Data-enabled iterative learning control: A zero-sum game design for time-scale-varying tasks ⋆

Zhihe Zhuang [a], Rodrigo A. González [b], Hongfeng Tao [a], Wojciech Paszke [c], Tom Oomen [b,d]

[a] *Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi, China.*

[b] *Control Systems Technology Section, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands.*

[c] *Institute of Automation, Electronic and Electrical Engineering, University of Zielona Góra, Zielona Góra, Poland.*

[d] *Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands.*

Abstract

Iterative learning control (ILC) is an intelligent control methodology for tackling iteration-invariant exogenous inputs. It is of great significance to develop its extrapolation for more general repetitive tasks with mutual similarity, e.g., tasks with different time scales. In practice, discrete-time ILC with sampling behavior for time-scale-varying tasks suffers from the failure of perfect corresponding learning and environment-dependent iteration-varying disturbances. This paper develops a novel direct data-based ILC algorithm using off-policy Q-learning for tasks with varying time scales, enabling the robust learning of an optimal ILC policy from experimental input/output (I/O) data. From a two-player zero-sum game perspective, the iteration-varying disturbance generated from the varying time scales of repetitive tasks is tackled quantitatively with a preset disturbance attenuation level. Further, to emphasize the importance of theoretical guarantees of reinforcement learning (RL)-based ILC designs, the data efficiency of the developed algorithm is enhanced based on Willems' Fundamental Lemma, and a rigorous convergence analysis is given. The simulation model of an F-16 aircraft autopilot is employed to show the effectiveness of the developed approach.

Key words: Iterative learning control; Time-scale-varying task; Data-based control; Reinforcement learning.

1 Introduction

Iterative learning control (ILC) is an effective methodology to reduce iteration-invariant disturbances in repetitive tasks. Without iteration-varying disturbances, ILC can achieve perfect tracking subject to an iteration-invariant reference and plant model, identical iteration length, and strictly resetting initial states. Since [5], the theory and application of ILC have gained widespread attention of the control community. ILC has been applied successfully to batch processes, industrial robots, healthcare devices, wafer stages, among other applications [10,2].

The condition of repetitiveness in ILC may affect its applicability in practical scenarios, as many real-world processes are not exactly repetitive. For instance, flexible manufacturing demands mass production of similar goods with different customized sizes and high precision [50]. As studied in [45], slight variations of the desired reference can deteriorate the ILC tracking performance, regardless of their mutual similarity. In this paper, we consider the ILC problem with time-scale-varying tasks. The repetitive reference trajectories with different time scales have the same shape across various time durations, e.g., the production of the same workpiece in different sizes [50]. Unlike the varying trial length problem, in which the operation may stop before a given desired length, the integrity of time-scale-varying tasks is retained by expanding and contracting the time scale,

⋆ This paper was not presented at any IFAC meeting. Corresponding author: Hongfeng Tao.

Email addresses: z.h.zhuang@outlook.com (Zhihe Zhuang), r.a.gonzalez@tue.nl (Rodrigo A. González), taohongfeng@jiangnan.edu.cn (Hongfeng Tao), w.paszke@iee.uz.zgora.pl (Wojciech Paszke), t.a.e.oomen@tue.nl (Tom Oomen).

with the varying tasks being known a priori [40]. Even if the lengths of different iterations vary along the iteration axis, a carefully designed ILC algorithm should conduct perfect tracking, in the ideal case, with respect to the time-scale-varying tasks.

The crucial issue for ILC with time-scale-varying tasks is the corresponding learning. For this purpose, it is straightforward to consider using time shifting for similar tasks with different time scales. In [49], a direct learning control method is developed for a class of iteration-varying trajectories remaining identical in spatial distribution but different in the time-scale sense. In [25,39,9,28], a time-scale shifting operator is proposed in ILC designs for the corresponding learning. Instead of conducting learning updates based on the time elapsed along a trajectory, an ILC methodology based on the spatial path is first proposed in [36], and developed for time-scale-varying tasks in [15]. The proposed time-scale shifting operator suits continuous-time systems, while it is not completely applicable to discrete-time systems with sampling behaviors. No matter how large the sampling frequency is, some tasks with particular time scales still fail in the corresponding learning updates in discrete-time ILC designs.

Further, the extrapolation property of ILC is considered in the design of ILC with basis functions [43,8,45] to tackle varying tasks, where the ILC input of a basic task is learned and projected to varying tasks. This idea is also employed in the cross-coupled ILC controller for different dynamics from different axes to reduce contour error [6], despite the repetitive measured errors from different time domains. The essential idea lies in discovering the unchanged properties and the corresponding transformation. The transformation for varying tasks naturally brings in the iteration-varying disturbance during the discrete-time ILC process. Handling this iteration-varying disturbance is also a vital challenge for discrete-time ILC with varying time scales. In this paper, a time-scale transformation scheme is developed, and the time-scale-varying problem is eventually transformed into a discrete-time ILC problem with iteration-varying disturbances.

Most ILC designs for varying tasks still require system dynamics knowledge, or at least partial knowledge for convergence guarantees [31,10]. System identification techniques are usually applied to acquire the model, where the model here is understood as a parametric system representation such as a state-space description [30]. When the model is acquired from data, model-based ILC can be seen as an *indirect* data-based control approach with separated design procedures [20]. The *direct* data-based ILC involves directly learning the ILC gains from the input/output (I/O) data, where the ILC problem is solved in a unified manner. It suits the case where controlled plants are difficult to model in practice, especially for systems with complex dynamics or interacting in complex environments. Also, fewer data may be required for the *direct* data-based control design [44].

Many promising data-based ILC designs have been developed from the *direct* perspective. In [14,11,22], an optimization-based adaptive ILC design is developed to directly learn ILC gains from I/O data of unknown models, where a dynamical linearization technique is introduced for the estimation of the system parameters. In [23,12], the plant inverse information is updated progressively in a model-free way to directly construct the robust and learning filters of well-established model-based ILC designs. Furthermore, specific intermediate experiments are designed for measuring the gradient of a cost, which is used for data-driven ILC designs [7]. In [19], this approximation process is conducted through offline I/O data and extended to Hammerstein-Wiener systems. To deal with iteration-varying disturbances from external noise or initial state shifting, the extended state observer (ESO) method is introduced in the *direct* data-based ILC design [22,21,35].

In addition, attempts to develop *direct* data-based ILC include approaches that combine reinforcement learning (RL) techniques [41,38,51,27]. The idea of learning from delayed rewards [46] is employed to enable the *direct* data-based paradigm. A Markov decision process (MDP) [42] is built in the iteration domain by considering future iteration costs in the ILC designs, which benefits the convergence performance in ILC [4]. In [41,51,27], the Q-learning approaches are used to solve the associated Bellman equation in a model-free manner. In [38], the actor-critic RL algorithm is employed to solve the Bellman optimality equation to learn the ILC feedforward parameters.

Also, even if RL would endow ILC with flexibility [1], efficiency [29], and autonomy [34], it should be noted that theoretical guarantees of most RL-based ILC design still remain an urgent challenge. Notably, ILC should be experimentally applied for each iteration. The data efficiency and robustness to iteration-varying disturbances (against repetitive restrictions) are of great importance in practical applications.

In this paper, a novel robust *direct* data-based ILC is developed to tackle tasks with varying time scales. By using a time-scale transformation scheme, the time-scale-varying tasks in discrete-time ILC are transformed into an ILC problem with an iteration-varying disturbance. By considering a two-player zero-sum game between the ILC input and the iteration-varying disturbance, a practical off-policy Q-learning-based ILC algorithm is developed for the considered iteration-domain MDP. The associated Bellman equation is efficiently solved in a model-free manner. The robust optimal policy is eventually learned, and it is proved to converge to the model-based solution. Finally, the simulation model of an F-16 aircraft autopilot is employed to verify the effectiveness

of the developed ILC algorithm. In summary, the main contributions are as follows:

1. The practical failure of corresponding learning for the time-scale-varying tasks is demonstrated to exist in discrete-time ILC. This failure is handled by a time-scale transformation scheme and is transformed into a general ILC problem with iteration-varying disturbances for robust learning.
2. By considering a zero-sum game between the ILC input and iteration-varying disturbance, a *direct* data-based off-policy ILC algorithm is developed, which can address the ILC problem with iteration-varying disturbances by introducing a preset disturbance attenuation level. Future iteration costs are considered to enable further learning efficiency, and the off-policy design gives practice guarantees.
3. The data collection requirements for the direct data-based ILC design are assessed and theoretically predicted using Willems' Fundamental Lemma [47]. The convergence of the developed Q-learning-based ILC design is analyzed with theoretical guarantees.

The remainder of the paper is structured as follows. In Section 2, the data-based formulation is given, and the time-scale-varying task is defined. The ILC problem considered in this paper is established through a time-scale transformation scheme. Then, in Section 3, the ILC problem is transformed into an iteration-domain two-player zero-sum game, and the *direct* data-based ILC design is given. The data efficiency and convergence analysis of the developed algorithm are given in Section 4. A numerical simulation is conducted in Section 5 to show the effectiveness of the approach. Finally, the conclusion and discussion on the future work are given in Section 6. Proofs of some technical results are given in the Appendix.

Notations. The notations $\mathbb{N}$ and $\mathbb{R}$ respectively denote the set of natural and real numbers, and $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ respectively denote the sets of n-dimensional real vectors and $n\times m$ real matrices. The notations $L_2^m[a,b]$ and $l_2^m[a,b]$ denote the space of Lebesgue square-integrable and square-summable m-dimensional sequences defined on an interval $[a,b]$, respectively. The notation $\mathrm{diag}\,(\cdot)$ is an operator to construct a square diagonal matrix, and $\mathrm{rank}\,(\cdot)$ represents the rank of a matrix. The notation $\mathcal{N}_n(\mu,\Sigma)$ represents a n-dimensional multivariate Gaussian distribution with mean vector $\mu\in\mathbb{R}^n$ and covariance matrix $\Sigma\in\mathbb{R}^{n\times n}$. For brevity, I_n denotes the $n\times n$ identity matrices, and $\mathbf{0}$ denotes a zero matrix with appropriate dimensions. The expressions $A\succ\mathbf{0}$ and $A\succeq\mathbf{0}$ mean that the square matrix A is positive definite and positive semi-definite, respectively. $\lambda_i(B)$ denotes the ith eigenvalue of the real matrix B.

2 Problem formulation

In this section, the I/O data framework executing repetitive tasks is given first. Then, a time-scale transformation approach is developed for the defined repetitive task with varying time scales. The resulting ILC problem to be addressed is later presented.

2.1 System dynamics

Consider a data sequence generated by the following multi-input multi-output (MIMO) linear time-invariant (LTI) model

$$y_k = G_k u_k + \nu_k, \tag{1}$$

where the subscript k is the iteration index, and y_k and u_k are the m-dimensional output and ℓ-dimensional input data sequences on iteration k, respectively. This paper considers repetitive tracking tasks using only the I/O data. The data sequences are defined on a finite time interval with varying iteration length N_k, i.e.,

$$u_k = \left[u_k^\top(0), u_k^\top(1), \ldots, u_k^\top(N_k-1)\right]^\top \in \mathbb{R}^{\ell N_k},$$
$$y_k = \left[y_k^\top(l), u_k^\top(l+1), \ldots, y_k^\top(N_k-1+l)\right]^\top \in \mathbb{R}^{mN_k},$$

where $l>0$ is the relative degree of the system dynamics in the time domain. For brevity, it is assumed in this paper that $l=1$. For the unknown relative degree $l>1$, refer to Lemma 1 of [48] for a data-driven ILC perspective.

The data transfer matrix $G_k\in\mathbb{R}^{mN_k\times\ell N_k}$ is unknown and parameter-invariant, which is iteration-varying concerning the varying iteration length N_k. Also, $\nu_k\in\mathbb{R}^{mN_k}$ is the unknown external iteration-varying disturbance. Define the reference sequence of the repetitive task with varying time scales as $r_k=[r_k^\top(1), r_k^\top(2), \ldots, r_k^\top(N_k)]^\top\in\mathbb{R}^{mN_k}$.

In this paper, the length of repetitive tasks varies since the continuous-time signals with varying time scales are tracked. The desired set points along the time axis are obtained to construct r_k in the classic ILC formulation. When time scales vary, the set-point values are changed for different iterations. The definition of the time-scale-varying task is given in Definition 1. Denote T_k as the iteration length of the continuous-time reference signal on iteration k.

Definition 1 (Time-scale-varying tasks) *Two signals $r_a^c\in L_2^m[0,T_a]$ and $r_b^c\in L_2^m[0,T_b]$ are identical in the time-scale sense if and only if two conditions are satisfied:*

(1) A time-scaling function $\beta_{ab}:[0,T_a]\to[0,T_b]$ is a continuously differentiable bijection, where $\beta_{ab}(0)=0$ and $\beta_{ab}(T_a)=T_b$.

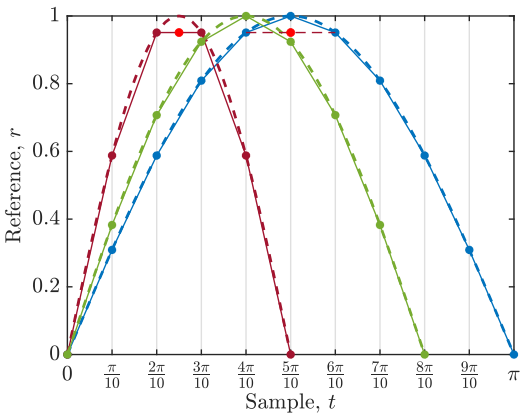

Fig. 1. The sampled-data ILC problem when tracking time-scale-varying tasks. Consider tracking a repetitive sinusoid profile with different time scales in the continuous-time domain (dashed lines). The sampling time is $\frac{\pi}{10}$. The discrete-time ILC design suffers from two issues: 1) different set points at the same time instant for tracking, e.g., when $t = \frac{\pi}{10}$, and 2) the iteration-varying uncertainty due to the unity time-scale transformation.

(2) For all $t_a \in [0, T_a]$, there exists a unique $t_b \in [0, T_b]$ such that $r_a^c(t_a) = r_b^c(\beta_{ab}(t_a)) = r_b^c(t_b)$.

The tracking task in Definition 1 is a continuous-time signal. An example of the time-scale-varying tasks is given as dashed profiles in Fig. 1. The three dashed sinusoid profiles belong to the time-scale-varying tasks as stated in Definition 1. The reference signals have the same shape but different durations, and the time-scaling operator in this example is a linear mapping $t_b = \beta_{ab}(t_a) = \frac{T_b}{T_a} t_a$.

ILC is employed in this paper to complete the time-scale-varying tasks. Note that the standard ILC methods, e.g., [10], typically use the discrete-time measurements of I/O data with a fixed sampling frequency, which is consistent with the sampling frequency the ILC controller operates at. The continuous-time signal is often parameterized in time with a series of desired set points for the discrete-time controller to track. Tracking the parametric set points in time can effectively complete the tracking task. See sampled-data control and its ILC applications [13,37].

However, the performance of traditional discrete-time ILC design suffers when tracking iteration-varying set points to complete the time-scale-varying tasks as illustrated in Fig. 1. Note that the learning process of ILC approaches demands strict correspondence between two successive iterations. In Fig. 1, the set points at the same sampling time instant, e.g., $t = \frac{\pi}{10}$, are not identical.

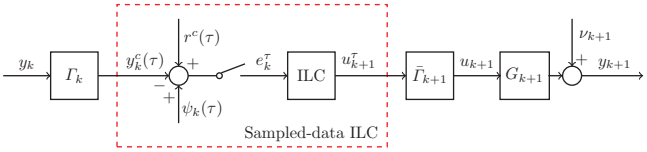

Fig. 2. The time-scale transformation scheme. The sampled data from the previous iteration y_k is transformed into a dimensionless continuous-time signal y_k^c by the time-scale transformation operator Γ_k. Then, a sampled-data ILC algorithm is developed to produce a discrete-time signal u_{k+1}^τ, which is retransformed by the discrete time-scale transformation operator $\bar{\Gamma}_{k+1}$ into an ILC input u_{k+1} for the next tracking task. Both ψ_k and ν_k are iteration-varying disturbances: ψ_k comes from time-scale-varying tasks, and ν_k is the external noise.

2.2 Time-scale transformation scheme

A time-scale transformation scheme is incorporated in the ILC design for correspondent learning. The time-scale transformation scheme is given in Fig. 2. Define the time-scale transformation operator Γ_k as

$$\Gamma_k : l_2^m[0, N_k] \to L_2^m[0, 1] \tag{2}$$

to map the output signals in the absolute time coordinates $t \in [0, N_k]$ into signals in the dimensionless unit coordinates $\tau \in [0, 1]$, i.e., for $t \in [0, N_k]$,

$$y_k^c(\tau) = \Gamma_k(y_k)(t) = y_k(t), \tau \in [t/N_k,\ (t+1)/N_k)\,. \tag{3}$$

This transformation is introduced for the purpose of correspondent learning. In particular, the non-identical iteration lengths are normalized, and thus the ILC controller only needs to track a stationary profile r^c in the dimensionless τ-coordinates.

The ILC design under the time-scale transformation scheme is a sampled-data control problem. As in Fig. 2, the sampled output data on iteration k is transformed into a continuous-time signal y_k^c in the dimensionless τ-coordinates by employing the time-scale transformation operator Γ_k. A discrete-time ILC control signal u_{k+1} is generated by the mapping operator $\bar{\Gamma}_{k+1}$, where $\bar{\Gamma}_{k+1} : l_2^\ell[0, 1] \to l_2^\ell[0, N_{k+1}]$, maps the dimensionless signals to the ILC control signals, i.e.,

$$u_{k+1}(t) = \bar{\Gamma}_{k+1}(u_{k+1}^\tau)(\tau) = u_{k+1}^\tau(\tau),\ t = N_{k+1}\tau. \tag{4}$$

Remark 1 *When the time scale of the next trial is significantly shortened, the high-frequency reference can exceed the bandwidth of the controlled plant. Integral windup with respect to the actuator may occur, and the unmodelled high-frequency response may result in a phase shift. Therefore, the range of variation in the time-scale-varying task should be carefully set based on the practical limitations. If it is unavoidable to reach input saturation, one can actively add a constraint on*

the feedforward input to avoid integral windup, where further analysis on the nonlinear dynamics subject to the actuator saturation should be considered.

Then, the optimal sampled-data ILC problem is defined as follows:

Definition 2 (Optimal sampled-data ILC problem) *Given a pre-defined ILC cost function $J^S_{k+1}(u, r^c, y^c_{k+1})$ with respect to the continuous-time reference signal r^c, the optimal sampled-data ILC problem is to find the optimal discrete-time input vector*

$$u_{k+1} = \arg\min_u J^S_{k+1}(u, r^c, y^c_{k+1}). \tag{5}$$

Note that the time-scale transformation scheme is developed for the corresponding learning from the true data in time-scale-varying tasks. When ILC is applied in the dimensionless unit coordinates, all previous time scales can be recorded and used for the input update in ILC process. However, when the discrete-time ILC with sampling behavior is employed to approximate sampled-data ILC, an inappropriate sampling rate will lead to an approximation error in the transformation process. To address this issue, it is better to choose the integer multiples of most time scales. For instance, choosing the least common multiple of time scales of all possible trials can fundamentally avoid the approximation error. In this regard, the sampled-data ILC problem can be handled by the solution to the following discrete-time ILC problem.

Definition 3 (Optimal discrete-time ILC problem) *Given a pre-defined ILC cost function $J_{k+1}(u, r^d, y_{k+1})$, the optimal discrete-time ILC problem is to find the optimal discrete-time input vector*

$$u_{k+1} = \arg\min_u J_{k+1}(u, r^d, y_{k+1}), \tag{6}$$

where r^d is a discrete-time sampled signal of r^c with certain sampling frequency.

In addition to the approximation error caused by the inappropriate sampling rate, other iteration-varying factors also occur when employing the time-scale transformation scheme for the purpose of corresponding learning. This ILC problem with iteration-varying disturbances is elaborated in the next subsection.

2.3 ILC problem with iteration-varying disturbances

The time-scale-varying problem is transformed into an ILC problem with iteration-varying disturbances. The iteration-varying disturbances originate from two aspects.

The first one lies in the time-scale transformation of the sampled output data y_k. To illustrate, consider the transformation case from $[0, \frac{\pi}{2}]$ to $[0, 1]$ in Fig. 1. No sampling point for this scale corresponds to $t = \frac{\pi}{4}$ under the sampling time $\frac{\pi}{10}$, which yields that the output data of the halfway point used for the ILC update is inaccurate. Suppose the transformed output signal is used for the ILC update with duration $[0, \pi]$. In that case, the outputs of the halfway point used for learning are inaccurate due to the sampling limitation, which will introduce iteration-varying transformation errors in time-scale-varying tasks. This transformation error is denoted by ψ_k in Fig. 2. The approximation error mentioned in Section 2.2 can also be included in ψ_k.

Another non-repetitive factor is the external unknown disturbance ν_k. When executing time-scale-varying tasks with mutual similarity in practice, the external environmental factors may be iteration-varying as occurring in, e.g., underwater robots [25] and kite-based marine hydrokinetic systems [15]. Both kinds of iteration-varying disturbances arise from the goal of using ILC to complete time-scale-varying tasks.

Then, we accurately describe the iteration-varying disturbances in error dynamics. For time-scale-varying tasks, the tracking error in the τ-coordinates on iteration k, denoted by e^τ_k, is given in Fig. 2 as

$$e^\tau_k = \mathcal{S}(r^c - y^c_k + \psi_k) \tag{7}$$

with

$$y^c_k = G^\tau u^\tau_k + \Gamma_k \nu_k, \tag{8}$$

where $\mathcal{S} : L^\ell_2[0, 1] \to l^\ell_2[0, 1]$ is a sampling operator, and $G^\tau = \Gamma_k G_k \bar{\Gamma}_k$, which is an iteration-invariant linear operator. The tracking error dynamic along the iteration domain is derived by

$$\begin{aligned} e^\tau_{k+1} &= \mathcal{S}(r^c - G^\tau u^\tau_{k+1} - \Gamma_{k+1}\nu_{k+1} + \psi_{k+1}) \\ &= \mathcal{S}(r^c - G^\tau u^\tau_k - \Gamma_k \nu_k + \psi_k) - \mathcal{S}G^\tau \Delta u^\tau_{k+1} \\ &\quad + \mathcal{S}\Gamma_k(\nu_k - \nu_{k+1}) + \mathcal{S}(\psi_{k+1} - \psi_k) \\ &= e^\tau_k - \mathcal{S}G^\tau \Delta u^\tau_{k+1} + w_{k+1}, \end{aligned} \tag{9}$$

where $\Delta u^\tau_{k+1} = u^\tau_{k+1} - u^\tau_k$ and $w_{k+1} = \mathcal{S}\Gamma_k(\nu_k - \nu_{k+1}) + \mathcal{S}(\psi_{k+1} - \psi_k)$.

Note that the tracking error can converge to zero if there are no iteration-varying factors, i.e., $w_{k+1} = 0$. The ILC problem considered in this paper is given in Definition 4.

Definition 4 (The ILC problem) *Given the iteration-varying disturbance w_{k+1}, the ILC problem in this paper consists in developing a robust direct data-based ILC update law*

$$u_{k+1} = f(u_k, e_k) \tag{10}$$

for the unknown linear system (1) such that the tracking error converges in norm as $k \to \infty$.

In the next section, a novel robust data-based design for the ILC problem stated in Definition 4 is given.

3 Robust direct data-based ILC design

In this section, the ILC challenge stated in Section 2.3 is transformed into an iteration-domain zero-sum game problem. For brevity, the superscript τ in the error dynamics (9), which represents the τ-coordinates, is omitted in the following, i.e.,

$$e_{k+1} = e_k - G\Delta u_{k+1} + w_{k+1}, \tag{11}$$

where $G \in \mathbb{R}^{mN\times \ell N}$ and N is the fixed number samples in the τ-coordinates. We assume that the iteration-varying disturbance signal has finite energy over an infinite iteration interval, i.e.,

$$\sum_{k=0}^{\infty} w_k^\top w_k < \infty. \tag{12}$$

Definition 5 *For all w_k satisfying (12), the ILC problem in this paper is to design an ILC algorithm such that the $l_2\,[0,\infty)$ gain of error system (11) is less than or equal to a preset disturbance attenuation level γ satisfying*

$$\sum_{k=0}^{\infty} \left(e_k^\top Q e_k + \Delta u_k^\top R \Delta u_k\right) \le \gamma^2 \sum_{k=0}^{\infty} w_k^\top w_k, \tag{13}$$

where the weighting matrices $Q \succ \mathbf{0}$ and $R \succeq \mathbf{0}$ are symmetric with appropriate dimensions.

With Definition 5, the ILC problem is now transformed into an iteration-domain $\mathcal{H}_\infty$ control problem: find an ILC design such that the error system (11) converges in the iteration domain while the disturbance attenuation condition (13) is satisfied. This problem is solved in a direct data-based perspective by being transformed into a zero-sum game problem.

3.1 Zero-sum game formulation

Based on (13), the data-based design employs a cost function that takes the performance of future iterations into consideration. The cost function is given by

$$\begin{aligned} J_{k+1}\left(\Delta u_{k+1}\right) = \sum_{i=k}^{\infty} \rho^{i-k} \big(& e_{i+1}^\top Q e_{i+1} \\ & + \Delta u_{i+1}^\top R \Delta u_{i+1} - \gamma^2 w_{i+1}^\top w_{i+1}\big), \end{aligned} \tag{14}$$

where $\rho \in (0,1]$ is a discounted factor to measure the future performance. Increasing ρ can speed up the trial convergence as studied in [4].

Remark 2 *Compared to the norm optimal ILC (NOILC), which only considers the cost of the next iteration, learning from future costs can achieve faster convergence speed [4]. Moreover, learning from delayed rewards benefits the policy learning process where only I/O data is used [46].*

Define a value function with respect to the state of the current iteration, i.e., e_k, as

$$V(e_k) = \sum_{i=k}^{\infty} \rho^{i-k} c\left(e_i, \Delta u_{i+1}, w_{i+1}\right), \tag{15}$$

where $c\left(e_i, \Delta u_{i+1}, w_{i+1}\right) = e_i^\top Q e_i + \rho \Delta u_{i+1}^\top R \Delta u_{i+1} - \rho\gamma^2 w_{i+1}^\top w_{i+1}$.

Lemma 1 *Minimizing the cost function (14) with respect to the control sequence $\{\Delta u_{k+1}\}_{k=0}^{\infty}$ is equivalent to minimizing the value function $V(e_k)$ defined in (15).*

Proof: The cost function (14) is reformulated as

$$\begin{aligned} & J_{k+1}\left(\Delta u_{k+1}\right) \\ & = -\rho^{-1} e_k^\top Q e_k + \rho^{-1}(e_k^\top Q e_k + \rho \Delta u_{k+1}^\top R \Delta u_{k+1} \\ & \quad - \rho\gamma^2 \Delta w_{k+1}^\top \Delta w_{k+1}) + \cdots \\ & = -\rho^{-1} e_k^\top Q e_k + \rho^{-1} c\left(e_k, \Delta u_{k+1}, w_{k+1}\right) \\ & \quad + \rho^0 c\left(e_{k+1}, \Delta u_{k+2}, w_{k+2}\right) + \cdots \\ & = -\rho^{-1} e_k^\top Q e_k + \rho^{-1} \sum_{i=k}^{\infty} \rho^{i-k} c\left(e_i, \Delta u_{i+1}, w_{i+1}\right) \\ & = -\rho^{-1} e_k^\top Q e_k + \rho^{-1} V(e_k), \end{aligned} \tag{16}$$

where the term $-\rho^{-1} e_k^\top Q e_k$ is known. □

Then, the goal is to determine an ILC policy to generate a control sequence $\{\Delta u_{k+1}\}_{k=0}^{\infty}$ that minimizes the value function $V(e_k)$ in the presence of an iteration-varying disturbance w_{k+1}. This process is an iteration-domain MDP. The MDP problem has a robust optimal solution by restating it as a two-player zero-sum game. The control input player Δu_{k+1} tries to find a policy to minimize the value function $V(e_k)$ as $k \to \infty$, while the disturbance player w_{k+1} seeks to maximize $V(e_k)$, i.e.,

$$\begin{aligned} & \min_{\Delta u_{k+1}} \max_{w_{k+1}} V(e_k) \\ & \text{s.t. } e_{k+1} = e_k - G\Delta u_{k+1} + w_{k+1}. \end{aligned} \tag{17}$$

By Bellman's Principle of Optimality, the solution to the following Bellman equation is the saddle point of (17):

$$V(e_k) = c\left(e_k, \Delta u_{k+1}, w_{k+1}\right) + \rho V(e_{k+1}), \tag{18}$$

whose model-based solution is given in the following proposition.

Proposition 1 *Given a known model G, the Bellman equation (18) has the unique analytic solution pairs*

$$\Delta u_{k+1} = L_u^* e_k, \tag{19}$$
$$w_{k+1} = L_w^* e_k, \tag{20}$$

where

$$\begin{aligned} L_u^* =& \left[R + G^\top P G + G^\top P \left(\gamma^2 I_{mN} - P\right)^{-1} P G\right]^{-1} \\ &\times \left[G^\top P + G^\top P \left(\gamma^2 I_{mN} - P\right)^{-1} P\right], \\ L_w^* =& \left[\gamma^2 I_{mN} - P + P G \left(R + G^\top P G\right)^{-1} G^\top P\right]^{-1} \\ &\times \left[P - P G \left(R + G^\top P G\right)^{-1} G^\top P\right] \end{aligned} \tag{21}$$

and P is the solution of the discounted game algebraic Riccati equation (DGARE)

$$\begin{aligned} P =& \rho P + Q - \rho \begin{bmatrix} -PG & P \end{bmatrix} \times \\ & \begin{bmatrix} R + G^\top P G & -G^\top P \\ -PG & P - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} -G^\top P \\ P \end{bmatrix}. \end{aligned} \tag{22}$$

Proof: See Appendix A. □

Proposition 1 gives a model-based solution to the Bellman equation (18). This solution requires the exact model information of the system dynamics and the controllable disturbance input w_{k+1}, which hinders its practical applications. In the next subsection, an efficient model-free algorithm, which only needs I/O data from repetitive experiments, is developed to solve the ILC problem stated in Definition 4.

3.2 Direct data-based ILC design

The goal in this subsection is to develop a data-based approach that updates the target policy $L = \{L_u, L_w\}$ from an initial stabilizing policy to the robust optimal policy in Proposition 1 without any model knowledge.

Denote the value function under a policy L by $V^L(e_k)$, which is defined as the cost that uses $\Delta u_{k+1} = L_u e_k$ and $w_{k+1} = L_w e_k$ from iteration k onward, i.e.,

$$V^L(e_k) = \sum_{i=k}^{\infty} \rho^{i-k} c\left(e_i, L_u e_i, L_w e_i\right). \tag{23}$$

Then, define the Q-function under policy L as

$$\mathbf{Q}^L(e_k, \Delta u_{k+1}, w_{k+1}) = c\left(e_k, \Delta u_{k+1}, w_{k+1}\right) + \rho V^L(e_{k+1}), \tag{24}$$

which is used to evaluate the cost that uses policy L from iteration $k+1$ onward for arbitrary input Δu_{k+1} and w_{k+1} generated on iteration k. In this case, we have

$$\mathbf{Q}^L(e_k, L_u e_k, L_w e_k) = V^L(e_k), \tag{25}$$

which yields the recursive form

$$\begin{aligned} \mathbf{Q}^L(e_k, \Delta u_{k+1}, w_{k+1}) =& c\left(e_k, \Delta u_{k+1}, w_{k+1}\right) \\ &+ \rho \mathbf{Q}^L(e_{k+1}, L_u e_{k+1}, L_w e_{k+1}). \end{aligned} \tag{26}$$

The Q-function takes the quadratic form as

$$\mathbf{Q}^L(e_k, \Delta u_{k+1}, w_{k+1}) = z_k^\top \Phi^L z_k, \tag{27}$$

where $z_k = [e_k^\top, \Delta u_{k+1}^\top, w_{k+1}^\top]^\top$, and the symmetric matrix $\Phi^L \in \mathbb{R}^{(2m+\ell)N \times (2m+\ell)N}$ takes the element form

$$\Phi^L \triangleq \begin{bmatrix} \phi_{ee} & \phi_{eu} & \phi_{ew} \\ \phi_{ue} & \phi_{uu} & \phi_{uw} \\ \phi_{we} & \phi_{wu} & \phi_{ww} \end{bmatrix}. \tag{28}$$

When the robust optimal policy is learned, Φ^L takes the model-based form

$$\Phi^L = \begin{bmatrix} Q + \rho P & -\rho P G & \rho P \\ -\rho G^\top P & \rho\left(G^\top P G + R\right) & -\rho G^\top P \\ \rho P & -\rho P G & \rho\left(P - \gamma^2 I_{mN}\right) \end{bmatrix}. \tag{29}$$

Subject to the unknown model G, Φ^L should be estimated by using the I/O data, which can directly generate the data-based policy $L = \{L_u, L_w\}$ as shown in the following.

Similar to Proposition 1, let $\frac{\partial \mathbf{Q}^L}{\partial \Delta u_{k+1}} = \mathbf{0}$ and $\frac{\partial \mathbf{Q}^L}{\partial w_{k+1}} = \mathbf{0}$ yielding

$$\begin{cases} \Delta u_{k+1} = -\phi_{uu}^{-1}\phi_{ue} e_k - \phi_{uu}^{-1}\phi_{uw} w_{k+1}, \\ w_{k+1} = -\phi_{ww}^{-1}\phi_{we} e_k - \phi_{ww}^{-1}\phi_{wu} \Delta u_{k+1}, \end{cases} \tag{30}$$

which yields $\Delta u_{k+1} = L_u e_k$ and $w_{k+1} = L_w e_k$ with

$$L_u = (\phi_{uu} - \phi_{uw}\phi_{ww}^{-1}\phi_{wu})^{-1}(\phi_{uw}\phi_{ww}^{-1}\phi_{we} - \phi_{ue}), \tag{31}$$
$$L_w = (\phi_{ww} - \phi_{wu}\phi_{uu}^{-1}\phi_{uw})^{-1}(\phi_{wu}\phi_{uu}^{-1}\phi_{ue} - \phi_{we}). \tag{32}$$

Since the components of Φ^L should satisfy the Bellman equation (26), it follows that

$$z_k^\top \Phi^L z_k = z_k^\top W z_k + \rho \chi_{k+1}^\top \Phi^L \chi_{k+1}, \tag{33}$$

where $W = \mathrm{diag}(Q, \rho R, \rho\gamma^2 I_{mN})$, and $\chi_{k+1} = [e_{k+1}^\top, (L_u e_{k+1})^\top, (L_w e_{k+1})^\top]^\top$. Based on (31), (32),

Algorithm 1 Direct data-based Q-learning ILC

1. **Initialization:** Given the linear system (11) with iteration-varying disturbance w_{k+1}, the trajectory reference r^d, the initial ILC input u_0 and the error e_0, a probing noise vector $\zeta_k \sim \mathcal{N}_{\ell N}(\mu, \Sigma)$ where Σ is non-degenerate, the weighting matrices Q and R, the discounted factor $\rho \in (0, 1]$, the disturbance attention level γ, a stabilizing Arimoto-type gain L_u^0, and the initial L_w^0. Set $k, i = 0$.
2. Apply $\Delta u_{k+1} = L_u^0 e_k + \zeta_{k+1}$ to the system (11) until $\eta = (2m + \ell) N$ iterations of linearly independent vectors z_k are collected.
3. Sort these η vectors with index $\{k_1, k_2, \cdots, k_\eta\}$, i.e., $z_{k_j} = [e_{k_j}^\top, \Delta u_{k_j+1}^\top, w_{k_j+1}^\top]^\top$, $j = 1, \cdots, \eta$, to construct
$$Z = \begin{bmatrix} z_{k_1} & z_{k_2} & \cdots & z_{k_\eta} \end{bmatrix} \in \mathbb{R}^{\eta \times \eta}. \tag{34}$$
4. **Repeat:**
   - Select the corresponding $\chi_{k_j+1}^i = [e_{k_j+1}^\top, (L_u^i e_{k_j+1})^\top, (L_w^i e_{k_j+1})^\top]^\top$ to construct
$$E_i = \begin{bmatrix} \chi_{k_1+1}^i & \chi_{k_2+1}^i & \cdots & \chi_{k_\eta+1}^i \end{bmatrix} \in \mathbb{R}^{\eta \times \eta}. \tag{35}$$
   - Solve the following data-based equation to get $\Phi^{L^{i+1}}$, i.e.
$$Z^\top \Phi^{L^{i+1}} Z = Z^\top W Z + \rho E_i^\top \Phi^{L^{i+1}} E_i. \tag{36}$$
   - Update the learning policy by (31) and (32) with superscript $i + 1$.
   - Set $i \to i + 1$.
5. **Until:** $\left\| L_u^{i+1} - L_u^i \right\| < \epsilon_u$ and $\left\| L_w^{i+1} - L_w^i \right\| < \epsilon_w$ for some $\epsilon_u, \epsilon_w > 0$.
6. Set $L_u^* = L_u^{i+1}$.
7. **Return:** The robust optimal ILC policy L_u^*.

and (33), an off-policy direct data-based Q-learning ILC algorithm is given in Algorithm 1.

By utilizing the collected data, Algorithm 1 provides an off-policy data-based robust ILC algorithm for the ILC problem stated in Section 2.3. In the off-policy Algorithm 1, the behavior policy employs the Arimoto-type ILC design, which is used to generate data first. The target policy is $L^i = \left\{L_u^i, L_w^i\right\}$, which is evaluated and improved as the policy iteration index i increases rather than the ILC process index k.

Remark 3 *In this paper, the zero-sum game perspective is introduced in the ILC designs based on the trial independence of ILC. The off-policy Algorithm 1 possesses two advantages for practical data-based ILC applications. One is that $\Delta u_{k+1} = L_u^{i+1} e_k$ will never be applied to the actual ILC process during the policy iteration for safety and efficiency reasons. Another is that it is not necessary to update the disturbance input w_{k+1} compared to the on-policy counterparts as discussed in [3,26].*

Remark 4 *The computation cost of Algorithm 1 consists of three parts: applying the Arimoto-type ILC in every ILC iteration k, selecting η linearly independent vectors to construct Z, and solving the generalized Sylvester equation (36) for every policy iteration i. The selection of η linearly independent vectors can be solved by transforming the collected data into the row echelon form via Gaussian elimination. This step could be computationally expensive with $\mathcal{O}(K(2m + \ell)^2 N^2)$, although it only needs to be performed once and offline. Solving the generalized Sylvester equation (36) during the policy iteration process has $\mathcal{O}((2m + \ell)^3 N^3)$. Nonetheless, efficient numerical methods can be employed [17], e.g., methods implemented in the Matlab command* dlyap. *Since major computational loads are offline and happen between two successive ILC trials, Algorithm 1 generally requires no additional hardware requirements compared to traditional ILC controllers in real-time applications.*

Note that the data can be generated by any ILC method, as long as it yields a stable process. The iterative learning process is essential but not the ultimate goal. In this sense, the Arimoto-type gain is a straightforward selection for the initial policy design. For a system represented by a state-space description (A, B, C, D), it is trivial to test its convergence condition by $\left|\lambda_i(I_\ell - L_u^0(j, j)CB)\right| < 1$ for all i, j, where $L_u^0(j, j)$ denotes the jth diagonal blocks of L_u^0 and CB is the input-output coupling matrix. For the case CB is of full column rank, one can recover CB by simply recording the response at time step one for a unit pulse input at step zero [31] and extended to every input for MIMO systems. For nonminimum phase systems, model-free ILC designs are also available [24]. In the next section, a comprehensive analysis of Algorithm 1 is given.

4 Theoretical analysis

In this section, the theoretical guarantee of data efficiency and convergence property of Algorithm 1 are presented. Willems' Fundamental Lemma [47] is first introduced to clear the requirement of collecting data in Algorithm 1. The convergence of the data-based design is later proved.

4.1 Data efficiency

Since the repetitive task lasts a finite time interval, collecting sufficient data along the time axis is not always feasible. This subsection quantitatively discusses how many iterations are needed for Algorithm 1 to collect data in the iteration domain. In particular, it is shown that operating with a persistently exciting (PE) input

of sufficient order, at most $\mathcal{K} = (mN+1)\ell N + mN$ iterations will be required for constructing a full row rank matrix Z for robust learning in Algorithm 1.

Denote the error sequence in the iteration domain by $\{e_k\}_{k=0}^{K-1}$, where $K < \infty$ is the number of iterations and $\{e_k\}_{k=0}^{K-1} = \{e_0, e_1, \cdots, e_{K-1}\}$. Define the Hankel matrix of depth J with respect to $\{e_k\}_{k=0}^{K-1}$ as

$$H_J\left(e_{[0,K-1]}\right) = \begin{bmatrix} e_0 & e_1 & \cdots & e_{K-J} \\ e_1 & e_2 & \cdots & e_{K-J+1} \\ \vdots & \vdots & \ddots & \vdots \\ e_{J-1} & e_J & \cdots & e_{K-1} \end{bmatrix}, \tag{37}$$

where $e_{[0,K-1]}$ is defined as

$$e_{[0,K-1]} = \begin{bmatrix} e_0^\top & e_1^\top & \cdots & e_{K-1}^\top \end{bmatrix}^\top. \tag{38}$$

By the definition of (37), Definition 6 gives a rank condition ensuring an input sequence is PE.

Definition 6 *An input sequence $\{u_k\}_{k=0}^{K-1}$, where $u_k \in \mathbb{R}^{\ell N}$, is PE of order J if* $\operatorname{rank}(H_J(u_{[0,K-1]})) = \ell N J$.

Willems' Fundamental Lemma is introduced by the following result to show how the data efficiency of Algorithm 1 is ensured.

Lemma 2 (Corollary 2, [47]) *Applying an $(\ell+m)N$-dimensional input sequence $\{\bar{u}_k\}_{k=0}^{K-1}$, which is PE of order $mN+J$, to a controllable linear system for collecting output $\{e_k\}_{k=0}^{K-1}$, where $e_k \in \mathbb{R}^{mN}$, we have*

$$\operatorname{rank}\left(\begin{bmatrix} H_1(e_{[0,K-J+1]}) \\ H_J(\bar{u}_{[0,K-1]}) \end{bmatrix}\right) = mN + (\ell+m)NJ. \tag{39}$$

It then follows from the Rouché-Capelli theorem that any J-long I/O trajectory $\{e_k, \bar{u}_k\}_{k=0}^{J-1}$ of the controllable system can be represented by the collected data under PE input of sufficient order [16]. In other words, the optimal policy can be directly learned based on the collected I/O data.

Willems' Fundamental Lemma can be applied in cases where the output data is perturbed by noise [18]. According to the zero-sum game design where w_{k+1} is also seen as an input of the system, consider the following representation of the system (11), i.e.,

$$e_{k+1} = e_k + \begin{bmatrix} -G & I_{mN} \end{bmatrix} \bar{u}_k, \tag{40}$$

where $\bar{u}_k = [\,\Delta u_{k+1}^\top,\ w_{k+1}^\top\,]^\top$ and (40) is controllable since G has full row rank when the relative degree $l = 1$. By Lemma 2, it suffices to ensure $\{\bar{u}_k\}_{k=0}^{K-1}$ is PE of order $mN+1$, and hence yielding

$$\operatorname{rank}\left(\begin{bmatrix} H_1(e_{[0,K-1]}) \\ H_1(\Delta u_{[1,K]}) \\ H_1(w_{[1,K]}) \end{bmatrix}\right) = (2m+\ell)\,N. \tag{41}$$

Then, there will always be $(2m+\ell)N$ linearly independent vectors $z_k = [e_k^\top, \Delta u_{k+1}^\top, w_{k+1}^\top]^\top$ to construct Z for robust learning in Algorithm 1. Assume $\{w_{k+1}\}_{k=0}^{K-1}$ is PE of order at least $mN+1$. It is given that $\{\bar{u}_k\}_{k=0}^{K-1}$ is PE of order $mN+1$ by showing the input sequence $\{\Delta u_{k+1}\}_{k=0}^{K-1}$ in Algorithm 1 is also PE of order $mN+1$.

Theorem 1 *The extended input sequence $\{\bar{u}_k\}_{k=0}^{K-1}$ of the system (40) is PE of order $mN+1$.*

Proof: To prove, it is required that $H_{mN+1}(\bar{u}_{[0,K-1]})$ has full row rank. Since $\{w_{k+1}\}_{k=0}^{K-1}$ is PE of order at least $mN+1$, it is sufficient to prove $H_{mN+1}(\Delta u_{[1,K]})$ has full row rank. According to Step 2 in Algorithm 1, we have

$$\begin{aligned} &\operatorname{rank}\left(H_{mN+1}(\Delta u_{[1,K]})\right) \\ &\quad = \operatorname{rank}\left(H_{mN+1}(L_u^0 e_{[0,K-1]} + \zeta_{[1,K]})\right). \end{aligned} \tag{42}$$

Since the covariance matrix Σ of the multivariate Gaussian distribution is chosen to be non-degenerate in Algorithm 1, all elements of the probing noise ζ_{k+1} are linearly independent. Then, $H_{mN+1}(\zeta_{[1,K]})$ has full row rank, which directly implies that $H_{mN+1}(\Delta u_{[1,K]})$ has full row rank. □

Note that the necessary requirement for a matrix to be full row rank is that the number of rows should be no more than the number of columns. Therefore, $K \geq (mN+1)\ell N + mN$ holds for the persistence of excitation order to be $mN+1$ in Theorem 1. However, ensuring $\{\bar{u}_k\}_{k=0}^{K-1}$ is PE of order $mN+1$ is not a necessary condition for (41) to hold. In practice, it might be sufficient to conduct only $(2m+\ell)N+1$ iterations for the successful robust learning in Algorithm 1.

4.2 Convergence of the off-policy Q-learning algorithm

The convergence of L_u^i and L_w^i in the developed off-policy ILC algorithm is given in this subsection. The $\mathcal{H}_\infty$ control problem solved in a data-driven manner using the zero-sum game perspective has been studied in the literature, e.g., [32,33]. This proof extends the existing results [32] to reveal the convergence of the model-free algorithm for the DGARE in (22). Two issues arise in the proof of the new result:

1. The discounted factor ρ is considered.
2. Extra input w_{k+1} is considered in the off-policy Q-learning formulation.

Notice in Algorithm 1 that there exists a model-based relationship between χ^i_{k+1} and z_k, i.e.

$$\chi^i_{k+1}=\begin{bmatrix} e_{k+1} \\ L^i_u e_{k+1} \\ L^i_w e_{k+1} \end{bmatrix}=\begin{bmatrix} I_{mN} & -G & I_{mN} \\ L^i_u & -L^i_u G & L^i_u \\ L^i_w & -L^i_w G & L^i_w \end{bmatrix}\begin{bmatrix} e_k \\ \Delta u_{k+1} \\ w_{k+1} \end{bmatrix}\triangleq\Theta_i z_k, \tag{43}$$

where $\Theta_i \in \mathbb{R}^{(2m+\ell)N\times(2m+\ell)N}$. Then, the stabilizing property of the defined matrix Θ_i, i.e., all its eigenvalues strictly stay inside the unit circle, is given first in the following technical lemma.

Lemma 3 *The matrix Θ_i is stable if and only if the matrix $I_{mN} - GL^i_u + L^i_w$ is also stable.*

Proof: Consider a matrix transformation by

$$\Sigma_i^{-1}\Theta_i\Sigma_i=\begin{bmatrix} I_{mN} & & \\ -L^i_u & I_{\ell N} & \\ -L^i_w & \mathbf{0} & I_{mN} \end{bmatrix}\begin{bmatrix} I_{mN} & -G & I_{mN} \\ L^i_u & -L^i_u G & L^i_u \\ L^i_w & -L^i_w G & L^i_w \end{bmatrix}$$
$$\times\begin{bmatrix} I_{mN} & & \\ L^i_u & I_{\ell N} & \\ L^i_w & \mathbf{0} & I_{mN} \end{bmatrix}=\begin{bmatrix} I_{mN}-GL^i_u+L^i_w & -G & I_{mN} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

which has diagonal blocks $I_{mN} - GL^i_u + L^i_w$ and $\mathbf{0}$. □

It is straightforward to know that the initial policy $L^0 = \{L^0_u, L^0_w\}$ is stabilizing since the Arimoto-type gain L^0_u satisfies $|\lambda_i(I_\ell - L^0_u(j,j)CB)| < 1$ and $L^0_w = \mathbf{0}$ in Algorithm 1. Then, given the initial stabilizing policy L^0, applying Algorithm 1 yields the stabilizing robust optimal policy as proved in the following.

Theorem 2 *Both L^{i+1}_u and L^{i+1}_w in Algorithm 1 converge to the optimum as $i \to \infty$, i.e.,*

$$\lim_{i\to\infty} L^{i+1}_u = L^*_u,\ \lim_{i\to\infty} L^{i+1}_w = L^*_w, \tag{44}$$

where L^*_u and L^*_w are the optimal solutions given in (21).

Proof: See Appendix B. □

Theorem 2 gives the convergence of the off-policy Q-learning algorithm. Its resulting policy is also a convergent ILC gain since L^*_u is also stabilizing and hence $|\lambda_i(I_\ell - L^*_u(j,j)CB)| < 1$ for all i, j.

In the next section, the simulation results are given to show the effectiveness of Algorithm 1.

5 Numerical simulation

In this section, the simulation model of an F-16 aircraft autopilot as in [26] is employed to show the effectiveness of the developed ILC method. The time-scale transformation scheme in Section 2.2 and Algorithm 1 is applied to the LTI F-16 aircraft autopilot system to track repetitive reference signals with varying time scales.

The discrete-time state-space representation of the F-16 aircraft autopilot as given by the system matrices as follows:

$$A=\begin{bmatrix} 0.906488 & -0.0816012 & -0.0005 \\ 0.074349 & 0.90121 & -0.000708383 \\ 0 & 0 & 0.132655 \end{bmatrix},$$
$$B=\begin{bmatrix} -0.00150808 \\ -0.0096 \\ 0.867345 \end{bmatrix}, C=\begin{bmatrix} 0 & 0 & 0.5 \end{bmatrix}, D=\begin{bmatrix} 0 \end{bmatrix} \tag{45}$$

and the sampling period is 0.1s. The state vector $x_k(t)$ includes the angle of attack, the pitch rate, and the elevator deflection angle, respectively. In this paper, the elevator deflection angle is considered as the output $y_k(t)$.

The desired reference signals are set to vary for different iterations but have the same shape in the continuous-time domain, which satisfies Definition 1. The trajectory reference in τ-coordinate is $r^c(\tau) = 3\sin(\frac{\pi}{10}\tau)$. The discrete-time reference vector r^d for Algorithm 1 is acquired with sampling period 0.1s. The iteration-varying references in Fig. 3 are obtained by $r_k(t) = \bar{\Gamma}_k(r^d)(\tau)$ as in (4). The output sequence is transformed by the time-scale transformation operator Γ_k, which can be implemented via a zero-order hold device followed by a simple data mapping operation.

The cost function is chosen as the form (14). The weighting matrices are set as $Q = qI_{mN}$, $R = rI_{\ell N}$ where $q, r = 1$. The discounted factor is chosen as $\rho = 0.95$, and the desired disturbance attention level is $\gamma = 5$. For $u_0 = \mathbf{0}$, $e_0 = r^d$ and $L^0_w = \mathbf{0}$, the Arimoto-type gain is set to be $L^0_u = 0.05I_{mN}$, which satisfies the ILC convergence condition. The probing noise ζ_k satisfies the multivariate normal distribution with mean vector $\mu = \mathbf{0}$ and covariance matrix $\Sigma = 10^{-6}I_{\ell N}$. The goal of both Arimoto-type gain and the probing noise is to collect sufficient data for the learning of Algorithm 1 in a safe manner. Therefore, slight adjustments will not significantly deteriorate the overall performance. The stop criteria for policy iteration of Algorithm 1 is set to be $\epsilon_u = 10^{-6}$ and $\epsilon_w = 10^{-6}$. For brevity, the iteration-varying disturbance w_k is also given, satisfying a multivariate normal distribution with the same parameters as ζ_k but attenuates with $w_k = \frac{\zeta_k}{k}$ along the iteration axis.

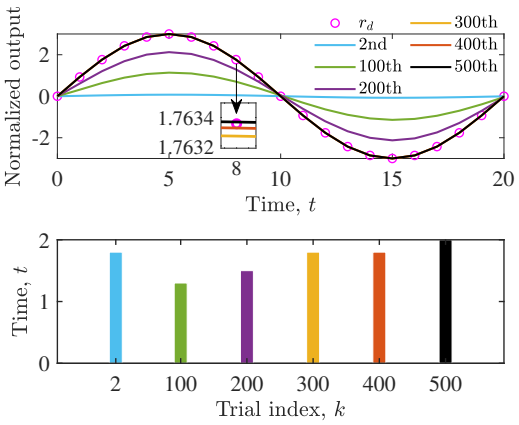

Fig. 3. Normalized tracking outputs of different trials and the corresponding variable lengths of the reference in the time-scale-varying task.

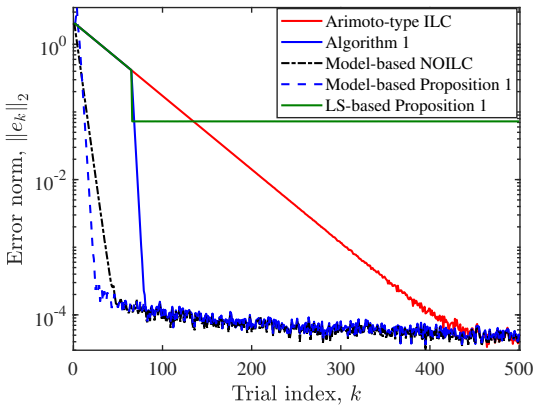

Fig. 4. Tracking error convergence of Algorithm 1 compared to the Arimoto-type ILC, the model-based NOILC, the model-based solution in Proposition 1, and the LS-based Proposition 1. LS-based Proposition 1 uses LS estimation of G from the collected data for the model-based implementation.

The tracking reference signals vary in length from 1s to 2s satisfying a continuous uniform distribution. A small sampling number $N = 20$ of sampled-data ILC is chosen to better show the tracking performance. Algorithm 1 is applied with 500 iterations. The outputs of the time-scale-varying tasks are given in Fig. 3. It is shown that the time-scale-varying tasks are completed effectively by the developed ILC algorithm, despite the corresponding learning and iteration-varying disturbance issues.

The convergence of the tracking error norm is given in Fig. 4. The standard 2-norm is employed for the calculation of the error norm. We also apply Arimoto-type ILC, the model-based NOILC as Theorem 2 in [45], and Proposition 1. The Arimoto-type ILC employs the same learning gain as the initial policy of Algorithm 1, and both NOILC and Proposition 1 use the same weighting parameters for comparisons. Least-squares (LS) estimation paired with Proposition 1 is also employed as an in-

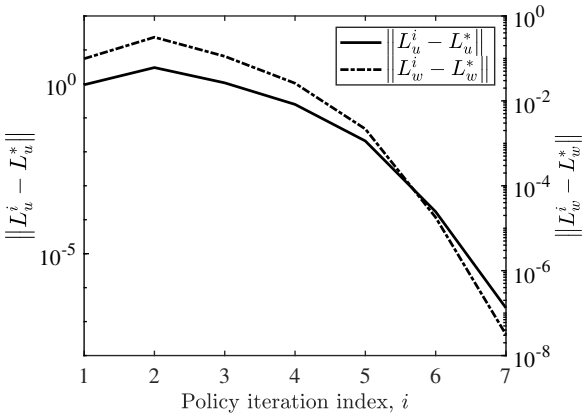

Fig. 5. Norm convergence of the ILC input policy L_u^i and the disturbance policy L_w^i.

direct data-based control method for comparison. Note that NOILC and Proposition 1 employ the model knowledge of the system, which is assumed unknown and directly learned based on the I/O data in our method. It is shown in the comparisons between these two model-based ILC designs that learning from future costs can achieve a faster convergence speed as studied in [4]. In addition, decreasing the discount factor ρ will slow down the trial convergence and make Proposition 1 closer to NOILC. This property also holds in the data-driven Algorithm 1. Once sufficient data is collected, Algorithm 1 converges to the boundary within several iterations by applying the learned policy. LS-based Proposition 1 can only converge to a higher boundary with the same amount of data used in Algorithm 1. This is because LS requires more data to be informative for Proposition 1 as discussed in [26,44].

Further, the convergence of the target policy $L^i = \{L_u^i, L_w^i\}$ is given in Fig. 5. For each policy iteration i, a generalized Sylvester equation (36) is efficiently solved. Both L_u^i and L_w^i converge to their optimum under preset stop criteria within 7 policy iterations. Since Algorithm 1 is off-policy, the behavior policy (Arimoto-type ILC) is used to collect data for Algorithm 1 to learn the robust optimal gain. The target policy will not be applied once sufficient data is collected for robust learning. Further, only L_u^* is applied, and L_w^i is merely used for evaluation and improvement during the policy iteration process, which is consistent with the practical situation.

The iteration-varying disturbance w_k is handled by the two-player zero-sum game design. The attenuation performance is quantified by a ratio between the tracking performance and the finite-energy disturbance. Denote the disturbance attenuation level with respect to k by

$$\gamma_k = \sqrt{\sum_{k=0}^{\infty} \left(e_k^\top Q e_k + \Delta u_k^\top R \Delta u_k\right) / \sum_{k=0}^{\infty} w_k^\top w_k}. \quad (46)$$

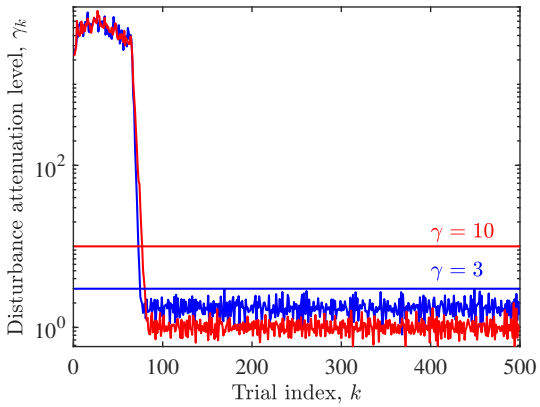

Fig. 6. Iteration-varying disturbance attenuation performance under different preset levels.

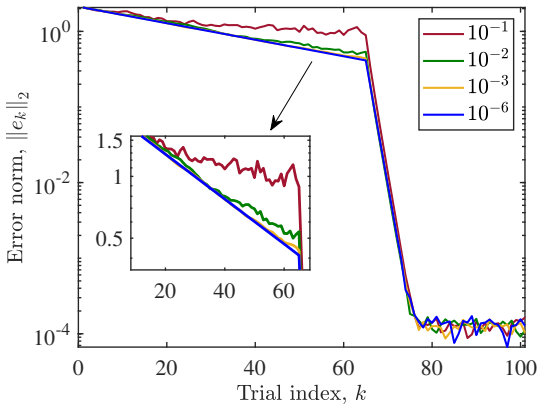

Fig. 7. Tracking error convergence of Algorithm 1 under different levels of probing noise.

Fig. 6 shows the change in disturbance attenuation level along the iteration axis. It is shown that γ_k converges to a low level quickly under Algorithm 1 and the preset disturbance attenuation condition (13) is satisfied. Adjusting γ will change the importance of eliminating the effect of w_{k+1} in the overall objective as designed in the cost function (14). The disturbance attenuation performance could be further improved by increasing γ. A small γ may result in failure of learning since the disturbance attenuation condition could be impossible to satisfy. In addition, Fig. 7 shows that Algorithm 1 is robust to different levels of probing noise.

6 Conclusion and future work

This paper develops a direct data-based ILC method for discrete-time LTI systems to track repetitive reference signals with varying time scales. It is pointed out that the discrete-time ILC for time-scale-varying tasks suffers from issues of corresponding learning. By adopting a time-scale transformation scheme, the challenge is stated as a trial-domain $\mathcal{H}_\infty$ control problem and is solved in a model-free manner from a two-player zero-sum game perspective.

Considering the future iteration performance in the cost function, an efficient direct data-based ILC algorithm is developed, i.e., Algorithm 1. Algorithm 1 is designed to be off-policy to ensure security and efficiency. The Arimoto-type ILC is utilized as the behavior policy to collect data for updating and evaluating the target policy, which converges to the model-based solution of solving a DGARE by value iteration. In addition, a sufficient condition for collecting data is given based on Willems' Fundamental Lemma. The feasibility and effectiveness of the developed ILC design are verified on the simulation model of an F-16 aircraft autopilot.

Future work remains to further explore how many iterations are required exactly for data-based learning. Also, it is theoretically challenging to extend the work to nonlinear systems. Control saturation could also be an interesting issue, which may bring in integral windup in the trial domain in practice.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China under Grant 62361136585, partially by the 111 Project under Grant B23008, partially by the Helicopter Transmission Technology Key Laboratory Open Subjects under Grant HTL-O-22G05, and partially by the National Science Centre in Poland under Grant 2023/48/Q/ST7/00205.

A Proof of Proposition 1

Proof: Let $\frac{\partial V(e_k)}{\partial \Delta u_{k+1}} = \mathbf{0}$ and $\frac{\partial V(e_k)}{\partial w_{k+1}} = \mathbf{0}$ to have

$$\begin{cases} \left(R + G^\top P G\right) \Delta u_{k+1} = G^\top P e_k + G^\top P w_{k+1}, \\ \left(\gamma^2 I_{mN} - P\right) w_{k+1} = P e_k - P G \Delta u_{k+1}, \end{cases} \tag{A.1}$$

which yields (19) and (20), where P satisfies the Bellman equation (18). Substituting (19) and (20) to (18) yields

$$P = \rho P + Q - \rho \left(P G L_u^* - P L_w^*\right). \tag{A.2}$$

For the sake of simplicity, let

$$L_1 = \left[R + G^\top P G + G^\top P \left(\gamma^2 I_{mN} - P\right)^{-1} P G\right]^{-1},$$
$$L_2 = \left[\gamma^2 I_{mN} - P + P G \left(R + G^\top P G\right)^{-1} G^\top P\right]^{-1}.$$

To prove that (A.2) is equivalent to DGARE (22), reformulate (A.2) by (21). Since P is symmetric and positive

definite, $\left(\gamma^2 I_{mN} - P\right)^{-1} P$ is symmetric. It follows that

$$\begin{aligned}
&PGL_u^* - PL_w^* \\
&= PGL_1 \left[G^\top P + G^\top P \left(\gamma^2 I_{mN} - P\right)^{-1} P\right] \\
&\quad - PL_2 \left[P - PG \left(R + G^\top PG\right)^{-1} G^\top P\right] \\
&= PGL_1 G^\top P + PGL_1 G^\top P \left(\gamma^2 I_{mN} - P\right)^{-1} P \\
&\quad + P\left(L_2 L_2^{-1} - \gamma^2 L_2 + L_2 P - L_2 P\right) \\
&= PGL_1 G^\top P + P\left(\gamma^2 I_{mN} - P\right)^{-1} PGL_1 G^\top P \\
&\quad + P - \left(\gamma^2 I_{mN} - P\right) L_2 P - PL_2 P,
\end{aligned} \tag{A.3}$$

where by the Matrix Inversion Lemma, we have

$$\begin{aligned}
&P - \left(\gamma^2 I_{mN} - P\right) L_2 P \\
&= P - \left(\gamma^2 I_{mN} - P\right) \left[\left(\gamma^2 I_{mN} - P\right)^{-1} - \left(\gamma^2 I_{mN} - P\right)^{-1}\right. \\
&\quad \left. \times PGL_1 G^T P \left(\gamma^2 I_{mN} - P\right)^{-1}\right] P \\
&= PGL_1 G^\top P \left(\gamma^2 I_{mN} - P\right)^{-1} P.
\end{aligned} \tag{A.4}$$

It follows from (A.3), (A.4), and the Schur complement of a matrix that

$$\begin{aligned}
&PGL_u^* - PL_w^* \\
&= PGL_1 G^\top P + P\left(\gamma^2 I_{mN} - P\right)^{-1} PGL_1 G^\top P \\
&\quad + PGL_1 G^\top P \left(\gamma^2 I_{mN} - P\right)^{-1} P - PL_2 P \\
&= \begin{bmatrix} -PG & P \end{bmatrix} \times \\
&\begin{bmatrix} L_1 & L_1 G^\top P \left(P - \gamma^2 I_{mN}\right)^{-1} \\ \left(P - \gamma^2 I_{mN}\right)^{-1} PGL_1 & -L_2 \end{bmatrix} \begin{bmatrix} -G^\top P \\ P \end{bmatrix} \\
&= \begin{bmatrix} -PG & P \end{bmatrix} \begin{bmatrix} R + G^\top PG & -G^\top P \\ -PG & P - \gamma^2 I_{mN} \end{bmatrix}^{-1} \begin{bmatrix} -G^\top P \\ P \end{bmatrix}.
\end{aligned}$$

Therefore, P also satisfies the DGARE (22). □

B Proof of Theorem 2

Proof: By (41), the constructed Z in Algorithm 1 is nonsingular and hence (36) has a unique solution $\Phi^{L^{i+1}}$, which is also the unique solution of (33). By the full-row-rank condition (41), there exist $(2m + \ell) N$ linearly independent z_k generated by Algorithm 1, which means $\Phi^{L^{i+1}}$ is also the unique solution of

$$\Phi^{L^{i+1}} = W + \rho \Theta_i^\top \Phi^{L^{i+1}} \Theta_i. \tag{B.1}$$

For the robust optimal policy, the solution is given in (29), which satisfies

$$\Phi^{L^*} = W + \rho \Theta_*^\top \Phi^{L^*} \Theta_*, \tag{B.2}$$

where the symbol $*$ is added to highlight the optimum. Then, the convergence of $\Phi^{L^{i+1}}$ is given by first proving the following identity

$$\Phi^{L^{i+1}} - \Phi^{L^*} = \rho \left[\Theta_i^\top (\Phi^{L^{i+1}} - \Phi^{L^*}) \Theta_i + \Lambda_{i,*}^\top \Phi^{L^*} \Lambda_{i,*}\right], \tag{B.3}$$

where $\Lambda_{i,*} \triangleq \Theta_i - \Theta_*$, and

$$\begin{aligned}
\Lambda_{i,*} &= \begin{bmatrix} I_{mN} & -G & I_{mN} \\ L_u^i & -L_u^i G & L_u^i \\ L_w^i & -L_w^i G & L_w^i \end{bmatrix} - \begin{bmatrix} I_{mN} & -G & I_{mN} \\ L_u^* & -L_u^* G & L_u^* \\ L_w^* & -L_w^* G & L_w^* \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ L_u^i - L_u^* & (L_u^* - L_u^i) G & L_u^i - L_u^* \\ L_w^i - L_w^* & (L_w^* - L_w^i) G & L_w^i - L_w^* \end{bmatrix}.
\end{aligned} \tag{B.4}$$

Note from (B.1) and (B.2) that

$$\begin{aligned}
&\Phi^{L^{i+1}} - \Phi^{L^*} = \rho \left[\Theta_i^\top \Phi^{L^{i+1}} \Theta_i - \Theta_*^\top \Phi^{L^*} \Theta_*\right] \\
&= \rho \left[\Theta_i^\top (\Phi^{L^{i+1}} - \Phi^{L^*}) \Theta_i + \Theta_i^\top \Phi^{L^*} \Theta_i - \Theta_*^\top \Phi^{L^*} \Theta_*\right].
\end{aligned} \tag{B.5}$$

By $\Theta_i = \Theta_* + \Lambda_{i,*}$, we have

$$\begin{aligned}
&\Theta_i^\top \Phi^{L^{i+1}} \Theta_i - \Theta_*^\top \Phi^{L^{i+1}} \Theta_* \\
&\quad = (\Theta_* + \Lambda_{i,*})^\top \Phi^{L^*} (\Theta_* + \Lambda_{i,*}) - \Theta_*^\top \Phi^{L^*} \Theta_* \\
&\quad = \Lambda_{i,*}^\top \Phi^{L^*} \Lambda_{i,*} + \Theta_*^\top \Phi^{L^*} \Lambda_{i,*} + \Lambda_{i,*}^\top \Phi^{L^*} \Theta_*.
\end{aligned} \tag{B.6}$$

To prove (B.3), $\Lambda_{i,*}^\top \Phi^{L^*} \Lambda_{i,*} = \Theta_i^\top \Phi^{L^*} \Theta_i - \Theta_*^\top \Phi^{L^*} \Theta_*$ should be proved, which holds if $\Theta_*^\top \Phi^{L^*} \Lambda_{i,*} = \mathbf{0}$ and $\Lambda_{i,*}^\top \Phi^{L^*} \Theta_* = \mathbf{0}$. We now prove these two identities. By (B.4), we have

$$\begin{aligned}
&\Lambda_{i,*}^\top \Phi^{L^*} \Theta_* = \\
&\begin{bmatrix} \mathbf{0} & (L_u^i - L_u^*)^\top & (L_w^i - L_w^*)^\top \\ \mathbf{0} & G^\top (L_u^* - L_u^i)^\top & G^\top (L_w^* - L_w^i)^\top \\ \mathbf{0} & (L_u^i - L_u^*)^\top & (L_w^i - L_w^*)^\top \end{bmatrix} \begin{bmatrix} \phi_{e*} & -\phi_{e*} G & \phi_{e*} \\ \phi_{u*} & -\phi_{u*} G & \phi_{u*} \\ \phi_{w*} & -\phi_{w*} G & \phi_{w*} \end{bmatrix},
\end{aligned} \tag{B.7}$$

where $\phi_{e*} = \phi_{ee}^* + \phi_{eu}^* L_u^* + \phi_{ew}^* L_w^*$, $\phi_{u*} = \phi_{ue}^* + \phi_{uu}^* L_u^* + \phi_{uw}^* L_w^*$, and $\phi_{w*} = \phi_{we}^* + \phi_{wu}^* L_u^* + \phi_{ww}^* L_w^*$. It is sufficient to prove that $\phi_{u*} = \mathbf{0}$ and $\phi_{w*} = \mathbf{0}$. By the Matrix

Inversion Lemma, we have

$$\begin{aligned}
&\phi_{u*} = \phi^*_{ue} + \\
&\phi^*_{uu}[\phi^*_{uu} - \phi^*_{uw}(\phi^*_{ww})^{-1}\phi^*_{wu}]^{-1}[\phi^*_{uw}(\phi^*_{ww})^{-1}\phi^*_{we} - \phi^*_{ue}] \\
&+ \phi^*_{uw}[\phi^*_{ww} - \phi^*_{wu}(\phi^*_{uu})^{-1}\phi^*_{uw}]^{-1}[\phi^*_{wu}(\phi^*_{uu})^{-1}\phi^*_{ue} - \phi^*_{we}] \\
&= \phi^*_{ue} + \phi^*_{uu}[(\phi^*_{uu})^{-1} - (\phi^*_{uu})^{-1}\phi^*_{uw}[-\phi^*_{ww} + \\
&\phi^*_{wu}(\phi^*_{uu})^{-1}\phi^*_{uw}]^{-1}\phi^*_{wu}(\phi^*_{uu})^{-1}][\phi^*_{uw}(\phi^*_{ww})^{-1}\phi^*_{we} - \phi^*_{ue}] \\
&+ \phi^*_{uw}[\phi^*_{ww} - \phi^*_{wu}(\phi^*_{uu})^{-1}\phi^*_{uw}]^{-1}[\phi^*_{wu}(\phi^*_{uu})^{-1}\phi^*_{ue} - \phi^*_{we}] \\
&= \phi^*_{uw}\left[(\phi^*_{ww})^{-1} + [\phi^*_{ww} - \phi^*_{wu}(\phi^*_{uu})^{-1}\phi^*_{uw}]^{-1}\phi^*_{wu}\times\right. \\
&\left.(\phi^*_{uu})^{-1}\phi^*_{uw}(\phi^*_{ww})^{-1} - [\phi^*_{ww} - \phi^*_{wu}(\phi^*_{uu})^{-1}\phi^*_{uw}]^{-1}\right]\phi^*_{we} \\
&= \phi^*_{uw}\left[[\phi^*_{ww} - \phi^*_{wu}(\phi^*_{uu})^{-1}\phi^*_{uw}]^{-1}\phi^*_{wu}(\phi^*_{uu})^{-1} - (\phi^*_{ww})^{-1}\right. \\
&\left.\phi^*_{wu}[\phi^*_{uu} - \phi^*_{uw}(\phi^*_{ww})^{-1}\phi^*_{wu}]^{-1}\right]\phi^*_{uw}(\phi^*_{ww})^{-1}\phi^*_{we} = \mathbf{0}
\end{aligned} \tag{B.8}$$

since it is straightforward that

$$\begin{aligned}
&[\phi^*_{ww} - \phi^*_{wu}(\phi^*_{uu})^{-1}\phi^*_{uw}]^{-1}\phi^*_{wu}(\phi^*_{uu})^{-1} \\
&= (\phi^*_{ww})^{-1}\phi^*_{wu}[\phi^*_{uu} - \phi^*_{uw}(\phi^*_{ww})^{-1}\phi^*_{wu}]^{-1}.
\end{aligned} \tag{B.9}$$

Similarly, it is trivial to prove $\phi_{w*} = \mathbf{0}$. Therefore, we have $\Lambda^\top_{i,*}\Phi^{L^*}\Theta_* = \mathbf{0}$ and similarly $\Theta^\top_*\Phi^{L^*}\Lambda_{i,*} = \mathbf{0}$, which completes the proof of (B.3).

Further, using (B.3), it follows that

$$\begin{aligned}
&\Phi^{L^{i+1}} - \Phi^{L^*} \\
&= \left[\Phi^{L^{i+1}} - \Phi^{L^*} - \rho\Theta_i^\top(\Phi^{L^{i+1}} - \Phi^{L^*})\Theta_i\right] + \\
&\left[\rho\Theta_i^\top(\Phi^{L^{i+1}} - \Phi^{L^*})\Theta_i - \rho(\Theta_i^2)^\top(\Phi^{L^{i+1}} - \Phi^{L^*})\Theta_i^2\right] + \cdots \\
&= \rho(\Theta_i^0)^\top\Lambda^\top_{i,*}\Phi^{L^*}\Lambda_{i,*}\Theta_i^0 + \rho(\Theta_i^1)^\top\Lambda^\top_{i,*}\Phi^{L^*}\Lambda_{i,*}\Theta_i^1 + \cdots \\
&= \rho\sum_{j=0}^{\infty}(\Theta_i^j)^\top\Lambda^\top_{i,*}\Phi^{L^*}\Lambda_{i,*}\Theta_i^j,
\end{aligned} \tag{B.10}$$

where the superscript j in Θ_i^j represents the jth power of Θ_i. The discrete-time Lyapunov function (B.1) has a unique solution $\Phi^{L^{i+1}} \succeq \mathbf{0}$ subject to $W \succeq \mathbf{0}$ and a stabilizing matrix Θ_i. Therefore, we have $\Phi^{L^{i+1}} - \Phi^{L^*} \succeq \mathbf{0}$ by (B.10). Similarly, we can prove $\Phi^{L^i} \succeq \Phi^{L^{i+1}}$. Therefore, $\Phi^{L^i} \succeq \Phi^{L^{i+1}} \succeq \Phi^{L^*}$.

Finally, it sufficient to show that $\lim_{i\to\infty}\Phi^{L^{i+1}} = \Phi^{L^*}$ to complete the proof since L_u^{i+1} and L_w^{i+1} are directly derived from the components of $\Phi^{L^{i+1}}$. This is equivalent to proving that Φ^{L^*} is the unique fixed point of Algorithm 1. Set $\left\{L_u^i, L_w^i\right\} = \{L_u^*, L_w^*\}$ in Algorithm 1 yielding

$$Z^\top\Phi^{L^{i+1}}Z = Z^\top WZ + \rho Z^\top\Theta_*^\top\Phi^{L^{i+1}}\Theta_* Z, \tag{B.11}$$

which has a unique solution for the nonsingular constructed matrix Z. Then, we have $\Phi^{L^*} \succeq \Phi^{L^{i+1}} \succeq \Phi^{L^*}$, which implies Φ^{L^*} is a fixed point of Algorithm 1. Further, by Proposition 1, $\{L_u^*, L_w^*\}$, derived from Φ^{L^*}, is the unique analytic solution pair of the Bellman equation (18) with respect to the cost function (14). Therefore, Φ^{L^*} is the unique fixed point of Algorithm 1. □

References

[1] Hyo-Sung Ahn. Reinforcement learning and iterative learning control: Similarity and difference. In *Proceedings of the International Conference on Mechatronics and Information Technology, Gwangju, Korea*, pages 3–5, 2009.

[2] Hyo-Sung Ahn, YangQuan Chen, and Kevin L Moore. Iterative learning control: Brief survey and categorization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1099–1121, 2007.

[3] Asma Al-Tamimi, Frank L Lewis, and Murad Abu-Khalaf. Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica*, 43(3):473–481, 2007.

[4] Notker Amann, David H Owens, and Eric Rogers. Predictive optimal iterative learning control. *International Journal of Control*, 69(2):203–226, 1998.

[5] Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.

[6] Kira L Barton and Andrew G Alleyne. A cross-coupled iterative learning control design for precision motion control. *IEEE Transactions on Control Systems Technology*, 16(6):1218–1231, 2008.

[7] Joost Bolder, Stephan Kleinendorst, and Tom Oomen. Data-driven multivariable ILC: enhanced performance by eliminating L and Q filters. *International Journal of Robust and Nonlinear Control*, 28(12):3728–3751, 2018.

[8] Joost Bolder and Tom Oomen. Rational basis functions in iterative learning control-with experimental verification on a motion system. *IEEE Transactions on Control Systems Technology*, 23(2):722–729, 2014.

[9] Chems Eddine Boudjedir, Mohamed Bouri, and Djamel Boukhetala. Model-free iterative learning control with nonrepetitive trajectories for second-order MIMO nonlinear systems-Application to a delta robot. *IEEE Transactions on Industrial Electronics*, 68(8):7433–7443, 2020.

[10] Douglas A Bristow, Marina Tharayil, and Andrew G Alleyne. A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96–114, 2006.

[11] Xuhui Bu, Qiongxia Yu, Zhongsheng Hou, and Wei Qian. Model free adaptive iterative learning consensus tracking control for a class of nonlinear multiagent systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(4):677–686, 2017.

[12] Cheng-Wei Chen and Tsu-Chin Tsao. Data-driven progressive and iterative learning control. *IFAC-PapersOnLine*, 50(1):4825–4830, 2017.

[13] Tongwen Chen and Bruce A. Francis. *Optimal Sampled-Data Control Systems*. Springer, 1995.

[14] Ronghu Chi, Zhongsheng Hou, Biao Huang, and Shangtai Jin. A unified data-driven design framework of optimality-based generalized iterative learning control. *Computers & Chemical Engineering*, 77:10–23, 2015.

[15] Mitchell Cobb, James Reed, Maxwell Wu, Kirti D Mishra, Kira Barton, and Chris Vermillion. Flexible-time receding horizon iterative learning control with application to marine hydrokinetic energy systems. *IEEE Transactions on Control Systems Technology*, 30(6):2767–2774, 2022.

[16] Claudio De Persis and Pietro Tesi. Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924, 2019.

[17] Feng Ding, Peter X Liu, and Jie Ding. Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle. *Applied Mathematics and Computation*, 197(1):41–50, 2008.

[18] Rogier Dinkla, Tom Oomen, Sebastiaan Mulders, and Jan-Willem van Wingerden. Data-enabled predictive repetitive control. *arXiv preprint arXiv:2408.15210*, 2024.

[19] Jianfei Dong. Robust data-driven iterative learning control for linear-time-invariant and Hammerstein–Wiener systems. *IEEE Transactions on Cybernetics*, 53(2):1144–1157, 2021.

[20] Florian Dörfler. Data-driven control: Part two of two: Hot take: Why not go with models? *IEEE Control Systems Magazine*, 43(6):27–31, 2023.

[21] Shoulin Hao, Tao Liu, and Wojciech Paszke. PIO based data-driven iterative learning control for nonlinear batch processes with nonrepetitive disturbances subject to input constraints. *IFAC-PapersOnLine*, 54(3):25–30, 2021.

[22] Yu Hui, Ronghu Chi, Biao Huang, and Zhongsheng Hou. Extended state observer-based data-driven iterative learning control for permanent magnet linear motor with initial shifts and disturbances. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(3):1881–1891, 2019.

[23] Pieter Janssens, Goele Pipeleers, and Jan Swevers. A data-driven constrained norm-optimal iterative learning control framework for LTI systems. *IEEE Transactions on Control Systems Technology*, 21(2):546–551, 2012.

[24] Gu-Min Jeong and Chong-Ho Choi. Iterative learning control for linear discrete time nonminimum phase systems. *Automatica*, 38(2):287–291, 2002.

[25] Sadao Kawamura and Norimitsu Sakagami. Analysis on dynamics of underwater robot manipulators based on iterative learning control and time-scale transformation. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1088–1094. IEEE, 2002.

[26] Bahare Kiumarsi, Frank L Lewis, and Zhong-Ping Jiang. H_∞ control of linear discrete-time systems: Off-policy reinforcement learning. *Automatica*, 78:144–152, 2017.

[27] Jinze Li, Senping Tian, Yunjian Peng, and Panpan Gu. Data-driven-based predictive optimal for a class of iterative learning control by Q-learning method. In *2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS)*, pages 1214–1220. IEEE, 2023.

[28] Xuefang Li, Dong Shen, and Jian-Xin Xu. Adaptive iterative learning control for MIMO nonlinear systems performing iteration-varying tasks. *Journal of the Franklin Institute*, 356(16):9206–9231, 2019.

[29] Jianan Liu, Zike Zhou, Wenjing Hong, and Jia Shi. Two-dimensional iterative learning control with deep reinforcement learning compensation for the non-repetitive uncertain batch processes. *Journal of Process Control*, 131:103106, 2023.

[30] L. Ljung. *System Identification: Theory for the User,* 2nd Edition. Prentice-Hall, 1999.

[31] Richard W Longman. Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, 73(10):930–954, 2000.

[32] Victor G Lopez, Mohammad Alsalti, and Matthias A Müller. Efficient off-policy Q-learning for data-based discrete-time LQR problems. *IEEE Transactions on Automatic Control*, 68(5):2922–2933, 2023.

[33] Biao Luo, Yin Yang, and Derong Liu. Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems. *IEEE Transactions on Cybernetics*, 51(7):3630–3640, 2020.

[34] Michael Meindl, Dustin Lehmann, and Thomas Seel. Bridging reinforcement learning and iterative learning control: Autonomous motion learning for unknown, nonlinear dynamics. *Frontiers in Robotics and AI*, 9:793512, 2022.

[35] Deyuan Meng. Control analysis and synthesis of data-driven learning for uncertain linear systems. *Automatica*, 148:110734, 2023.

[36] Kevin L Moore, Mohua Ghosh, and Yang Quan Chen. Spatial-based iterative learning control for motion control applications. *Meccanica*, 42:167–175, 2007.

[37] Tom Oomen, Jeroen van de Wijdeven, and Okko Bosgra. Suppressing intersample behavior in iterative learning control. *Automatica*, 45(4):981–988, 2009.

[38] Maurice Poot, Jim Portegies, and Tom Oomen. On the role of models in learning control: Actor-critic iterative learning control. *IFAC-PapersOnLine*, 53(2):1450–1455, 2020.

[39] Patrick M Sammons, David Hoelzle, and Kira Barton. Time-scale transformed iterative learning control for a class of nonlinear systems with uncertain trial duration. *IEEE Transactions on Control Systems Technology*, 28(5):1972–1979, 2019.

[40] Dong Shen and Xuefang Li. A survey on iterative learning control with randomly varying trial lengths: Model, synthesis, and convergence analysis. *Annual Reviews in Control*, 48:89–102, 2019.

[41] Bing Song. *From model-based to data-driven discrete-time iterative learning control.* PhD thesis, Columbia University, 2019.

[42] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction.* MIT Press, 2018.

[43] Jeroen van de Wijdeven and Okko H Bosgra. Using basis functions in iterative learning control: Analysis and design theory. *International Journal of Control*, 83(4):661–675, 2010.

[44] Henk J Van Waarde, Jaap Eising, Harry L Trentelman, and M Kanat Camlibel. Data informativity: A new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, 65(11):4753–4768, 2020.

[45] Jurgen Van Zundert, Joost Bolder, and Tom Oomen. Optimality and flexibility in iterative learning control for varying tasks. *Automatica*, 67:295–302, 2016.

[46] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards.* PhD thesis, King's College Cambridge, United Kingdom, 1989.

[47] Jan C Willems, Paolo Rapisarda, Ivan Markovsky, and Bart LM De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.

[48] Yuxin Wu and Deyuan Meng. Data-based trackability criteria and control design for disturbed learning systems. *Automatica*, 155:111113, 2023.

[49] Jian-Xin Xu. Direct learning of control efforts for trajectories with different time scales. *IEEE Transactions on Automatic Control*, 43(7):1027–1030, 1998.

[50] Libin Xu, Weimin Zhong, Jingyi Lu, Furong Gao, Feng Qian, and Zhixing Cao. Learning of iterative learning control for flexible manufacturing of batch processes. *ACS Omega*, 7(23):19939–19947, 2022.

[51] Yueqing Zhang, Bing Chu, and Zhan Shu. Model-free predictive optimal iterative learning control using reinforcement learning. In *2022 American Control Conference (ACC)*, pages 3279–3284. IEEE, 2022.