t_f]$.

A3) $C(t)B(t)$ has full row rank in the interval $t \in [0, t_f]$, that is:

$$\mathrm{rank}(C(t)B(t)) = q \forall t \in [0, t_f]. \tag{2}$$

A4) The initial condition $x_i(0)$ remains constant in all iterations, i.e.

$$x_i(0) = x_0 i = 0,1, \ldots \tag{3}$$

Remark 1. Assumptions A1 and A2 have been considered in order to guarantee the existence and uniqueness of the solution to the homogeneous differential equations that will be encountered later.

Remark 2. Assumption A3 is a standard assumption in ILC design, which guarantees the existence of learning gains [14, 29, 32, 34, 47]. It is clear that a necessary condition for assumption A3 to hold is that the number of system inputs must not be less than the number of outputs ($p \geq q$). Since the goal is to independently control all system outputs through appropriate selection of the inputs, the number of inputs must be greater than or at least equal to the number of outputs.

Remark 3. ILC systems usually have the same initial states x_0 for each working cycle [34, 42, 46, 47]. Thus, assumption A4 is not a restriction. This assumption is required for the perfect convergence of any type of ILC. Clearly, if the initial conditions of the system change randomly, it will be impossible to achieve perfect convergence because the controller encounters new conditions in each iteration, and therefore, learning and using past experiences to improve the performance becomes ineffective.

Let us define the problem of optimal iterative learning control (OPILC) as follows:

Consider system (1) and suppose that operates in a repetitive mode in the finite continuous time interval $t \in [0, t_f]$ and assumptions A1-A4 hold. A desired output trajectory $y_d(t) \in \mathbb{R}^q$ is given. The error between the system output and $y_d(t)$ at iteration i is defined as follows:

$$e_i(t) = y_d(t) - y_i(t) i = 0,1,2, \ldots \tag{4}$$

Suppose that we are at the start of the current iteration i, and all the system signals in the previous iteration $i-1$, such as $u_{i-1}(t), x_{i-1}(t)$ and $e_{i-1}(t)$, are known and stored in the controller memory. Determine the system input in the current iteration i, that is $u_i(t)$, so that the system output approaches $y_d(t)$ by minimizing the following quadratic cost function:

$$J_i = \frac{1}{2}\{\|e_i(.)\|_Q^2 + \|v_i(.)\|_R^2\} i = 1,2, \ldots \tag{5}$$

where

$$v_i(t) = u_i(t) - u_{i-1}(t), \tag{6}$$

and

$$\|e_i(.)\|_Q = \left[\int_0^{t_f} e_i^T(t) Q(t) e_i(t) dt\right]^{\frac{1}{2}},$$

$$\|v_i(.)\|_R = \left[\int_0^{t_f} v_i^T(t) R(t) v_i(t) dt\right]^{\frac{1}{2}}, \tag{7}$$

and T denotes the transpose, $Q(t) \in \mathbb{R}^{q \times q}$ and $R(t) \in \mathbb{R}^{p \times p}$ are the weight symmetric and positive definite matrices whose entries are the continuous functions of $t \in [0, t_f]$.

The physical interpretation of the given cost function (5) is that we want to bring the output of the system close to the desired output trajectory, and at the same time, the input of the system does not differ much from the previous iteration. The relative degrees of these two opposing goals is absorbed in matrices $Q(t)$ and $R(t)$.

III. Problem Solution

Let us define the vectors $z_i(t)$ and $w_i(t)$ as follows:

$$\begin{cases} z_i(t) = x_i(t) - x_{i-1}(t) \\ w_i(t) = y_i(t) - y_{i-1}(t). \\ 0 \leq t \leq t_f i = 1,2, \ldots \end{cases} \tag{8}$$

Differentiating $z_i(t)$ with respect to t yields:

$$\begin{cases} \dot{z}_i(t) = \dot{x}_i(t) - \dot{x}_{i-1}(t) \\ w_i(t) = y_i(t) - y_{i-1}(t). \\ 0 \leq t \leq t_f i = 1,2, \ldots \end{cases}$$

By substituting $\dot{x}_{i-1}(t)$, $\dot{x}_i(t)$, $y_{i-1}(t)$ and $y_i(t)$ from (1) into the above equations, one obtains:

$$\begin{cases} \dot{z}_i(t) = A(t)x_i(t) + B(t)u_i(t) + \eta_x(t) - A(t)x_{i-1}(t) \\ \qquad -B(t)u_{i-1}(t) - \eta_x(t) \\ \qquad = A(t)(x_i(t) - x_{i-1}(t)) + B(t)(u_i(t) - u_{i-1}(t)) \\ w_i(t) = y_i(t) - y_{i-1}(t) = C(t)x_i(t) + \eta_y(t) \\ \qquad -C(t)x_{i-1}(t) - \eta_y(t) \\ \qquad = C(t)(x_i(t) - x_{i-1}(t)) \end{cases}.$$

Inserting $x_i(t) - x_{i-1}(t)$ and $u_i(t) - u_{i-1}(t)$ from (8) and (6), respectively, into the above equations, yields:

$$\begin{cases} \dot{z}_i(t) = A(t)z_i(t) + B(t)v_i(t) \\ w_i(t) = C(t)z_i(t) \end{cases}.$$

$$0 \leq t \leq t_f, i = 1,2, \ldots \tag{9}$$

On the other hand, substituting $y_i(t)$ from (8) into (4) gives:

$e_i(t) = y_d(t) - y_{i-1}(t) - w_i(t),$
that is:

$$e_i(t) = r_i(t) - w_i(t) i = 1,2,\ldots \quad (10)$$

where

$$r_i(t) = e_{i-1}(t). \quad (11)$$

Substituting $e_i(t)$ from (10) into (5) yields:

$$J_i = \frac{1}{2}\int_0^{t_f} \{[r_i(t) - w_i(t)]^T Q(t)[r_i(t) - w_i(t)] + v_i^T(t)R(t)v_i(t)\} dt.$$
$$i = 1,2,\ldots \quad (12)$$

Now, let us consider relation (9). This relation represents a hypothetical linear time-varying system, where $z_i(t)$, $v_i(t)$, and $w_i(t)$ are its state, input, and output vectors, respectively. $r_i(t)$ is the given desired output for this hypothetical system, and minimizing cost function (12) for this hypothetical system implies addressing a problem similar to the linear quadratic tracking (LQT) problem.

A review of the LQT problem and its solution procedure is provided in the Appendix. Based on this solution, and utilizing equation (A-3), the proposed OPILC problem is solved as follows:

$$v_i(t) = -R^{-1}(t)B^T(t)[K(t)z_i(t) - g_i(t)].$$
$$0 \le t \le t_f, i = 1,2,\ldots \quad (13)$$

and the optimum value of cost function (12) is obtained as (see (A-8)):

$$J_i^* = \frac{1}{2} z_i^T(0)K(0)z_i(0) - z_i^T(0)g_i(0) + \phi_i(0).$$
$$i = 1,2,\ldots \quad (14)$$

where $K(t) \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix that is the solution of the following Riccati matrix differential equation (see (A-4)):

$$\dot{K}(t) = -K(t)A(t) - A^T(t)K(t) + K(t)B(t)R^{-1}(t)B^T(t)K(t) - C^T(t)Q(t)C(t). \quad (15)$$

and $g_i(t) \in \mathbb{R}^n$ is a vector that is the solution of the following linear vector differential equation (see (A-6)):

$$\dot{g}_i(t) = -[A(t) - B(t)R^{-1}(t)B^T(t)K(t)]^T g_i(t) - C^T(t)Q(t)e_{i-1}(t).$$
$$i = 1,2,\ldots \quad (16)$$

The third term in the right-hand side of (14), i.e., $\phi_i(0)$, is obtained from the backward integration of the following scalar differential equation (see (A-9)):

$$\dot{\phi}_i(t) = -\frac{1}{2} e_{i-1}^T(t)Q(t)e_{i-1}(t) + \frac{1}{2} g_i^T(t)B(t)R^{-1}(t)B^T(t)g_i(t).$$
$$i = 1,2,\ldots \quad (17)$$

By comparing (12) with (A-2), it is observed that the matrix F in the cost function (12) is zero. Therefore, from (A-5), (A-7), and (A-10), it follows that the terminal conditions of the differential equations (15), (16), and (17) are all zero, that is

$$K(t_f) = 0, g_i(t_f) = 0, \phi_i(t_f) = 0. \quad (18)$$

Using (6) and (13), the input to system (1) at the i -th iteration is expressed in terms of its input at the previous iteration and other available information as follows

$$u_i(t) = u_{i-1}(t) - R^{-1}(t)B^T(t)\{K(t)[x_i(t) - x_{i-1}(t)] - g_i(t)\}.$$
$$0 \le t \le t_f, i = 1,2,\ldots \quad (19)$$

Remark 4. Equation (15) is independent of the iterations. Therefore, it is solved only once prior to the first iteration, and the resulting gain $K(t)$ is used in all subsequent iterations. It should be noted that although matrix $K(t)$ has n^2 components, its symmetry implies that Riccati equation (15) represents a system of $\frac{n(n+1)}{2}$ first-order differential equations instead of n^2 equations.

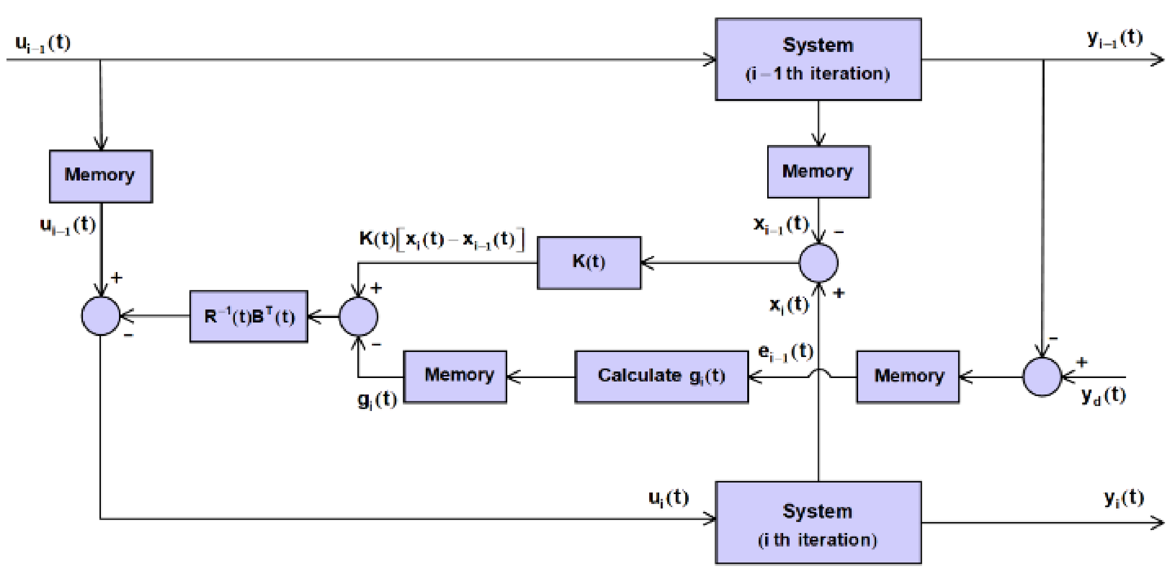

Fig. 1. The block diagram of the proposed OPILC.

Remark 5. The existence of $e_{i-1}(t)$ in equation (16) makes it an iteration-dependent equation. Therefore, it is necessary to re-compute $g_i(t)$ for each iteration. Given $e_{i-1}(t)$, one can compute $g_i(t)$ off-line in the time between iteration i and $i-1$, when the system is being reset. After this time, $e_{i-1}(t)$ can be removed from the memory.

Considering (19), the obtained OPILC can be illustrated as Fig. 1

As mentioned in Section I, in [42] Algorithm 9.5, one can find results similar to those of (15), (16), and (19) for the time-invariant case. The main differences between these two approaches can be summarized as follows:

1) The results given in Algorithm 9.5 of [42] are based on time-invariant self-adjoint operators acting on a Hilbert space. These assumptions ensure the applicability of classical spectral theory and operator analysis. However, such results are not directly extendable to time-varying operators, whose domains, boundedness, or self-adjointness properties may evolve over time. Consequently, the results of Algorithm 9.5 in [42] cannot be applied directly to time-varying systems. In contrast, our approach is fundamentally different from that of [42] and has been specifically developed from the outset to handle time-varying systems.

2) The method of proving convergence in [42], even for the time-invariant case, involves many mathematical premises, including the theory of self-adjoint operators in Hilbert spaces and the properties of these operators (see Theorems 9.1, 9.2, and 9.3 in [42]). However, our scheme to prove convergence is completely different from the method used in [42] and only relations (14), (16), and (17) and the concept of the transition matrix of the homogeneous linear differential equation are used in it (see Theorems 1, 2, and 3 in Section IV).

IV. Convergence Analysis

One of the desired features in the ILC design is to achieve convergence, and particularly monotonic convergence. Convergence implies that the tracking error converges to zero as the number of iterations increases. That is:

$$\lim_{i\to\infty} e_i(t) = 0 \forall t \in (0, t_f], \tag{20}$$

where $e_i(t)$ is the tracking error defined in (4).

The concept of monotonic convergence is that the norm of the tracking error in each iteration is smaller than the previous iteration. It means to have:

$$\begin{cases} \|e_i(.)\| < \|e_{i-1}(.)\| \text{if} \|e_{i-1}(.)\| \neq 0 \\ \|e_i(.)\| = \|e_{i-1}(.)\| = 0 \text{if} \|e_{i-1}(.)\| = 0 \end{cases} \forall i = 1,2,\dots \tag{21}$$

where $\|e_i(.)\|$ is a chosen norm for $e_i(t)$.

Monotonic convergence is a stronger condition than general convergence, as the latter can be deduced from the former. However, the reverse does not hold in the general case.

For the convergence analysis of the proposed OPILC, the following theorem is required.

Theorem 1. For any $i = 1,2,\dots$ the following three statements are equivalent:

$$e_{i-1}(t) = 0 \forall t \in (0, t_f]. \tag{22}$$

$$B^T(t) g_i(t) = 0 \forall t \in [0, t_f]. \tag{23}$$

$$v_i(t) = 0 \forall t \in [0, t_f]. \tag{24}$$

Proof. (22)⇔ (23): Suppose (22) holds. Then, substituting $e_{i-1}(t) = 0$ into (16) yields:

$$\dot{g}_i(t) = F_1(t) g_i(t), \tag{25}$$

where

$$F_1(t) = -[A(t) - B(t) R^{-1}(t) B^T(t) K(t)]^T. \tag{26}$$

Since all of the entries of $F_1(t)$ are bounded and continuous functions in the interval $t \in [0, t_f]$, for any given terminal condition $g_i(t_f)$, homogenous equation (25) has a unique solution in the interval $t \in [0, t_f]$, and this solution is obtained as follows:

$$g_i(t) = \Phi_1(t, t_f) g_i(t_f) \forall t \in [0, t_f], \tag{27}$$

where $\Phi_1(.,.)$is the state transition matrix of equation (25).

By applying the terminal condition $g_i(t_f) = 0$, as given in (18), to (27), we obtain

$$g_i(t) = 0 \forall t \in [0, t_f]. \tag{28}$$

Left-multiplying both sides of (28) by $B^T(t)$ yields relation (23).

Conversely, assume that equation (23) holds. Then, using (16), we obtain the following:

$$\dot{g}_i(t) = -A^T(t) g_i(t) - C^T(t) Q(t) e_{i-1}(t). \tag{29}$$

Left-multiplying both sides of (29) by $B^T(t)$ gives:

$$B^T(t)\dot{g}_i(t) = -B^T(t)A^T(t)g_i(t) - B^T(t)C^T(t)Q(t)e_{i-1}(t). \tag{30}$$

We have:

$$\frac{d}{dt}[\underbrace{B^T(t) g_i(t)}_{0}] = \dot{B}^T(t) g_i(t) + B^T(t)\dot{g}_i(t) = 0. \tag{31}$$

Thus,

$$B^T(t)\dot{g}_i(t) = -\dot{B}^T(t) g_i(t). \tag{32}$$

Combining (32) with (30) yields the following result:

$$[C(t)B(t)]^T Q(t) e_{i-1}(t) = \left[\dot{B}^T(t) - B^T(t)A^T(t)\right] g_i(t). \tag{33}$$

By left-multiplying both sides of the above equation by $C(t)B(t)$, the following relation is obtained:

$$C(t)B(t)[C(t)B(t)]^T Q(t) e_{i-1}(t) = C(t)B(t)\left[\dot{B}^T(t) - B^T(t)A^T(t)\right] g_i(t). \tag{34}$$

Since, according to Assumption A3, $C(t)B(t)$ has full row rank, it follows that $C(t)B(t)[C(t)B(t)]^T$ is invertible.

Multiplying both sides of equation (34) from the left by $Q^{-1}(t)(C(t)B(t)[C(t)B(t)]^T)^{-1}$ yields:

$$e_{i-1}(t) = Q^{-1}(t)(C(t)B(t)[C(t)B(t)]^T)^{-1} \times C(t)B(t)[\dot{B}^T(t) - B^T(t)A^T(t)]g_i(t). \quad (35)$$

Substituting $e_{i-1}(t)$ from (35) into (29) yields:

$$\dot{g}_i(t) = F_2(t)g_i(t). \quad (36)$$

where

$$F_2(t) = -\{A(t) + [\dot{B}(t) - A(t)B(t)](C(t)B(t))^T \times (C(t)B(t)[C(t)B(t)]^T)^{-1}C(t)\}^T. \quad (37)$$

According to Assumption A2, all entries of $A(t), B(t), C(t)$ and $\dot{B}(t)$ are continuous and bounded functions in the interval $t \in [0, t_f]$. It then follows from (37) that all entries of $F_2(t)$ are also continuous and bounded. As a result, for any given terminal condition $g_i(t_f)$, equation (36) has a unique solution in the interval $t \in [0, t_f]$, and this solution is obtained as follows:

$$g_i(t) = \Phi_2(t, t_f)g_i(t_f)\forall t \in [0, t_f], \quad (38)$$

where $\Phi_2(.,.)$is the state transition matrix of equation (36).

Applying the terminal condition $g_i(t_f) = 0$, as given in (18), to (38) gives (28). Then substituting $g_i(t)$ from (28) into (35) yields (22).

(23)⇔ (24): Suppose (23) holds. Then, substituting $B^T(t)g_i(t) = 0$ into (13) yields:

$$v_i(t) = -R^{-1}(t)B^T(t)K(t)z_i(t). \quad (39)$$

By substituting $v_i(t)$ from (39) into (9), the following homogeneous equation is obtained:

$$\dot{z}_i(t) = F_3(t)z_i(t), \quad (40)$$

where

$$F_3(t) = A(t) - B(t)R^{-1}(t)B^T(t)K(t). \quad (41)$$

All of the entries of $F_3(t)$ are continuous and bounded functions in the interval $t \in [0, t_f]$. Hence, for any given initial condition $z_i(0)$, equation (40) has a unique solution in the interval $t \in [0, t_f]$, as follows:

$$z_i(t) = \Phi_3(t, 0)z_i(0)\forall t \in [0, t_f], \quad (42)$$

where $\Phi_3(.,.)$is the state transition matrix of equation (40).

From (3) together with the definition of $z_i(t)$ in (8), it follows that:

$$z_i(0) = 0 i = 1,2,\ldots \quad (43)$$

Substituting $z_i(0)$ from (43) into (42) yields the following result:

$$z_i(t) = 0\forall t \in [0, t_f]. \quad (44)$$

By substituting $z_i(t)$ from (44) into (39), one can obtain (24).

Conversely, if (24) holds, then from (9) and (43), we can obtain (44). Finally, substituting $v_i(t)$ and $z_i(t)$ from (24) and (44), respectively, into (13) yields (23).

Now, we are in a position that we can analyze the convergence properties of the proposed OPILC.

Theorem 2. The proposed OPILC guarantees monotonic convergence in the sense of norm $\|e_i(.)\|_Q$ defined in (7).

Proof. Substituting $z_i(0)$ from (43) into (14) yields that the optimum value of cost function (12) is as follows:

$$J_i^* = \phi_i(0) i = 1,2,\ldots \quad (45)$$

Integrating both sides of equation (17) over the interval $[0, t_f]$ gives:

$$\phi_i(t_f) - \phi_i(0) = -\frac{1}{2}\|e_{i-1}(.)\|_Q^2 + \frac{1}{2}\int_0^{t_f} g_i^T(t)B(t)R^{-1}(t)B^T(t)g_i(t)dt.$$

By inserting $\phi_i(t_f)$ and $\phi_i(0)$ from (18) and (45), respectively, into the above equations, the following expression is obtained:

$$J_i^* = \frac{1}{2}\|e_{i-1}(.)\|_Q^2 - \frac{1}{2}\int_0^{t_f} g_i^T(t)B(t)R^{-1}(t)B^T(t)g_i(t)dt. \quad (46)$$

On the other hand, based on (7), (10) and (12), J_i^* has the following form:

$$J_i^* = \frac{1}{2}\{\|e_i(.)\|_Q^2 + \|v_i(.)\|_R^2\}. \quad (47)$$

From (46) and (47) one gets:

$$\frac{1}{2}\|e_i(.)\|_Q^2 = \frac{1}{2}\|e_{i-1}(.)\|_Q^2 - \frac{1}{2}\int_0^{t_f}\{v_i^T(t)R(t)v_i(t) + g_i^T(t)B(t)R^{-1}(t)B^T(t)g_i(t)\}dt. \quad (48)$$

Since $R(t)$ and, consequently, $R^{-1}(t)$ are symmetric positive definite matrices for all $t \in [0, t_f]$, equation (48) yields the following result:

$$\|e_i(.)\|_Q \leq \|e_{i-1}(.)\|_Q i = 1,2,\ldots \quad (49)$$

and (49) holds as an equality if and only if one has:

$$v_i(t) = 0\forall t \in [0, t_f], \quad (50)$$

and:

$$B^T(t)g_i(t) = 0\forall t \in [0, t_f]. \quad (51)$$

Taking Theorem 1 into account, each of the relations (50) and (51) is equivalent to (22).

Therefore, the proposed OPILC is convergent, and this convergence is monotonic in the sense of norm $\|e_i(.)\|_Q$.

Theorem 3. The proposed OPILC guarantees monotonic convergence in the sense of cost function (5).

Proof. By adding and subtracting $\frac{1}{2}\|v_{i-1}(.)\|_R^2$ to the right-hand side of (46), one obtains:

$$J_i^* = J_{i-1}^* - \frac{1}{2}\int_0^{t_f}\{v_{i-1}^T(t)R(t)v_{i-1}(t) + g_i^T(t)B(t)R^{-1}(t)B^T(t)g_i(t)\}dt. \quad i = 2,3,\ldots \quad (52)$$

Since $R(t)$, and therefore, $R^{-1}(t)$ are symmetric positive definite for all $t \in [0, t_f]$, (52) yields:

$$J_i^* \leq J_{i-1}^* i = 2,3,\dots \tag{53}$$

and (53) holds as an equality if and only if we have:

$$v_{i-1}(t) = 0, B^T(t) g_i(t) = 0.$$
$$\forall t \in [0, t_f], i = 2,3,\dots \tag{54}$$

According to Theorem 1, (54) is equivalent to:

$$e_{i-2}(t) = e_{i-1}(t) = 0 \forall t \in (0, t_f], i = 2,3,\dots \tag{55}$$

Consequently, the proposed OPILC is convergent, and this convergence is monotonic in the sense of cost function (5).

V. Illustrative Examples and Comparison with Other ILCs

In this section, four simulation examples are provided to demonstrate the applicability and performance of the proposed OPILC.

All simulations were performed on a computer with a 12th Gen Intel® Core™ i3-12100 CPU @ 3.30 GHz, 16.0 GB of RAM, running Windows 10 Pro (version 22H2). MATLAB R2016b was used for implementation and numerical computations.

Example 1. Consider a second-order repetitive system in the form of (1), where the coefficient matrices, time duration of iterations, and disturbances $\eta_x(t)$, $\eta_y(t)$ are given as follows:

$$A(t) = \begin{bmatrix} -t & 1 \\ 2\,sin(\pi t) & -\mathrm{Ln}(t+1) \end{bmatrix}, B(t) = \begin{bmatrix} e^t \\ sinh(t) \end{bmatrix},$$
$$C(t) = [0.5 \quad t^2], t_f = 1,$$
$$\eta_x(t) = \begin{bmatrix} e^{sin(4\pi t)} \\ t\,cos(2\pi t) \end{bmatrix}, \eta_y(t) = 0.05(t - 0.5).$$

To perform the simulation, the initial condition of the system and its input at the initial iteration $i = 0$ are set to be zero. That is:

$$x_i(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{for} i = 0,1,\dots, u_0(t) = 0 \text{for} 0 \leq t \leq 1.$$

The desired output trajectory $y_d(t)$ is chosen as:

$$y_d(t) = sin(60t - 92t^2 + 48t^3 - 8t^4) \text{ for} 0 \leq t \leq 1.$$

In order to compare the performance of the method proposed in this paper with the approach presented in [48], the simulation of this example is conducted in two cases. In these cases, the method proposed in this paper and the approach presented in [48] are used, respectively.

Case 1. Using the OPILC proposed in this paper

The symmetric positive definite matrices $Q(t)$ and $R(t)$ in cost function (5), are chosen as follows:
$Q(t) = 10^4, R(t) = 1.$

The time response of the system output, along with the desired output trajectory, is depicted in Fig. 2. This figure indicates that the system output successfully converges to the desired output trajectory by increasing the number of iterations.

The values of $\frac{1}{2}\|e_i(.)\|_Q^2$ and cost function J_i versus the number of iterations are given in Fig. 3 (in the log scale). This figure demonstrates that the convergence is monotonic with respect to both $\|e_i(.)\|_Q$ and J_i, which are consistent with Theorems 2 and 3.

A key aspect of iterative learning control is how the control input evolves over successive trials. Figure 4 illustrates the progression of the input profiles across iterations. It can be observed that the input converges to a particular signal that was not known a priori and only emerged as a result of the learning process.

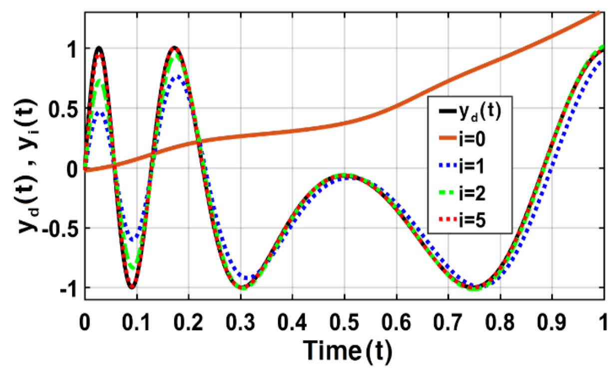

Fig. 2. The output trajectories $y_d(t)$ and $y_i(t)(i = 0,1,2,5)$ for Example 1 (Case 1).

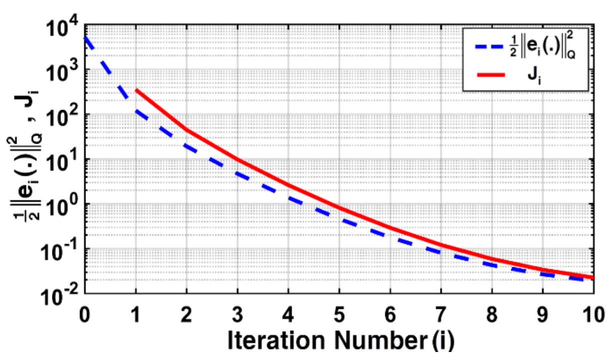

Fig. 3. The monotonic convergence of $\|e_i(.)\|_Q$ and J_i with respect to i for Example 1 (Case 1).

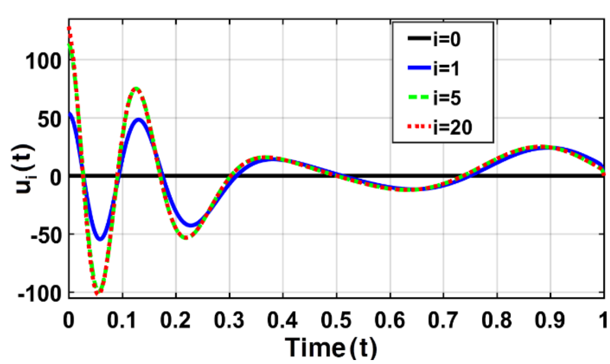

Fig. 4. The system input at iterations $i = 0,1,5,20$ for Example 1 (Case 1).

Case 2. Using the ILC presented in [48]

First, we provide a brief review the method given in [48]. In [48], the system under control is discrete-time, MIMO, and LTV, which is represented by:

$$\begin{cases} x_i(k+1) = A_D(k)x_i(k) + B_D(k)u_i(k) \\ y_i(k) = C_D(k)x_i(k) \\ k = 0,1,\ldots,N i = 0,1,\ldots \end{cases}. \quad (56)$$

and the ILC problem is defined as follows:

Given System (56) with boundary conditions $x_i(0) = x_0$ $(i \geq 0)$ and $u_0(k)(0 \leq k \leq N-1)$, and reference output trajectory $y_d(k)(1 \leq k \leq N)$, find an appropriate control input $u_i(k)$ $(0 \leq k \leq N-1$ and $i \geq 1)$ such that the system output follows the reference trajectory.

To solve this problem, the following ILC is presented by [48]:

$$\begin{matrix} u_{i+1}(k) = u_i(k) + F(k+1)e_i(k+1) \\ k = 0,1,\ldots,N-1; i = 0,1,\ldots \end{matrix}. \quad (57)$$

where $F(k+1)$ is a time-varying learning gain matrix with appropriate dimensions, and $e_i(k+1)$ shows the tracking error, which is:

$e_i(k+1) = y_d(k+1) - y_i(k+1)$.

Using two-dimensional (2-D) system theory, it has been shown in [48] that if the gain matrix $F(k)$ is chosen such that the following relation holds:

$$\rho\{I - C_D(k)B_D(k-1)F(k)\} < 1, \\ k = 1,2,\ldots,N. \quad (58)$$

where I is the identity matrix and $\rho\{.\}$ represents the spectral radius of the matrix.

Then ILC (57) guarantees:

$$\lim_{i\to\infty} e_i(k) = 0 \text{for} k = 1,2,\ldots,N. \quad (59)$$

Moreover, it has been shown that there exists a gain matrix $F(k)$ that yields (58) only if matrix $C_D(k)B_D(k-1)$ has full-row rank, and in this case, condition (58) can be achieved by calculating $F(k)$ as follows [48]:

$$F(k) = (C_D(k)B_D(k-1))^T\{C_D(k)B_D(k-1) \\ \times (C_D(k)B_D(k-1))^T\}^{-1}(I - \Phi(k)), \\ \text{for} k = 1,2,\ldots,N. \quad (60)$$

where $\Phi(k)$ is an arbitrary matrix with appropriate dimensions that satisfies the condition $\rho\{\Phi(k)\} < 1$ for $k = 1,2,\ldots,N$.

When system (56) is SISO, then $C_D(k)B_D(k-1)$, $\Phi(k)$, and $F(k)$ are scalars, and consequently, formula (60) simplifies to:

$$F(k) = \frac{1-\Phi(k)}{C_D(k)B_D(k-1)} \text{for} k = 1,2,\ldots,N. \quad (61)$$

and condition $\rho\{\Phi(k)\} < 1$ is reduced to $|\Phi(k)| < 1$ for $k = 1,2,\ldots,N$.

It should be noted that the above ILC is given for discrete-time systems, whereas this study focuses on continuous-time system. For this reason, first, the system given in Example 1 is discretized by choosing the sampling time $\Delta t = 0.001$. Therefore, we have:

$$N = \frac{t_f}{\Delta t} = 10^3.$$

In addition, since the system is SISO, (61) is used to calculate the learning gain $F(k)$. To have the maximum possible convergence rate, $\Phi(k)$ is chosen as $\Phi(k) = 0.9975$ (for $k = 1,2,\ldots,10^3$) through trial and error.

The trajectories obtained for the system output from the simulation in some iterations are shown in Fig. 5. Although according to this figure, the output of the system is closer to the desired trajectory with the increase in the number of iterations, by comparing this figure with Fig. 2, it is clearly inferred that the convergence rate in Fig. 5 is much slower than in Fig. 2. To evaluate this more precisely, let us use the following total square error of tracking:

$$EE(i) = \sum_{k=0}^{N} e_i^2(k) = \sum_{k=0}^{1000} e_i^2(k). \quad (62)$$

Figure 6 shows the total squared error of tracking (in the log scale) for both cases 1 and 2. It can be observed that the convergence rate of the OPILC proposed in this paper is much faster than that in the method presented in [48].

To enable a more comprehensive comparison of the performance of the OPILC proposed in this paper with other ILCs, we employ quantitative performance metrics including error reduction rate, convergence speed, and computational complexity. For this purpose, let $I_{n\%}$ (for $n = 50,10,1$) denote the minimum number of iterations required for the error to become less than or equal to n percent of its initial value (i.e., the error at iteration zero), as a measure of the error reduction rate. Similarly, let $I_{10^{-3}}$ represent the minimum number of iterations required for the absolute value of error to reach or fall below 10^{-3}, serving as a measure of convergence speed. To assess computational complexity, T_1 is used to denote the time required to execute a single iteration on the hardware platform specified at the beginning of this section.

The results obtained from the comparison of Cases 1 and 2 for Example 1 are presented in Table I. It should be noted that the term "error" in this table refers to the total square error as defined in equation (62). According to this table, the error reduction rate and convergence speed of the proposed method are significantly higher than those of the ILC method presented in [48]. Specifically, the error is reduced by 90% after just one iteration using the proposed method, whereas the ILC in [48] requires 710 iterations to achieve the same reduction. Moreover, the convergence speed of the proposed method is over 75 times faster than that of the ILC in [48]. Although the proposed scheme requires approximately 3.8225 times more computational time per iteration compared to the ILC method presented in [48], as will be clarified in Remark 6, this increase does not introduce any practical impediments.

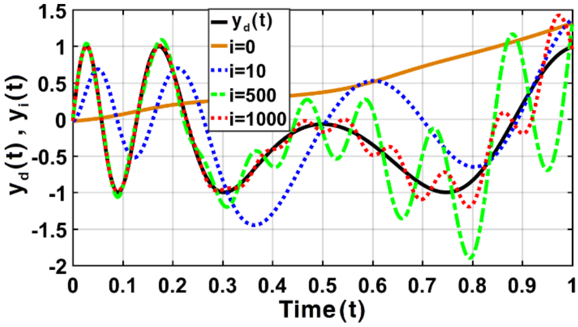

Fig. 5. The output trajectories $y_d(t)$ and $y_i(t)$ $(i = 0,10,500,1000)$ for Example 1 (Case 2).

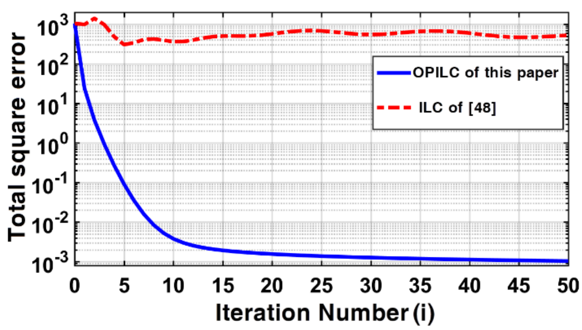

Fig. 6. The total square errors of tracking with respect to i for Example 1 (comparison of Cases 1 and 2).

TABLE I. Comparison of Cases 1 and 2 for Example 1

Method		ILC of [48]	OPILC proposed in this paper
Error Reduction Rate	$I_{50\%}$	285	1
	$I_{10\%}$	710	1
	$I_{1\%}$	1220	2
Convergence Speed	$I_{10^{-3}}$	4150	55
Computational Complexity	T_1 (sec.)	0.012535	0.047915

Example 2. The rotation dynamics of a rigid body in space can be described by Euler equations in the principal coordinates as follows [49, 50]:

$$\begin{cases} J_x\dot{\omega}_x = (J_y - J_z)\omega_y\omega_z + u_x \\ J_y\dot{\omega}_y = (J_z - J_x)\omega_z\omega_x + u_y. \\ J_z\dot{\omega}_z = (J_x - J_y)\omega_x\omega_y + u_z \end{cases} \tag{63}$$

where ω_x, ω_y, and ω_z are the components of the angular velocity vector ω along the principal axes; u_x, u_y, and u_z are the torque inputs applied about the principal axes; and J_x,J_y, and J_z are the principal moments of inertia.

Assuming that the body is axial-symmetric, i.e., $J_x = J_y = J$, rotational dynamics along the x and y axes are considered, under the assumption that $\omega_z(t)$ is a known function of time. This yields an LTV system of the form

$$\begin{cases} \dot{x} = \underbrace{\begin{bmatrix} 0 & \Omega(t) \\ -\Omega(t) & 0 \end{bmatrix}}_{A(t)} x + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{B(t)} u \\ y = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C(t)} x \end{cases}. \tag{64}$$

where

$$x = \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}, y = \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}, u = \frac{1}{J}\begin{bmatrix} u_x \\ u_y \end{bmatrix}, \Omega(t) = \frac{(J-J_z)}{J}\omega_z(t). \tag{65}$$

Let us consider model (64). This is a two input-two output (TITO) system. It is desired to control this system iteratively in a given time interval $[0, t_f]$.

For simulations, we choose $t_f = 2$ and $\Omega(t) = t\sin(2\pi t)$. Moreover, the desired output trajectories are selected as follows:

$$\begin{aligned} y_d^{(1)}(t) &= \tanh\left[20\cos(t)\right]\sin(t^2), \\ y_d^{(2)}(t) &= 10te^{-2t}\cos(\pi\sqrt{t}). \end{aligned} \tag{66}$$

The initial conditions and the initial input of the system (the input of the system in iteration $i = 0$) are all chosen to be zero.

In order to compare the performance of the OPILC proposed in this paper with the methods presented in [51] and [14], the simulation of this example is conducted in three cases. In these cases, the procedure proposed in this paper and the ILCs presented in [51] and [14] are used, respectively.

Case 1. Using the OPILC proposed in this paper

The symmetric positive definite matrices $Q(t)$ and $R(t)$ in cost function (5) are chosen as follows:
$Q(t) = 10^4 I, R(t) = I.$

The trajectories obtained for the system outputs from the simulation at some iterations, along with the desired outputs trajectories, are shown in Figs. 7a and 7b. These figures indicate that by increasing the iteration number, the outputs quickly converge to the given desired outputs trajectories, such that at iteration $i = 1$, the system outputs exactly follow the desired outputs trajectories.

Furthermore, Fig. 8 shows $\frac{1}{2}\|e_i(.)\|_Q^2$ and J_i (in the log scale) versus the number of iterations. According to this figure, both $\|e_i(.)\|_Q$ and J_i converge to zero with a monotonic convergence rate, which confirms Theorems 2 and 3.

An essential feature of ILC lies in the evolution of the control input over repeated trials. As shown in Figs. 9a and 9b, the system inputs profiles gradually adjust through iterations and eventually converge to two consistent patterns that are not predefined but arise from the learning dynamics.

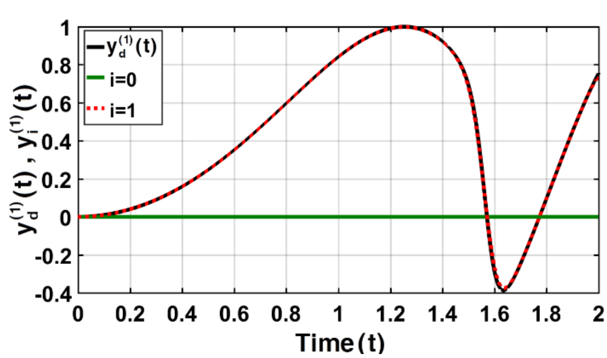

Fig. 7a. The output trajectories $y_d^{(1)}(t)$ and $y_i^{(1)}(t)(i = 0,1)$ for Example 2 (Case 1).

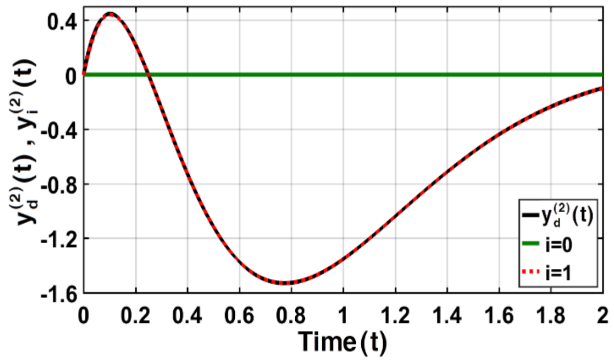

Fig. 7b. The output trajectories $y_d^{(2)}(t)$ and $y_i^{(2)}(t)(i=0,1)$ for Example 2 (Case 1).

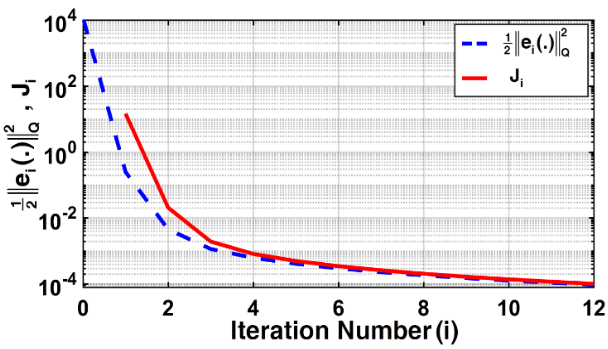

Fig. 8. The monotonic convergence of $\|e_i(.)\|_Q$ and J_i with respect to i for Example 2 (Case 1).

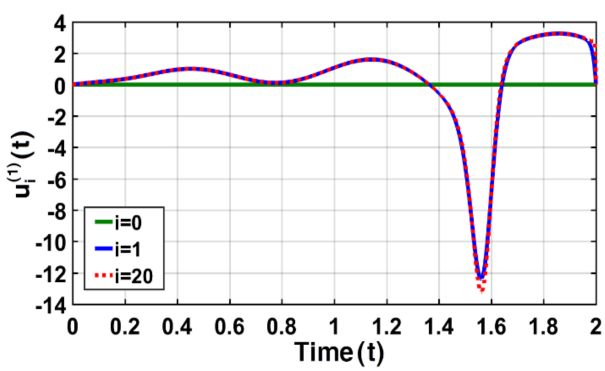

Fig. 9a. The system input $u_i^{(1)}(t)$ at iterations $i=0,1,20$ for Example 2 (Case 1).

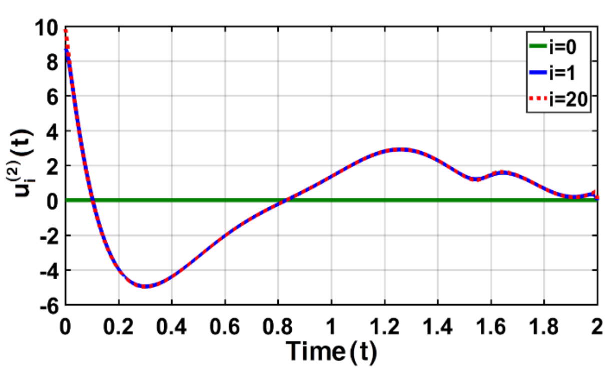

Fig. 9b. The system input $u_i^{(2)}(t)$ at iterations $i=0,1,20$ for Example 2 (Case 1).

Case 2. Using the ILC presented in [51]

The iterative learning tracking control problem for a class of MIMO nonlinear continuous-time LTV systems is studied in [51]. The system at the i-th iteration is described as equation (67):

$$\begin{cases} \dot{x}_i(t) = f(t, x_i(t)) + B(t)u_i(t) \\ y_i(t) = C(t)x_i(t) \\ x_i(0) = x_0 \\ \quad 0 \le t \le t_f i = 0,1,\dots \end{cases} . \tag{67}$$

where $B(t) \in \mathbb{R}^{n\times p}$ and $C(t) \in \mathbb{R}^{q\times n}$ are time-varying coefficients, and $f(.,.)$ is a nonlinear function that is piecewise continuous in $t \in [0, t_f]$ and globally Lipschitz in $x \in \mathbb{R}^n$, i.e.,

$$\exists 0 < \gamma < \infty : \|f(t, x_1) - f(t, x_2)\| \le \gamma \|x_1 - x_2\|, \\ \forall x_1, x_2 \in \mathbb{R}^n, \forall t \in [0, t_f]. \tag{68}$$

γ is the Lipschitz constant.

The ILC presented in [51] for system (67) is a PD-type one and has the following form:

$$u_{i+1}(t) = u_i(t) + K_P(t)e_{i+1}(t) + \lambda(t)K_D(t)\dot{e}_{i+1}(t). \tag{69}$$

where $e_{i+1}(t)$ is the tracking error at iteration $i+1$, which is defined as (4); $K_P(t)$ and $K_D(t)$ are the proportional and derivative gains, respectively, with appropriate dimensions; and $\lambda(t) \ge 1$ is an exponentially increasing function with respect to t such as e^{Mt} $(M > 0)$.

As proved in [51], if a unique control input $u_d(t)$ exists for system (67) such that it generates the desired output trajectory $y_d(t)$, i.e., $u_d(t)$ and $y_d(t)$ satisfy the following equations:

$$\begin{cases} \dot{x}_d(t) = f(t, x_d(t)) + B(t)u_d(t) \\ y_d(t) = C(t)x_d(t)) \\ x_d(0) = x_0 \\ \quad 0 \le t \le t_f \end{cases} . \tag{70}$$

where $x_d(t)$ is the state corresponding to $y_d(t)$.

Then for any initial input $u_0(t)(0 \le t \le t_f)$ and initial state $x_i(0) = x_0 (i = 0,1,\dots)$, we have:

$\lim_{i\to\infty} e_i(t) = 0 \forall t \in [0, t_f]$.

If the following condition holds:

$$\rho\{[I + \lambda(t)K_D(t)C(t)B(t)]^{-1}\} < 1 \forall t \in [0, t_f]. \tag{71}$$

where I denotes the identity matrix and $\rho\{.\}$ is the spectral radius of the matrix.

It is clear that if system (67) is linear similar to (1), that is if $f(.,.)$ is as follows:

$$f(t, x_i(t)) = A(t)x_i(t), A(t) \in \mathbb{R}^{n\times n}. \tag{72}$$

and all entries of $A(t)$ are the piecewise continuous and bonded functions in the interval $t \in [0, t_f]$, then condition (68) is satisfied. Consequently, let us try to use ILC (69) for linear system (64).

For any given differentiable desired output trajectory $y_d(t) = \left[y_d^{(1)}(t) \quad y_d^{(2)}(t)\right]^T$, such as the one given in (66), a unique control input $u_d(t) = \dot{y}_d(t) - A(t)y_d(t)$ is obtained for system (64) such that it generates $y_d(t)$. Hence, ILC (69) is applicable to system (64).

For system (64), the following gains of ILC (69) satisfy the convergence condition (71):

$$K_P(t) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, K_D(t) = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \lambda(t) = e^{0.8t}.$$

The desired output trajectories and the obtained trajectories for the system outputs are given in Fig. 10a and 10b for some iterations. Although these figures show that as the number of iterations increases, the system outputs tend towards the desired output trajectories, comparing these figures with Figs. 7a and 7b indicates that the convergence rate in Figs. 7a and 7b is much faster than in Figs. 10a and 10b. For a more accurate comparison, the following performance index, i.e., the 2-norm of the tracking error, is considered:

$$\|e_i(.)\|_2 = \left[\int_0^{t_f} e_i^T(t) e_i(t) dt\right]^{\frac{1}{2}}. \tag{73}$$

The obtained results for $\|e_i(.)\|_2$ in both cases 1 and 2 are illustrated in Fig. 11. This figure clearly demonstrates that the tracking performance of the proposed OPILC is superior to that of the method in [51].

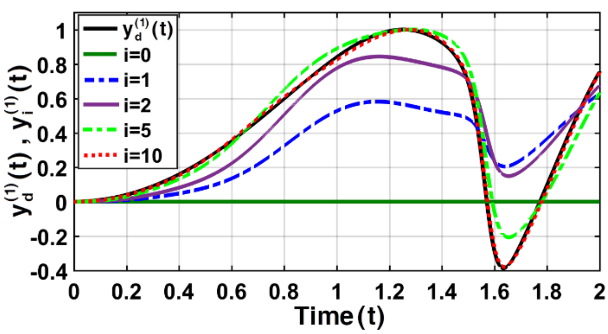

Fig. 10a. The output trajectories $y_d^{(1)}(t)$ and $y_i^{(1)}(t)$ $(i = 0,1,2,5,10)$ for Example 2 (Case 2).

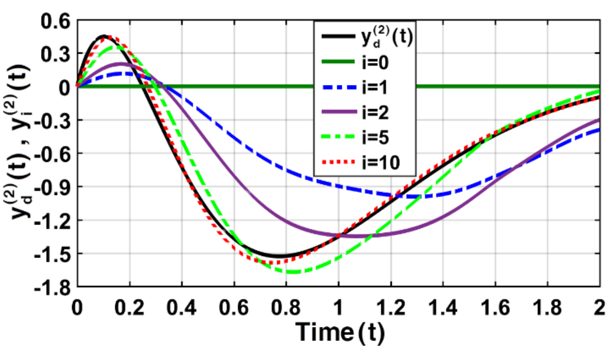

Fig. 10b. The output trajectories $y_d^{(2)}(t)$ and $y_i^{(2)}(t)$ $(i = 0,1,2,5,10)$ for Example 2 (Case 2).

Quantitative performance metrics for comparing Cases 1 and 2 in Example 2 are provided in Table II. In this table, the term "error" refers to the 2-norm of the tracking error as defined in equation (73). As observed, the proposed method exhibits a substantially higher error reduction rate and faster convergence speed compared to the ILC method introduced in [51]. Remarkably, a single iteration of the proposed method results in a 99% reduction in error, whereas the ILC approach in [51] requires 19 iterations to achieve a similar level of improvement. Furthermore, the convergence speed of the proposed scheme is approximately 19 times greater than that of the ILC in [51]. Despite the fact that the per-iteration execution time is nearly 4.1 times greater than that of the ILC approach in [51], as elaborated in Remark 6, such a rise in computational effort does not translate into any significant practical constraints.

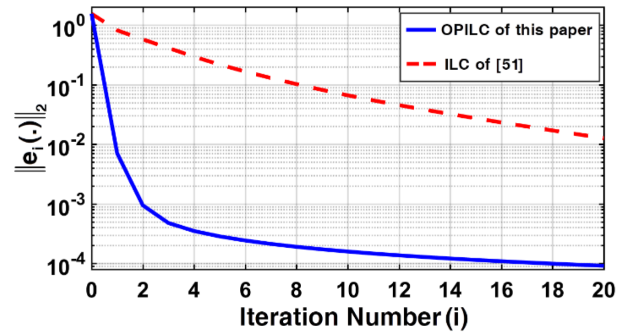

Fig. 11. The 2-norm of the tracking error $e_i(t)$ with respect to i for Example 2 (comparison of Cases 1 and 2).

Case 3. Using the ILC presented in [14]

Recently, an ILC approach has been presented in [14] for linear discrete-time, MIMO, and LTV systems described by (56), where the system has the same number of inputs and outputs and different initial time points. However, in this paper, we focus on systems that have fixed initial time points. For such systems, the structure of the ILC given in [14] is in the form of relation (57), but the method of determining the gain matrix $F(k)$ is different from equation (58). In [14], it has been shown that if $F(k)$ is chosen to make:

TABLE II. Comparison of Cases 1 and 2 for Example 2

Method		ILC of [51]	OPILC proposed in this paper
Error Reduction Rate	$I_{50\%}$	2	1
	$I_{10\%}$	7	1
	$I_{1\%}$	19	1
Convergence Speed	$I_{10^{-3}}$	38	2
Computational Complexity	T_1 (sec.)	0.078188	0.319841

$$\sup_{k \in \{1,2,\ldots,N\}} \|I - F(k) C_D(k) B_D(k-1)\| < 1, \tag{74}$$

then (59) holds.

Let us apply the above ILC to the system explained in Example 2. Since this ILC is given for discrete-time systems and in Example 2, the system is continuous-time, the given system is first discretized by choosing the sampling time $\Delta t = 2 \times 10^{-4}$. Therefore, we have:

$$N = \frac{t_f}{\Delta t} = 10^4.$$

The learning gain $F(k)$ is chosen as follows:

$$F(k) = \begin{bmatrix} 1 & -0.5 \\ -0.25 & 2 \end{bmatrix} \text{for} k = 1,2,\ldots, 10^4.$$

It is easy to show that this learning gain satisfies convergent condition (74).

As a result, Figs. 12a and 12b show the actual tracking performance of the system outputs $y_i^{(k)}(t)$ to the desired trajectories $y_d^{(k)}(t)$ at the iterations $i = 0,10,100,500$ for $k = 1,2$, respectively. As seen in these figures, with the increase of the iteration number, the system outputs converge to the desired trajectories. However, comparing these figures with Figs. 7a and 7b demonstrates that the convergence rate in Figs. 7a and 7b is much faster than in Figs. 12a and 12b. Let us use the following index to evaluate this fact more precisely:

$$EE(i) = \sum_{k=0}^{N} e_i^T(k)\, e_i(k) = \sum_{k=0}^{10^4} e_i^T(k) e_i(k). \qquad (75)$$

This index is an extension of (62) to the MIMO case. Hence, it can be stated that it is the total square error of tracking.

Figure 13 shows the profiles of $EE(i)$ versus iterations for both cases 1 and 3. It is observed that the system with the OPILC proposed in this paper has a much better convergence performance in speed and precision than the one with the ILC presented in [14].

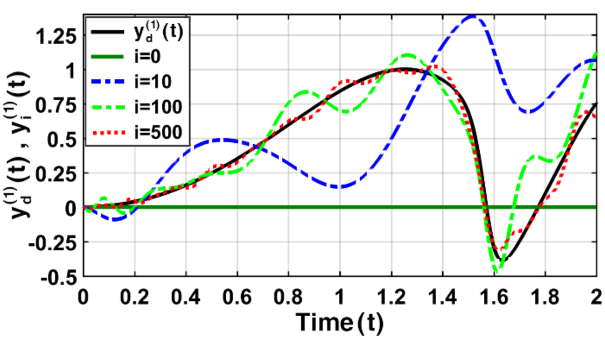

Fig. 12a. The output trajectories $y_d^{(1)}(t)$ and $y_i^{(1)}(t)$ ($i = 0,10,100,500$) for Example 2 (Case 3).

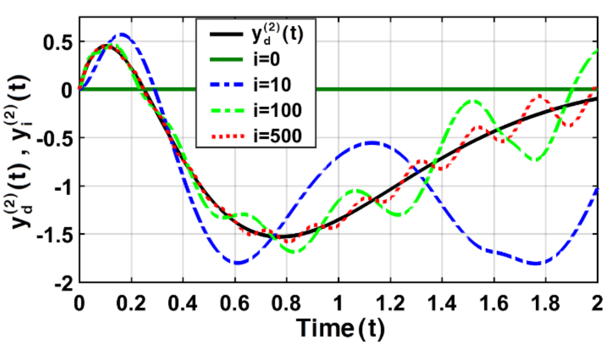

Fig. 12b. The output trajectories $y_d^{(2)}(t)$ and $y_i^{(2)}(t)$ ($i = 0,10,100,500$) for Example 2 (Case 3).

Table III reports the quantitative performance metrics for comparing Cases 1 and 3 in Example 2. In this table, the error is defined as the total square error of tracking, as formulated in equation (75). The results clearly demonstrate that the proposed method offers a dramatic improvement over the ILC approach introduced in [14], both in terms of error reduction

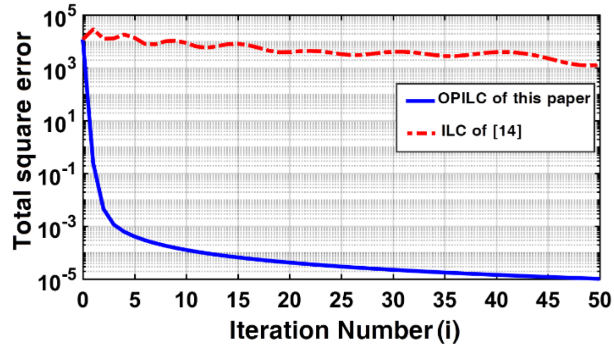

Fig. 13. The total square errors of tracking with respect to i for Example 2 (comparison of Cases 1 and 3)

rate and convergence speed. Specifically, the proposed algorithm achieves a 99% reduction in tracking error after only a single iteration, while the ILC method in [14] requires 390 iterations to reach a comparable level of accuracy. Moreover, the convergence speed of the proposed strategy is approximately 4062 times faster than that of the referenced ILC. Although the iterative execution time is approximately 1.8 times longer relative to the ILC framework described in [14], as will be discussed in greater detail in Remark 6, this increased computational burden does not undermine the practicality or operational efficiency of the proposed method.

Example 3. This example, which considers a practical MIMO process, is taken from [52]. The schematic diagram of a quadruple-tank process is shown in Fig. 14. In this process, the target is to control the level in the lower two tanks with two pumps. The process inputs are $v_{\text{in}1}$ and $v_{\text{in}2}$ (input voltages to the pumps) and the outputs are $v_{\text{out}1}$ and $v_{\text{out}2}$ (voltages from lower two tanks level measurement devices).

The voltage applied to Pump j is $v_{\text{in}j}$ and the corresponding flow is $k_j v_{\text{in}j}$ ($j = 1,2$). The parameters $\gamma_1,\ \gamma_2 \in (0,1)$ are determined from how the valves are set prior to an experiment. The flow to Tank 1 is $\gamma_1 k_1 v_{\text{in}1}$ and the flow to Tank 4 is $(1-\gamma_1)k_1 v_{\text{in}1}$ and similarly for Tank 2 and Tank 3.

The linearized model of this process around a stationary point is obtained as follows [52]:

$$\begin{cases} \dot{x} = \underbrace{\begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{bmatrix}}_{B} u \\ y = \underbrace{\begin{bmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{bmatrix}}_{C} x \end{cases} . \qquad (76)$$

and A_j is the cross-section of Tank j; a_j is cross-section of the outlet hole and h_j is water level of Tank j ($j = 1,2,3,4$). The acceleration due to gravity is denoted by g, and the superscript 0 indicates the stationary point.

TABLE III. Comparison of Cases 1 and 3 for Example 2

Method		ILC of [14]	OPILC proposed in this paper
Error Reduction Rate	$I_{50\%}$	17	1
	$I_{10\%}$	57	1
	$I_{1\%}$	390	1
Convergence Speed	$I_{10^{-3}}$	16250	4
Computational Complexity	T_1 (sec.)	0.178568	0.319841

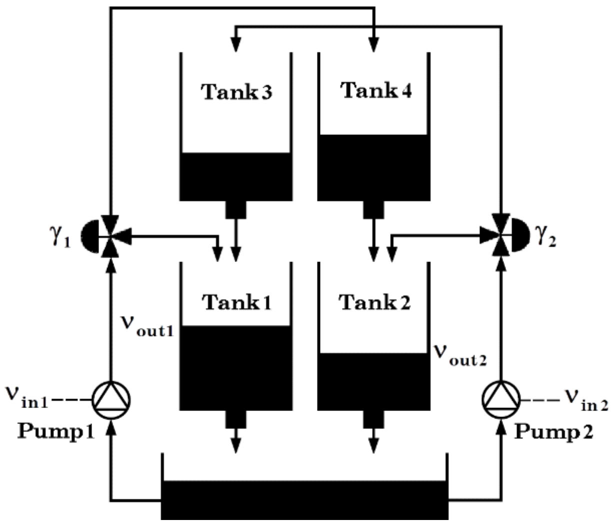

Fig. 14. Schematic diagram of the quadruple-tank process.

In the state-space model (76), the variables and parameters are defined as follows:

$$x=\left[x_1\ x_2\ x_3\ x_4\right]^T=\left[h_1\ h_2\ h_3\ h_4\right]^T-\left[h_1^0\ h_2^0\ h_3^0\ h_4^0\right]^T,$$
$$u=\left[u_1\ u_2\right]^T=\left[\nu_{\text{in}1}\ \nu_{\text{in}2}\right]^T-\left[\nu_{\text{in}1}^0\ \nu_{\text{in}2}^0\right]^T,$$
$$y=\left[y_1\ y_2\right]^T=\left[\nu_{\text{out}1}\ \nu_{\text{out}2}\right]^T-\left[\nu_{\text{out}1}^0\ \nu_{\text{out}2}^0\right]^T,$$
$$T_j=\frac{A_j}{a_j}\sqrt{\frac{2h_j^0}{g}}\qquad j=1,2,3,4. \tag{77}$$

The linear model (76) is a fourth-order system that has two inputs and two outputs. We wish to control this system iteratively in a finite time internal $[0,t_f]$.

For simulation the parameters and stationary point of the process are chosen as [52], which are follows:

$A_1=A_3=28\text{cm}^2, A_2=A_4=32\text{cm}^2,$
$a_1=a_3=0.071\text{cm}^2, a_2=a_4=0.057\text{cm}^2,$
$k_c=0.5\dfrac{V}{\text{cm}}, k_1=3.33\dfrac{\text{cm}^3}{\text{Vs}}, k_2=3.35\dfrac{\text{cm}^3}{\text{Vs}},$
$\gamma_1=0.7, \gamma_2=0.6, g=981\dfrac{\text{cm}}{s^2},$
$h_1^0=12.4\text{cm}, h_2^0=12.7\text{cm},$
$h_3^0=1.8\text{cm}, h_4^0=1.4\text{cm},$
$\nu_{\text{in}1}^0=\nu_{\text{in}2}^0=3V.$

The time duration of iterations is selected ten seconds ($t_f=10sec$). The desired output trajectories, and the state and output disturbances are chosen as follows:

$$y_{1d}(t)=\begin{cases}\frac{1}{6}t & 0\le t\le 3\\ 0.5 & 3\le t\le 7,\\ \frac{1}{6}(10-t) & 7\le t\le 10\end{cases}$$
$$y_{2d}(t)=0.01\sinh\left(5\sin(0.2\pi t)\right).$$

$$\eta_x(t)=\begin{bmatrix}\cos(t)\tanh(0.75t)\\ 0.5\sin(t)\\ 0.25\cos(\pi t)\\ -\text{Ln}(t+1)\end{bmatrix},\eta_y(t)=\begin{bmatrix}0.01t^2\\ 0.5(1-e^{\sin(0.25\pi t)})\end{bmatrix}.$$

The system initial conditions and the initial input all selected to be zero.

To compare the performance of the proposed method with the approach in [14], simulations of this example are conducted under two scenarios, applying the method proposed here and the approach of [14], respectively.

Case 1. Using the OPILC proposed in this paper

The symmetric positive definite matrices $Q(t)$ and $R(t)$ in the cost function (5) are chosen as follows:
$Q(t)=10^4I, R(t)=I.$

The simulated output trajectories of the system at several iterations, along with the desired trajectories, are shown in Figs. 15a and 15b. These plots demonstrate that as the iteration index increases, the system outputs quickly converge to the desired trajectories. In fact, at iteration $i=2$, a perfect match is observed between the actual and desired outputs.

Fig. 16 further illustrates the evolution of $\frac{1}{2}\|e_i(.)\|_Q^2$ and J_i over iterations. Both measures are seen to decrease steadily and approach zero, which supports the theoretical findings established in Theorems 2 and 3.

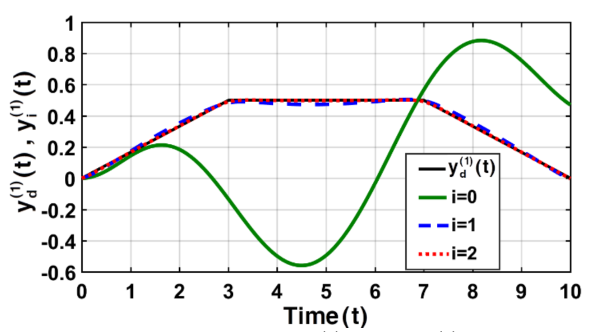

Fig. 15a. The output trajectories $y_d^{(1)}(t)$ and $y_i^{(1)}(t)(i=0,1,2)$ for Example 3 (Case 1).

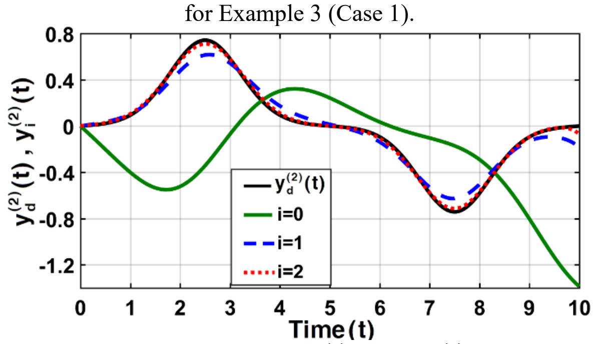

Fig. 15b. The output trajectories $y_d^{(2)}(t)$ and $y_i^{(2)}(t)(i=0,1,2)$ for Example 3 (Case 1).

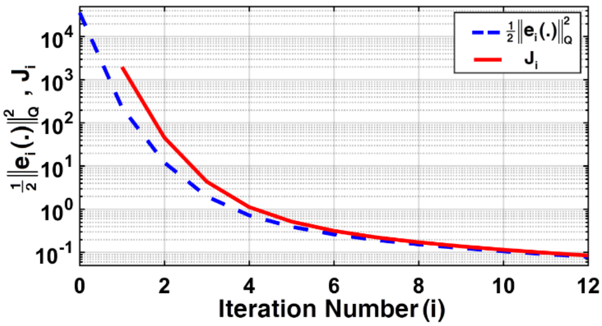

Fig. 16. The monotonic convergence of $\|e_i(.)\|_Q$ and J_i with respect to i for Example 3 (Case 1).

The system input profiles, shown in Figs. 17a and 17b, change gradually with each iteration and ultimately settle into two consistent forms. These patterns are not predetermined, but instead result from the underlying learning mechanism.

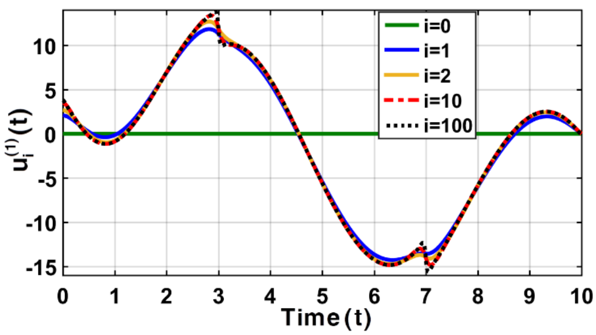

Fig. 17a. The system input $u_i^{(1)}(t)$ at iterations $i = 0,1,2,10,100$ for Example 3 (Case 1).

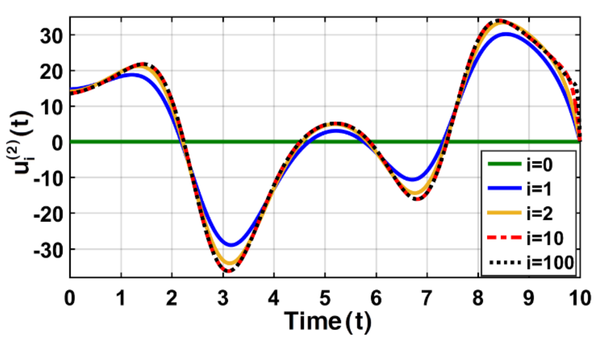

Fig. 17b. The system input $u_i^{(2)}(t)$ at iterations $i = 0,1,2,10,100$ for Example 3 (Case 1).

Case 2. Using the ILC presented in [14]

The ILC proposed in [14], which is discussed in Case 3 of Example 2, is now applied to the system in Example 3. Since the method in [14] is designed for discrete-time systems, and the system in Example 3 is continuous-time, we first discretize the system using a sampling time of $\Delta t = 0.01$. Then, we have:

$$N = \frac{t_f}{\Delta t} = 10^3.$$

The following learning gain is chosen:

$$F(k) = \begin{bmatrix} 10 & -5 \\ 0.8 & 20 \end{bmatrix} \text{for} k = 1,2,\ldots,10^3.$$

It can be readily shown that this learning gain satisfies the convergence condition (74).

Figures 18a and 18b illustrate the tracking performance of the system outputs $y_i^{(k)}(t)$ relative to the desired trajectories $y_d^{(k)}(t)$ at iterations $i = 0,50,100$, for $k = 1,2$, respectively. As evident from these figures, the tracking accuracy improves progressively with increasing iteration number, leading to convergence toward the reference signals. Nevertheless, a comparative analysis with Figs. 15a and 15b reveals that the convergence rate in Case 1 is considerably higher. To provide a more rigorous assessment, the total squared tracking error $EE(i)$, as defined in equation (75), is employed. Figure 19 presents the evolution of $EE(i)$ across iterations for both Case 1 and Case 2. The results confirm that the OPILC scheme proposed in this study offers significantly enhanced convergence performance, both in terms of speed and accuracy, compared to the ILC approach reported in [14].

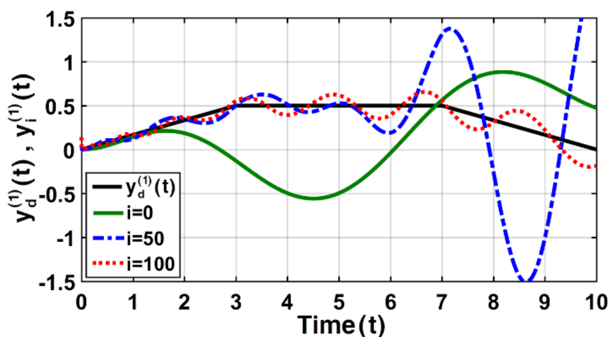

Fig. 18a. The output trajectories $y_d^{(1)}(t)$ and $y_i^{(1)}(t)(i = 0,50,100)$ for Example 3 (Case 2).

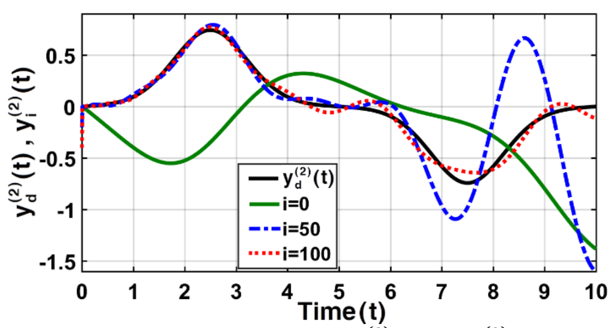

Fig. 18b. The output trajectories $y_d^{(2)}(t)$ and $y_i^{(2)}(t)(i = 0,50,100)$ for Example 3 (Case 2).

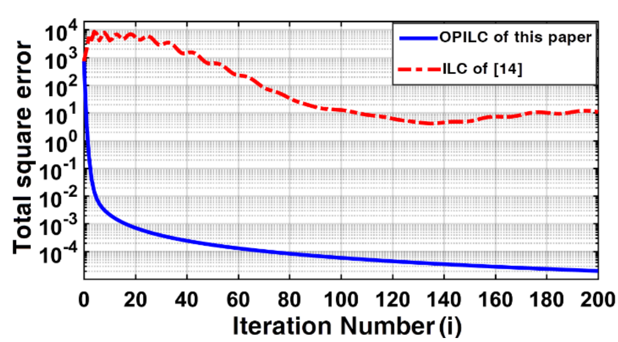

Fig. 19. The total square errors of tracking with respect to i for Example 3 (comparison of Cases 1 and 2)

Table IV presents the quantitative performance measures comparing Cases 1 and 2 for Example 3. In this table, the error metric refers to the total square error of tracking, as defined in equation (75). As the data indicate, the proposed method also demonstrates substantially superior error reduction rate and convergence speed in Example 3 compared to the ILC method reported in [14]. Specifically, a single iteration of the proposed algorithm leads to a 99% reduction in tracking error, whereas the ILC approach in [14] requires 232 iterations to achieve a similar reduction. Additionally, the convergence speed of the proposed method in this example exceeds that of the referenced ILC by a factor of over 83. While the method requires a per-iteration runtime approximately 1.7 times that of the ILC technique referenced in [14], this elevated computational demand, as will be further addressed in Remark 6, does not pose any meaningful limitations in practical applications.

TABLE IV. Comparison of Cases 1 and 2 for Example 3

Method		ILC of [14]	OPILC proposed in this paper
Error Reduction Rate	$I_{50\%}$	56	1
	$I_{10\%}$	71	1
	$I_{1\%}$	232	1
Convergence Speed	$I_{10^{-3}}$	1330	16
Computational Complexity	T_1 (sec.)	0.034856	0.058784

Example 4. A relatively complex fifth-order fully time-varying MIMO system is given in the form of (1) with the following coefficients:

$$A(t)=0.4\begin{bmatrix} cos(2\pi t) & t & 2t-4 & e^{-2t} & 0.01(t^2-1) \\ sin(\pi t^2) & e^{-t} & 2\,sin(3\pi t) & 0.1\,cosh(t) & 0.2\,sin(0.5\pi\sqrt{t}) \\ (t)^{e^{-0.5t}} & t\,cos(4\pi t) & \mathrm{Ln}(t+0.01) & 0.1t & -t^2 \\ -20\,tan^{-1}(4t) & sin(cosh(t)) & -\dfrac{te^{-0.1t}}{t+10} & 3\,cos^{-1}(0.02t) & 4\,cos(2\pi t^3) \\ 25-t & -\mathrm{Ln}(1.1+sin(20t)) & (t+1)^{sin^{-1}(0.01t)} & e^{cos(8t)} & \dfrac{t}{t+25} \end{bmatrix},$$

$$B(t)=0.02\begin{bmatrix} 2+te^{-t} & t-cos(t) & \sqrt[3]{sin(\pi t)} \\ t^2 & \mathrm{Ln}(0.1+t) & te^{-t}cos(\pi\sqrt{t}) \\ sin(4t) & \dfrac{3t^2}{t^2+4} & 7e^{sinh^{-1}(0.25t)} \\ \sqrt{\dfrac{2}{t+1}} & e^{2t} & tanh[sin(t^2)] \\ -t & \sqrt{10-t} & \dfrac{e^{2t}}{t+3} \end{bmatrix},$$

$$C(t)=0.1\begin{bmatrix} \dfrac{1}{1+t^{25-t}} & t^{t+1} \\ (sin^{-1}(e^{-0.02t}))^3 & cos(t^2) \\ \sqrt[5]{t} & \dfrac{sinh^{-1}(\mathrm{Ln}(1+t))}{e^t+7} \\ e^{sin(\pi t)} & 0.5t-3 \\ -\dfrac{1}{tan^{-1}(t)+0.1} & |t-2.5| \end{bmatrix}^T.$$

We aim to control this complex system iteratively in the time interval [0,5].

Let the desired trajectories be defined as:

$$\begin{cases} y_d^{(1)}(t)=1.25t^2-0.05t^4 \\ y_d^{(2)}(t)=t\,tan^{-1}(2.5-t) \end{cases} \quad 0\le t\le 5.$$

To perform the simulation, the initial conditions of the system and its input at the initial iteration $i=0$ are all assumed to be zero. The disturbances $\eta_x(t)$ and $\eta_y(t)$ are chosen as follows:

$$\eta_x(t)=0.1\begin{bmatrix} te^{-0.5t} \\ cosh(sin(t)) \\ t(20-t) \\ -5\,cos(0.5t) \\ tan^{-1}(t) \end{bmatrix}, \eta_y(t)=0.5\begin{bmatrix} sin(\mathrm{Ln}(t+1)) \\ \mathrm{Ln}(sin(t)+2) \end{bmatrix}.$$

This example presents a particularly challenging scenario, under which previously reported ILC methods do not achieve convergence, to the best of our knowledge. Let us now apply the proposed OPILC to this system.

The symmetric positive definite matrices $Q(t)$ and $R(t)$ as follows:

$Q(t)=10I, R(t)=I.$

Here, due to the limitation in the number of pages, only the results obtained for $\frac{1}{2}\|e_i(.)\|_Q^2$ and cost function J_i from the simulation are given in Fig. 20 (in the log scale on both axes). This figure proves that the learning convergence is monotonic from the point of view of both $\|e_i(.)\|_Q$ and J_i, consistent with Theorems 2 and 3.

To study the effect of the weighing matrices $Q(t)$ and $R(t)$ in cost function (5) on the performance of the proposed OPILC, the simulation of Example 4 is repeated using $Q(t)=\lambda I$ and $R(t)=I$, for several different values of the positive scalar λ. As a result, the obtained profiles for the 2-norm of the tracking error $e_i(t)$ and the input variation $v_i(t)$, which are defined as (73), are given in Figs. 21 and 22 in the log scales, respectively. These figures clearly demonstrate that with the increase of λ, the convergence rate of $e_i(t)$ increases, while that of $v_i(t)$ decreases. To comprehend and explain this phenomenon, it can be stated that by choosing large or small weighing matrices $Q(t)$ and $R(t)$ relative to each other, the desired importance can be assigned to each term $\|e_i(.)\|_Q^2$ and $\|v_i(.)\|_R^2$ in cost function (5). That is, if the minimization of the tracking error has a higher significance, then $Q(t)$ must be chosen larger than $R(t)$. However, if the prevention of sharp and fast variation of the system input is of interest, then $Q(t)$ must be chosen to be small relative to $R(t)$.

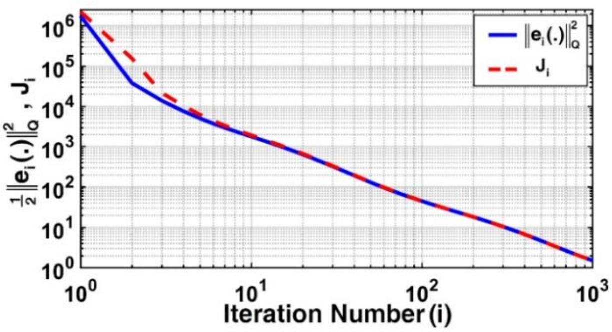

Fig. 20. The monotonic convergence of $\|e_i(.)\|_Q$ and J_i with respect to i for Example 4.

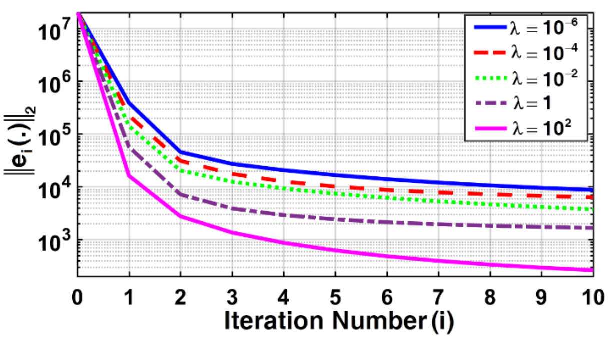

Fig. 21. The profiles of the 2-norm of the tracking error $e_i(t)$ for several different values of λ for Example 4.

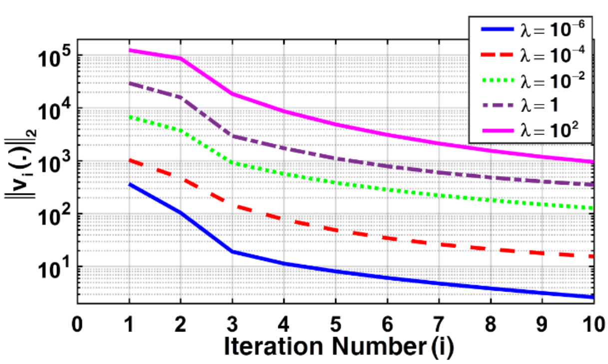

Fig. 22. The profiles of the 2-norm of the input variation $v_i(t)$ for several different values of λ for Example 4.

Remark 6. The proposed OPILC has been compared with several existing ILC approaches. Due to space limitations, only three of these comparisons are presented in Examples 1 through 3. The results clearly demonstrate that the proposed OPILC achieves significantly higher error reduction rate and faster convergence speed than previous methods. Furthermore, the convergence behavior is monotonic, an important characteristic for any ILC algorithm, which has also been theoretically proven in Theorems 2 and 3. However, as shown in Tables I to IV, the proposed method exhibits higher computational complexity than some existing ILC approaches, leading to a longer computation time for generating the control input in each iteration. To identify the source of this increased computation time, let us analyze the structure of the proposed OPILC. As noted in Remark 4, the matrix Riccati differential equation (15) is solved only once, prior to the first iteration, and the resulting gain matrix $K(t)$ is used throughout all subsequent iterations. Therefore, the time required to solve equation (15) does not contribute to the per-iteration computational burden. The main source of increased computation lies in solving the linear vector differential equation (16) to compute $g_i(t)$, which must be carried out before each iteration, as discussed in Remark 5. Nevertheless, in practical industrial repetitive systems (such as robotic manipulators) a reset period typically exists between two successive iterations. This period is usually long enough to accommodate the solution of equation (16), making its computational burden practically negligible. Thus, the additional computational effort does not pose any serious challenge or limitation in real-world applications.

VI. Conclusion

This paper introduced a novel optimal ILC approach for MIMO continuous-time LTV systems. The proposed method transforms the ILC problem into a linear quadratic tracking-like (LQT) problem. Subsequently, LQT theory is employed to derive an explicit closed-loop learning control law that guarantees both accurate tracking and minimized control effort. In contrast to many existing ILC methods for continuous-time systems that depend on derivative-based learning algorithms, which are sensitive to measurement noise, the proposed design avoids these terms, mitigating noise amplification.

Through rigorous mathematical analysis, it has been proven that the proposed OPILC guarantees the monotonic convergence of both the tracking error and the associated cost function, a property that is essential for any high-performance ILC. The effectiveness of the method is demonstrated through four comprehensive simulation examples, which show that the proposed scheme achieves faster convergence and higher tracking accuracy compared to existing ILC methods. Additionally, the method demonstrates significant computational efficiency by determining certain learning gains prior to the first iteration and updating others during system reset periods.

In summary, the proposed OPILC framework offers a powerful solution for iterative learning control in continuous-time systems, providing a foundation for further advancements and extensions to more complex systems. Extending the approach to discrete-time LTV systems is straightforward. However, the formulation and solution of the problem for each of the following cases require further investigation:

1. Output Measurement: The current approach assumes full access to the state vector. Extending this method to cases where only output measurements are available is an important future step.

2. System Constraints: Incorporating input and state constraints into the OPILC framework is crucial for practical applications where physical limitations must be considered.

3. Iteration-Varying Disturbances: The method assumes iteration-invariant disturbances, but many real-world systems experience iteration-varying disturbances. Accounting for this in the optimization framework could improve robustness.

4. System Uncertainties: Future work could explore handling system uncertainties and unknown parameters within the OPILC framework, enhancing its robustness in practical applications.

5. Nonlinear Systems: The proposed method is tailored for linear systems. Extending it to nonlinear systems remains a valuable research direction, as many real-world systems exhibit nonlinear dynamics.

REFERENCES

[1] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, pp. 1099-1121, 2007, doi: 10.1109/TSMCC.2007.905759.

[2] J.-X. Xu, "A survey on iterative learning control for nonlinear systems," International Journal of Control, vol. 84, pp. 1275-1294, 2011, doi: 10.1080/00207179.2011.574236.

[3] A. Madady, "A self-tuning iterative learning controller for time variant systems," Asian Journal of Control, vol. 10, pp. 666-677, 2008, doi: 10.1002/asjc.67.

[4] D. Shen and Y. Wang, "Survey on stochastic iterative learning control," Journal of Process Control, vol. 24, pp. 64-77, 2014, doi: 10.1016/j.jprocont.2014.04.013.

[5] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," Journal of Robotic systems, vol. 1, pp. 123-140, 1984, doi: 10.1002/rob.4620010203.

[6] D. Shen, "Iterative learning control with incomplete information: A survey," IEEE/CAA Journal of Automatica Sinica, vol. 5, pp. 885-901, 2018, doi: 10.1109/JAS.2018.7511123.

[7] F. Afsharnia, A. Madady, and M. B. Menhaj, "Adaptive iterative learning control for unknown linear time-varying continuous systems," Transactions of the Institute of Measurement and Control, vol. 42, pp. 981-996, 2020, doi: 10.1177/0142331219880110.

[8] J. Zhou, C. T. Freeman, and W. Holderbaum, "Multiple-model iterative learning control with application to stroke rehabilitation," Control Engineering Practice, vol. 154, p. 106134, 2025, doi: 10.1016/j.conengprac.2024.106134.

[9] H. Farokhi Moghadam, N. Vasegh, and S. M. Seyed Moosavi, "Adaptive repetitive control for periodic disturbance rejection with unknown period," International Journal of Industrial Electronics Control and Optimization, vol. 3, pp. 187-194, 2020, doi: 10.22111/ieco.2019.31017.1197.

[10] F. Afsharnia, A. Madady, and M. B. Menhaj, "Neural iterative learning identifier-based iterative learning controller for time-varying nonlinear systems," Asian Journal of Control, vol. 24, pp. 1998-2012, 2022, doi: 10.1002/asjc.2617.

[11] R. Movahed Asl and A. Madady, "Stabilization of two-dimensional mixed continuous-discrete-time systems via dynamic output feedback with application to iterative learning control design," Transactions of the Institute of Measurement and Control, vol. 44, pp. 172-187, 2022, doi: 10.1177/01423312211022205.

[12] X. Zhang and Z. Hou, "Data-driven predictive point-to-point iterative learning control," Neurocomputing, vol. 518, pp. 431-439, 2023, doi: 10.1016/ j.neucom.2022.11.014.

[13] Y. Tao, H. Tao, Z. Zhuang, V. Stojanovic, and W. Paszke, "Quantized iterative learning control of communication-constrained systems with encoding and decoding mechanism," Transactions of the Institute of Measurement and Control, vol. 46, pp. 1943-1954, 2024, doi: 10.1177/01423312231225782.

[14] Y. Kong, X.-D. Li, and X. Li, "Iterative learning control approach for linear discrete time-varying systems with different initial time points," Journal of the Franklin Institute, vol. 361, p. 106702, 2024, doi: 10.1016/j.jfranklin.2024.106702.

[15] J. Liu, Y. Liu, R. Chi, and Z. Hou, "Error tracking-driven adaptive backstepping ILC of nonlinear systems with initial shifts," Journal of the Franklin Institute, vol. 362, p. 107455, 2025, doi: 10.1016/j.jfranklin.2024.107455.

[16] S-M Hashem Zadeh, S. Khorashadizadeh, M-M Fateh, and M. Hadadzarif, "Optimal sliding mode control of a robot manipulator under uncertainty using PSO," Nonlinear Dynamics, vol. 84, pp. 2227–2239, 2016, doi: 10.1007/s11071-016-2641-4.

[17] B. Dai, J. Gong, C. Li, and H. Ning, "Iterative learning control realized using an iteration-varying forgetting factor based on optimal gains," Transactions of the Institute of Measurement and Control, vol. 43, pp. 2334-2344, 2021, doi: 10.1177/0142331221996507.

[18] H. Jay, "Model based iterative learning control with a quadratic criterion for linear time-varying systems," Automatica, vol. 36, pp. 641-657, 2000, doi: 10.1016/S0005-1098(99)00194-6.

[19] D. H. Owens and K. Feng, "Parameter optimization in iterative learning control," International Journal of Control, vol. 76, pp. 1059-1069, 2003, doi: 10.1080/0020717031000121410.

[20] S. S. Saab, "Optimal selection of the forgetting matrix into an iterative learning control algorithm," IEEE Transactions on Automatic Control, vol. 50, pp. 2039-2043, 2005, doi: 10.1109/TAC.2005.860232.

[21] J. Hätönen, D. H. Owens, and K. Feng, "Basis functions and parameter optimisation in high-order iterative learning control," Automatica, vol. 42, pp. 287-294, 2006, doi: 10.1016/j.automatica.2005.05.025.

[22] A. Madady, "PID type iterative learning control with optimal gains," International Journal of Control, Automation, and Systems, vol. 6, pp. 194-203, 2008.

[23] S. Mishra, U. Topcu, and M. Tomizuka, "Optimization-based constrained iterative learning control," IEEE Transactions on Control Systems Technology, vol. 19, pp. 1613-1621, 2010, doi: 10.1109/TCST.2010.2083663.

[24] D. H. Owens, "Multivariable norm optimal and parameter optimal iterative learning control: a unified formulation," International Journal of Control, vol. 85, pp. 1010-1025, 2012, doi: 10.1080/00207179.2012.673136.

[25] A. Madady, "An extended PID type iterative learning control," International Journal of Control, Automation and Systems, vol. 11, pp. 470-481, 2013, doi: 10.1007/s12555-012-0350-4.

[26] H. Sun and A. G. Alleyne, "A computationally efficient norm optimal iterative learning control approach for LTV

systems," Automatica, vol. 50, pp. 141-148, 2014, doi: 10.1016/j.automatica.2013.09.009.
[27] J. Van Zundert, J. Bolder, and T. Oomen, "Optimality and flexibility in iterative learning control for varying tasks," Automatica, vol. 67, pp. 295-302, 2016, doi: 10.1016/j.automatica.2016.01.026.
[28] A. Madady, H.-R. Reza-Alikhani, and S. Zamiri, "Optimal N-parametric type iterative learning control," International Journal of Control, Automation and Systems, vol. 16, pp. 2187-2202, 2018, doi: 10.1007/s12555-017-0259-z.
[29] F. Memon and C. Shao, "An optimal approach to online tuning method for PID type iterative learning control," International Journal of Control, Automation and Systems, vol. 18, pp. 1926-1935, 2020, doi: 10.1007/s12555-018-0840-0.
[30] F. Memon and C. Shao, "Robust optimal PID type ILC for linear batch process," International Journal of Control, Automation and Systems, vol. 19, pp. 777-787, 2021, doi: 10.1007/s12555-019-1033-1.
[31] Y. Zhang, B. Chu, and Z. Shu, "Parameter optimal iterative learning control design: from model-based, data-driven to reinforcement learning," IFAC-PapersOnLine, vol. 55, pp. 494-499, 2022, doi: 10.1016/ j.ifacol.2022.07.360.
[32] Z. Zhuang, H. Tao, Y. Chen, V. Stojanovic, and W. Paszke, "An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, pp. 3461-3473, 2022, doi: 10.1109/TSMC.2022.3225381.
[33] F. Arnold and R. King, "Norm-optimal iterative learning control in an integer-valued control domain," International Journal of Control, vol. 96, pp. 170-181, 2023, doi: 10.1080/00207179.2021.1983213.
[34] Y. Zhou, K. Gao, X. Tang, H. Hu, D. Li, and F. Gao, "Conic input mapping design of constrained optimal iterative learning controller for uncertain systems," IEEE Transactions on Cybernetics, vol. 53, pp. 1843-1855, 2022, doi: 10.1109/TCYB.2022.3155754.
[35] K. Tsurumoto, W. Ohnishi, T. Koseki, M. van Haren, and T. Oomen, "Combined time-domain optimization design for task-flexible and high performance ILC," in 2024 European Control Conference (ECC), 2024, pp. 1190-1195, doi: 10.23919/ECC64448.2024.10590964.
[36] M. van Haren, K. Tsurumoto, M. Mae, L. Blanken, W. Ohnishi, and T. Oomen, "A frequency-domain approach for enhanced performance and task flexibility in finite-time ILC," European Journal of Control, p. 101033, 2024, doi: 10.1016/j.ejcon.2024.101033.
[37] E. Rafajłowicz and W. Rafajłowicz, "Iterative learning in optimal control of linear dynamic processes," International Journal of Control, vol. 91, pp. 1522-1540, 2018, doi: 10.1080/00207179.2017.1320810.
[38] Z. Zhuang, H. Tao, Y. Chen, T. Oomen, W. Paszke, and E. Rogers, "Optimal iterative learning control design for continuous-time systems with nonidentical trial lengths using alternating projections between multiple sets," Journal of the Franklin Institute, vol. 360, pp. 3825-3848, 2023, doi: 10.1016/j.jfranklin.2023.02.006.
[39] Y. Wu and D. Meng, "Data informativity for tracking control of learning systems: Test and design conditions," Automatica, vol. 171, p. 111885, 2025, doi: 10.1016/j.automatica.2024.111885.
[40] K. Gao, J. Lu, Y. Zhou, and F. Gao, "A computationally efficient policy optimization scheme in feedback iterative learning control for nonlinear batch process," Computers & Chemical Engineering, p. 109005, 2025, doi: 10.1016/j.compchemeng.2025.109005.
[41] A. Madady and H. R. Reza-Alikhani, "A guaranteed monotonically convergent iterative learning control," Asian Journal of Control, vol. 14, pp. 1299-1316, 2012, doi: 10.1002/asjc.397.
[42] D. H. Owens, Iterative learning control: An optimization paradigm. London: Springer, 2016, ISBN: 978-1-4471-6772-3.
[43] R. Dosthosseini, N. Banisaeid, and A. Dashti, "Dynamic modelling and optimal control of a rotor of active magnetic bearings," International Journal of Industrial Electronics Control and Optimization, vol. 2, pp. 59-70, 2019, doi: 10.22111/IECO.2018.25595.1053.
[44] P. L. Falb and M. Athans, Optimal control: An introduction to the theory and its applications: McGraw-Hill, 2000, ISBN: 0070024138, 9780070024137.
[45] F. L. Lewis, D. Vrabie, and V. L. Syrmos, Optimal control: John Wiley & Sons, 2012, ISBN-10. 0470633492 ; ISBN-13. 978-0470633496.
[46] P. D. Nguyen and N. H. Nguyen, "An intelligent parameter determination approach in iterative learning control," European Journal of Control, vol. 61, pp. 91-100, 2021, doi: 10.1016/j.ejcon.2021.06.001.
[47] X. Ruan and Y. Liu, "Monotone convergence rate of norm-optimal-gain-arguable iterative learning control for LDTI systems," Asian Journal of Control, vol. 24, pp. 920-941, 2022, doi: 10.1002/asjc.2498.
[48] X.-D. Li, J. K. Ho, and T. W. Chow, "Iterative learning control for linear time-variant discrete systems based on 2-D system theory," IEE Proceedings-Control Theory and Applications, vol. 152, pp. 13-18, 2005, doi: 10.1049/ip-cta:20041125.
[49] M. J. Sidi, Spacecraft dynamics and control: a practical engineering approach vol. 7: Cambridge university press, 1997, ISBN-10: 0521787807.
[50] M. Tranninger, R. Seeber, J. G. Rueda-Escobedo, and M. Horn, "Strong detectability and observers for linear time-varying systems," Systems & Control Letters, vol. 170, p. 105398, 2022, doi: 10.1016/j.sysconle.2022.105398.
[51] R. Ma and G. Zhang, "Iterative learning tracking control for a class of MIMO nonlinear time-varying systems," International Journal of Modelling, Identification and Control, vol. 27, pp. 271-278, 2017, doi: 10.1504/IJMIC.2017.084721.
[52] K. H. Johansson, "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," IEEE Transactions on control systems technology, vol. 8, pp. 456-465, 2000, doi: 10.1109/87.845876.

Appendix

Let us consider a linear time-varying system whose state $x(t) \in \mathbb{R}^n$, input $u(t) \in \mathbb{R}^p$, and output $y(t) \in \mathbb{R}^q$ are related by the system of equations:

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t) \\ y(t) = C(t)x(t) \end{cases}. \qquad \text{(A-1)}$$

For system (A-1), the linear quadratic tracking (LQT) problem is defined as follows [44, 45]:

The desired output $\mathrm{r}(\mathrm{t}) \in \mathbb{R}^{\mathrm{q}}$ is given, while $\mathrm{u}(\mathrm{t})$ is unconstrained in magnitude. Determine $\mathrm{u}(\mathrm{t})$ such that the following quadratic cost function is minimized:

$$J = \frac{1}{2}\left[r(t_f) - y(t_f)\right]^T F\left[r(t_f) - y(t_f)\right]$$
$$+\frac{1}{2}\int_0^{t_f}\{[r(t) - y(t)]^T Q(t)[r(t) - y(t)] + u^T(t)R(t)u(t)\}\, dt. \quad \text{(A-2)}$$

where the terminal time t_f is specified, $Q(t) \in \mathbb{R}^{q\times q}$ and $R(t) \in \mathbb{R}^{p\times p}$ are the positive definite matrices, and $F \in \mathbb{R}^{q\times q}$ is a constant positive semidefinite matrix.

The calculus of variations has been employed to demonstrate that the above optimal control problem admits a unique solution, expressed analytically as follows (see Chapter 9 of [44] or Chapter 12 of [45]):

$$u(t) = -R^{-1}(t)B^T(t)[K(t)x(t) - g(t)]. \quad \text{(A-3)}$$

where $K(t) \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix, obtained from the solution of the following Riccati-type matrix differential equation:

$$\dot{K}(t) = -K(t)A(t) - A^T(t)K(t)$$
$$+K(t)B(t)R^{-1}(t)B^T(t)K(t)$$
$$-C^T(t)Q(t)C(t), \quad \text{(A-4)}$$

with the terminal condition

$$K(t_f) = C^T(t_f)FC(t_f). \quad \text{(A-5)}$$

The vector $g(t) \in \mathbb{R}^n$ is the solution of the linear vector differential equation:

$$\dot{g}(t) = -[A(t) - B(t)R^{-1}(t)B^T(t)K(t)]^T g(t)$$
$$-C^T(t)Q(t)r(t), \quad \text{(A-6)}$$

with the terminal condition

$$g(t_f) = C^T(t_f)Fr(t_f). \quad \text{(A-7)}$$

It has also been shown that applying input (A-3) to system (A-1) yields the minimum of the cost function (A-2), as follows

$$J^* = \frac{1}{2}x^T(0)K(0)x(0) - x^T(0)g(0) + \phi(0). \quad \text{(A-8)}$$

where $\phi(0)$ is obtained from the backward integration of the following scalar differential equation (see Chapter 9 of [44] or Chapter 12 of [45]):

$$\dot{\phi}(t) = -\frac{1}{2}r^T(t)Q(t)r(t) + \frac{1}{2}g^T(t)B(t)R^{-1}(t)B^T(t)g(t), \quad \text{(A-9)}$$

with the terminal condition:

$$\phi(t_f) = \frac{1}{2}r^T(t_f)Fr(t_f). \quad \text{(A-10)}$$

Naser Taghvamanesh was born in 1988 in Boroujerd, Iran. He received his B.Sc., M.Sc. degrees in electrical engineering from Shahid Sattari University of Science and Technology and Tafresh University in 2005 and 2014, respectively. He is currently pursuing the Ph.D. degree in the Control Department at the Faculty of Electrical Engineering, Tafresh University. His research interests are learning control systems, convex optimization, and optimal control.

Ali Madady was born in Aqdash, Saveh, Iran in 1966. He received his B.Sc., M.Sc. and Ph.D. degrees in electrical and electronic engineering from Amir-Kabir University of Technology (Tehran Polytechnic), Iran, in 1989, 1993 and 2001, respectively. From 1991 to 1996, he was a part-time research member of Iranian Research Organization for Science and Technology (IROST). Also from 1994 to 2001, he was a part-time lecturer of Shamsipour Institute of Technology (SIT), Tehran, Iran. In 2001, he joined Tafresh University, Tafresh, Iran, where he is currently an Associate Professor and originator and head of Control Engineering Group. His research interests include iterative learning, adaptive control, robust control and fractional-order systems.