# Predictive Optimal Iterative Learning Control for Nonlinear Systems using the Koopman Operator

Xinyue Tao, Hongfeng Tao[*], Luyuan Gao [*], Wojciech Paszke[†], Eric Rogers[‡]

15.07.2024

**Abstract**

This paper develops a predictive optimal iterative learning control design for nonlinear systems based on the Koopman operator. Iterative learning control applies to systems that make repeated executions, known as trials, over a finite duration, termed the trial length. Once a trial is complete, all information generated is available to update the control signal for the subsequent trial. The first step in design is to approximately model the nonlinear system as a high-dimensional linear model using the Koopman operator and extended dynamic mode decomposition, which is applied on each trial. Then, an iterative learning control law is designed with predictive action over an infinite duration in the trial-to-trial direction. The robust convergence of the tracking error is analyzed, and a numerical case study highlights the design's effectiveness.

Keywords: iterative learning control, nonlinear systems, Koopman operator, predictive action

[*] Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, China, CONTACT Hongfeng Tao. Email: taohongfeng@jiangnan.edu.cn

[†] Institute of Automation, Electronic and Electrical Engineering, University of Zielona Góra, Zielona Góra, Poland

[‡] School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, United Kingdom

# 1 Introduction

Iterative learning control (ILC) is applicable to systems that repeat the same finite-duration task over and over again. Each completion is termed a trial, and the duration is the trial's length. Once a trial is complete, the system resets, and all information generated during a trial is available for use in updating the control input for the subsequent trial. Hence, ILC updates the input, a signal, whereas in adaptive control, the controller, a system, is updated. In fact, ILC can be applied to examples where a finite-duration operation is completed, and then there is a stoppage or time gap before the operation repeats. Hence ILC can be applied to batch processes.

An application area for ILC is pick and place robots, i.e., the robot collects an object from a specified location, transfers it over a finite duration, places it on a conveyor under synchronization, returns to the specified location and repeats this sequence over and over again. The first work, widely credited to Arimoto et al. [3], was in this particular application area. Since then the number of application areas has continued to expand and there has been applications in healthcare, e.g., robotic assisted stroke rehabilitation. Possible sources for the early literature include the survey papers [4, 1]. The research text [16] gives a comprehensive coverage ILC design methods and application in both engineering and healthcare, with references to the original work.

The use of predictive action in ILC designs can occur in either the trial to trial direction or along the trial. In the former case, the integration of predictive control into the ILC enables the utilisation of a greater quantity of future input and output information in the control design, which may result in enhanced control capabilities (see, e.g. Reference [2, 19]). The use of predictive action along a trial has been addressed in Reference [22]. In these cases the dynamics are assumed to be linear. For predictive action in the trial to trial direction, the predictive ILC problem can be reformulated into a structure similar to the solution of the Riccati equation in the linear quadratic regulator (LQR) framework [11]. Reinforcement learning algorithms have found increasing use in solving optimal control problems. For example, the research work [21] proposes an improved general value iteration algorithm for discrete-time zero-sum games. Meanwhile, in reference [7], a discounted value iteration algorithm with accelerated learning and an adaptive acceleration interval is proposed.

Previous research on predictive ILC for nonlinear systems includes Reference [23], where a piecewise linearization method was used to obtain a piecewise affine approximate model of a nonlinear system. In Reference [25], a dynamic linearization approach is used to design a predictive ILC law. The piecewise linearization method applies the Taylor series expansion in segments but requires considerable information about the system dynamics model. The iterative dynamic linearization approach uses partial differential derivative techniques. Still, several adjustable parameters are involved in ILC design, which may be problematic in some applications, especially for tuning to achieve performance.

The Koopman operator is one technique for approximating nonlinear systems by a linear approximate model using observable functions [9, 10]. However, the Koopman operator is infinite-dimensional, so approximation by a finite-dimensional representation is required, particularly for applications. The most common method used for this purpose is the data-driven method dynamic mode decomposition(DMD) [18]. Also, the extended dynamic mode decomposition (EDMD) modifies the DMD for use with controlled system dynamics [24]. The data-driven Koopman operator can be applied to obtain a linear representation of a nonlinear system with unknown dynamics for control systems purposes (see, e.g., Reference [12]).

This paper develops a predictive optimal ILC (POILC) law for nonlinear systems, where the prediction is in the trial-to-trial direction, using the Koopman operator. In this paper, the theory of the Koopman operator and an extended dynamic mode decomposition (EDMD) is used to obtain a data-driven linear model of a nonlinear system. It is established that the design for nonlinear systems can be converted to POILC design for linear systems with non-repeating model uncertainty. Properties of the design are also established including robustness. Finally, a numerical case study is given to demonstrate the application of the new design, including a comparison with three alternative ILC laws. Other research on the use of this operator in ILC design includes Reference [17].

The rest of this paper is organized as follows. Section II describes the control problem and introduces the basics of the Koopman operator and the EDMD method. Section III develops the POILC design, Section IV gives a robust convergence analysis of the new design, and Section V is the numerical case study. Finally, Section VI summarises the novel contributions in this paper.

# 2 Background and problem formulation

In this section, we present the modeling framework and formulate the control problem. Section 2.1 introduces the Koopman operator theory and its data-driven approximation using the EDMD method, which provides a lifted linear representation of the original nonlinear system. Section 2.2 then formulates

ILC problem, defining the control objective in the ILC framework. The detailed controller design will be given in Section 3.

## 2.1 Koopman operator

Consider the uncontrolled system

$$x(t+1) = f(x(t)), \tag{1}$$

where $x \in \mathbb{R}^n$, (for notational simplicity, the dependence of variables on $t$ is suppressed except where absolutely needed in the rest of this paper). Then (see, e.g.,Reference [9]), the Koopman operator $\mathcal{K} : \mathcal{F} \rightarrow \mathcal{F}$ is defined as

$$\left(\mathcal{K}\psi\right)(x) = \psi\left(f(x)\right), \tag{2}$$

for all $\psi\left(x\right) : \mathbb{R}^n \rightarrow \mathbb{R}$ belonging to $\mathcal{F}$, which is a space of functions invariant under the Koopman operator and also termed observables. Moreover, this operator is linear, and typically infinite-dimensional even in the case when the dynamics are nonlinear, and fully captures all properties of the underling system provided that all entries in the state vector lie in $\mathcal{F}$. This operator completely captures all properties of the underlying system under the condition provided $\mathcal{F}$ contains all entries in the state vector.

Consider the controlled system

$$x(t+1) = f(x(t), u(t)). \tag{3}$$

Then a number of methods exist to extend the Koopman operator to this case. One method is to define the extended state vector as $\bar{x}(t) = \left[\begin{array}{cc} x^T(t) & u^T(t) \end{array}\right]^T$, and then the dynamics of (3) is described by

$$\hat{x} = F(\bar{x}) := \left[\begin{array}{c} f(x(t), u(0)) \\ \mathcal{S}u(t) \end{array}\right], \tag{4}$$

where $\mathcal{S}$ is the left shift operator, i.e., if $u(i)$ denotes the $i$ th entry in $u$, $\left(\mathcal{S}u\right)(i) = u\left(i+1\right)$. In this setting, the Koopman operator is also defined.

For applications, including control law design, the requirement is a finite dimensional approximation of the Koopman operator. One approach to this problem is the extended dynamic mode decomposition algorithm (EDMD) [9, 10, 24]. This method, given a collection of data points $\{\bar{x}_j, \hat{x}_j\}$, $j = 1, \ldots, L$, satisfying (4), aims to find a matrix $\mathcal{G}$, the transpose of the finite-dimensional approximation of $\mathcal{K}$ that minimizes

$$\sum_{j=1}^{L} ||\phi(\hat{x}_j) - \mathcal{G}\phi(\bar{x}_j)||^2$$

where

$$\phi(\bar{x}) = \begin{bmatrix} \phi_1(\bar{x}) & \dots & \phi_{N_\phi}(\bar{x}) \end{bmatrix}^T \tag{5}$$

is a vector of observables (lifting functions), which is used to embed the nonlinear system into a higher-dimensional space where its evolution can be approximated linearly. Also, this problem cannot be solved in finite time unless the entries in (3) are selected appropriately.

Consider the problem of approximating the nonlinear dynamics of the system (3), a vector of observables $\phi(\bar{x})$ is designed as

$$\phi(\bar{x}) = \begin{bmatrix} \psi(x) \\ u(0) \end{bmatrix}, \tag{6}$$

where $\psi(x) = [\psi_1(x), \dots, \psi_{N_\psi}(x)]^\top$, and $N_\psi$ is the the number of observables related to the system state. To ensure identifiability and numerical stability, the chosen basis functions should satisfy linear independence. In practice, $\psi(x)$ is constructed as a vector of basis functions, such as polynomial functions, Gaussian basis functions, or thin-plate spline functions. For a state vector $x = [x_1, ..., x_n]^\top$, a vector of polynomial basis function typically takes the form:

$$\psi(x) = [1, x_1, x_2, \dots, x_n, x_1^2, x_1 x_2, \dots, x_n^2, \dots], \tag{7}$$

where monomials up to a selected degree are included.

A Gaussian basis function is commonly defined as:

$$\psi_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma^2}\right), \tag{8}$$

where $c_i$ is the center and $\sigma$ is the bandwidth of $i$th basis.

A thin-plate spline function is typically given by:

$$\psi_i(x) = \|x - c_i\|^2 \log \|x - c_i\|, \tag{9}$$

where $c_i$ is the center of the $i$th basis.

Suppose that the system is to be approximated by the linear finite-dimensional model

$$z(t+1) = Az(t) + Bu(t), \tag{10}$$

where $z = \psi(x)$. Suppose that the following data sets have been obtained from the nonlinear system(3)

$$\begin{aligned} X_1 &= \begin{bmatrix} x(1) & \dots & x(L) \end{bmatrix}^T, \\ X_2 &= \begin{bmatrix} \hat{x}(1) & \dots & \hat{x}(L) \end{bmatrix}^T, \\ U &= \begin{bmatrix} u(1) & \dots & u(L) \end{bmatrix}^T, \end{aligned} \tag{11}$$

where $\hat{x}(j) = f(x(j), u(j)), j = 1, \ldots, L$. Then the matrices $A$ and $B$ can be obtained as the solution of the optimization problem

$$\min_{A,B} \left\| \mathbf{X_2} - A\mathbf{X_1} - B\mathbf{U} \right\|_F , \tag{12}$$

where $F$ denotes the Frobenius norm and

$$\mathbf{X_1} = \left[ \begin{array}{ccc} \psi(x(1)) & \ldots & \psi(x(L)) \end{array} \right], \tag{13}$$

also,

$$\mathbf{X_2} = \left[ \begin{array}{ccc} \psi(\hat{x}(1)) & \ldots & \psi(\hat{x}(L)) \end{array} \right], \tag{14}$$

The solution to this least square problem is

$$[A, B] = \mathbf{X_2} \left[ \mathbf{X_1}, \mathbf{U} \right]^{\dagger}. \tag{15}$$

where † denotes the Moore–Penrose pseudo-inverse of a matrix.

The prediction of $x$ can be written as

$$\tilde{x}(t) = Cz(t), \tag{16}$$

The matrix $C$ is the solution of

$$\min_{C} \left\| X - C\mathbf{X_1} \right\|_F , \tag{17}$$

i.e.,

$$C = X\mathbf{X_1}^{\dagger}. \tag{18}$$

The obtained Koopman-based model provides a linear representation of the system dynamics, which will be utilized in the controller design presented in Section 3.

## 2.2 ILC problem

Before designing the controller, we now formulate the control problem and specify the tracking objective in the ILC setting. This paper considers systems described by the following state space model in the ILC setting

$$\left\{ \begin{array}{rl} x_k(t+1) = & f(x_k(t), u_k(t)) \\ y_k(t) = & C_p x_k(t) \end{array} \right. ,t \in [0, N], k \in \mathbb{Z}_+, \tag{19}$$

where $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}^l$ and $y_k(t) \in \mathbb{R}^m$ denote the system state, input and output, respectively. The integer $N$ denotes the number of samples along the trial, and if the dynamics have been obtained by sampling a continuous system then $N$ times the sampling period. Moreover, $k \in \mathbb{Z}_+$ is the trial number. $f$ is state transition function. $C_p$ is constant matrices with compatible dimensions.

Given that the trial length is finite, the sample values of a variable along a trial can be assembled into a finite dimensional column vector, termed supervectors in the literature(see e.g., Reference [4]). For the system considered, the lifted input, output, and, supplied desired output, or reference vector, are

$$u_k = \left[\begin{array}{cccc} u_k^{\mathrm{T}}(0) & u_k^{\mathrm{T}}(1) & \cdots & u_k^{\mathrm{T}}(N-1) \end{array}\right]^{\mathrm{T}}, \tag{20}$$

$$y_k = \left[\begin{array}{cccc} y_k^{\mathrm{T}}(1) & y_k^{\mathrm{T}}(2) & \cdots & y_k^{\mathrm{T}}(N) \end{array}\right]^{\mathrm{T}}, \tag{21}$$

and

$$r = \left[\begin{array}{cccc} y_r^{\mathrm{T}}(1) & y_r^{\mathrm{T}}(2) & \cdots & y_r^{\mathrm{T}}(N) \end{array}\right]^{\mathrm{T}}, \tag{22}$$

Moreover, the lifted tracking error on trial $k$ is

$$e_k = r - y_k. \tag{23}$$

This description is extensively used in ILC analysis and design, where for linear dynamics, many designs have been followed through to experimental validation and implementation (see, e.g., Reference [16]), which, in turn, cites the original work.

In ILC, previous trial information is used to update the control signal for the subsequent trial, where once a trial is complete, all information generated on a trial is available for use in updating the control signal. In this paper, the structure of the control law has the form

$$u_{k+1} = u_k + Le_k, \tag{24}$$

where $L$ is termed the learning gain and acts on the previous trial error. Predictive action in an ILC law can be applied from trial-to-trial ($k$) or along the trial ($t$). One form of predictive optimal iterative learning control (POILC) is based on minimizing the quadratic cost function [11]

$$J(u_{k+1}) = \sum_{i=1}^{\infty} \gamma^{i-1}(\|e_{k+i-1}\|_Q^2 + \gamma\|\Delta u_{k+i}\|_R^2), \tag{25}$$

where $\Delta u_{k+i} = u_{k+i} - u_{k+i-1}$, $\gamma > 0$ is a weighting parameter, and the prediction horizon is infinite. The norms are

$$\|e_{k+i-1}\|_Q = \sqrt{e_{k+i-1}^{\mathrm{T}} Q e_{k+i-1}}, \|\Delta u_{k+i}\|_R = \sqrt{\Delta u_{k+i}^{\mathrm{T}} R \Delta u_{k+i}},$$

where $Q$ and $R$ are symmetric positive definite matrices.

This cost function includes the error of all possible future trials and the corresponding change in input on two successive trials. The weighting parameter $\gamma > 0$ determines the importance of more distant (future) errors and incremental inputs compared to the present ones. By including more future signals in the performance criterion, the ILC law becomes less short-sighted. Penalizing the difference between the control inputs on two successive trials aims to prevent a significant jump in the control signal from one trial to the next. If this issue is still a problem, then one option is designing with a constraint on this term, i.e., a constrained ILC design(see, e.g., Reference [6]).

# 3 ILC Design

Based on the data-driven model identification method and ILC control objective presented in the previous section, this section develops an ILC controller aimed at improving tracking performance for the nonlinear system.

The approach used is to apply the Koopman operator theory on each trial, and then ILC design is based on the resulting linear approximation. Consider the observable function $\psi(x)$, and define $z_k(t)$ as

$$z_k(t) = \psi(x_k(t)). \tag{26}$$

Then for trial $k$, a linear approximation ${\mathcal{K}_{N_\phi}}^k$ is computed using the EDMD method [24] of the previous section. Hence the approximate linear system has the form

$$\left\{ \begin{aligned} z_k(t+1) &= A_k z_k(t) + B_k u_k(t) \\ \tilde{y}_k(t) &= C_k z_k(t) \end{aligned} \right. ,t \in [0, N], \ \ k \in \mathbb{Z}_+, \tag{27}$$

where $z \in \mathbb{R}^{N_\psi}$, $\tilde{y} \in \mathbb{R}^m$ is the prediction of output $y$, $A_k \in \mathbb{R}^{N_\psi \times N_\psi}$ is the system matrix, $B_k \in \mathbb{R}^{N_\psi \times l}$ is the input matrix and $C_k \in \mathbb{R}^{m \times N_\psi}$ is the output matrix. $A_k$ and $B_k$ can be computed by solving (15). $C_k$ can be obtained by multiplying the output matrix $C_p$ with the solution of (18).

Define $\tilde{y}_k = [\tilde{y}_k^{\mathrm{T}}(1) \quad \tilde{y}_k^{\mathrm{T}}(2) \quad \ldots \quad \tilde{y}_k^{\mathrm{T}}(N)]^{\mathrm{T}}$, the lifted model description of the dynamics of is, using (20) and (21),

$$\tilde{y}_k = G_k u_k + d_k, \tag{28}$$

where

$$G_k = \begin{bmatrix} C_k B_k & 0 & \cdots & 0 \\ C_k A_k B_k & C_k B_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ C_k {A_k}^{N-1} B_k & \cdots & \cdots & C_k B_k \end{bmatrix}, \tag{29}$$

and

$$d_k = \left[ \begin{array}{cccc} (C_k A_k)^T & (C_k A_k^2)^T & \cdots & (C_k A_k^N)^T \end{array} \right]^{\mathrm{T}} z_k(0). \tag{30}$$

Although the data-driven approach provides an approximate linear model of the original nonlinear system, this model is derived through a finite-dimensional approximation of the infinite-dimensional Koopman operator. As a result, modeling errors are inevitably introduced, and these errors can be viewed as a form of model uncertainty. Moreover, this identification process is repeated across different trials based on newly collected data, hence the identified linear model vary with each trial, the induced modeling error inherently vary with trial. To accurately capture this variability, the modeling error is formulated as a non-repeating model uncertainty, resulting in the lifted model dynamics for design of the form

$$y_k = (G_k + \Delta G_k)u_k + (d_k + \Delta d_k), \tag{31}$$

where $\Delta G_k$ and $\Delta d_k$ represent the modeling error and are assumed to satisfy $||\Delta G_k|| \leq \beta_{\Delta G_k}$ and $||\Delta d_k|| \leq \beta_{\Delta d_k}$ where $|| \cdot ||_2$ denotes the 2-norm, and $\beta_{\Delta G_k} > 0$, $\beta_{\Delta d_k} > 0$ are real scalars.

The output errors on future trials are predicted by $G_k$ as

$$e_{k+i} = e_{k+i-1} - G_k \Delta u_{k+i}, \tag{32}$$

$$\Delta u_{k+i} = L_k e_{k+i-1}, \tag{33}$$

and the cost function for design as

$$J(u_{k+1}) = e_k^{\mathrm{T}} P_k e_k = V(e_k), \tag{34}$$

where $P_k$ is a symmetric, positive definite matrix. Moreover,

$$e_k^{\mathrm{T}} P_k e_k = \|e_k\|_Q^2 + \gamma \|\Delta u_{k+1}\|_R^2 + \gamma e_{k+1}^{\mathrm{T}} P_k e_{k+1}. \tag{35}$$

The associated Hamiltonian is

$$H(e_k, \Delta u_{k+1}) = \|e_k\|_Q^2 + \gamma \|\Delta u_{k+1}\|_R^2 + \gamma e_{k+1}^{\mathrm{T}} P_k e_{k+1} - e_k^{\mathrm{T}} P_k e_k. \tag{36}$$

and proceeding by standard steps gives the optimal $\Delta u_{k+1}$ as

$$\Delta u_{k+1} = (R + {G_k}^{\mathrm{T}} P_k G_k)^{-1} {G_k}^{\mathrm{T}} P_k e_k. \tag{37}$$

Hence the optimal learning gain in the ILC law is

$$L_k^* = (R + {G_k}^{\mathrm{T}} P_k G_k)^{-1} {G_k}^{\mathrm{T}} P_k, \tag{38}$$

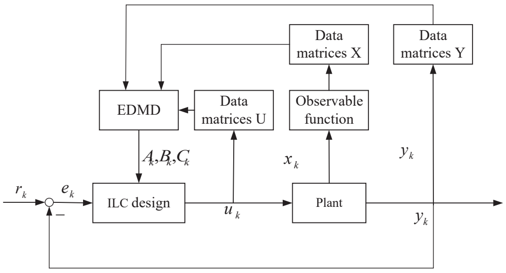

Figure 1: Block diagram of POILC based on the Koopman operator.

where $P_k$ is the solution of the discrete algebraic Riccati equation

$$P_k = Q + \gamma P_k - \gamma P_k G_k (R + {G_k}^{\mathrm{T}} P_k G_k)^{-1} {G_k}^{\mathrm{T}} P_k. \tag{39}$$

To compute $P_k$, of the various methods available, the iteration method is used (see, e.g., Reference [8]). This method is based on the iterative procedure given by the following equations

$$L_k{}^j = (R + {G_k}^{\mathrm{T}} P_k{}^j G_k)^{-1} {G_k}^{\mathrm{T}} P_k{}^j, \tag{40}$$

and

$$P_k{}^{j+1} = Q + \gamma {L_k{}^j}^{\mathrm{T}} R L_k{}^j + \gamma (I - G_k L_k{}^j)^{\mathrm{T}} P_k{}^j (I - G_k L_k{}^j). \tag{41}$$

The assumptions made (see, e.g., Reference [8, 20] for the details), ensure that the starting values can be taken as $L_k^0 = \hat{L}$ where $(I - G_k \hat{L})$ is a stability matrix and $P_k^0 = 0$. In application, the iteration in $j$ is run until a specified convergence is achieved, as specified in the design algorithm given next.

Figure 1 gives a block diagram of the ILC design, which can be computed by the algorithm given below.

Next convergence conditions for the various steps in this implementation algorithm are developed.

# 4 Convergence analysis

The approximation of the Koopman operator is updated at each trial. Furthermore there is modeling error between the approximate linear system and the original nonlinear system. Consequently, it is difficult to achieve perfect tracking, i.e., trial to trial error convergence to zero, whereas a robust tracking task can be achieved, where the tracking error converges to a small neighbourhood of the origin as the number of trials increases.

Firstly, it is important to give the results of the boundedness of the system trajectories. To this effect, the following assumption is required.

**Algorithm 1** POILC design based on the Koopman operator

```
Require: Initial input $u_0$ ; reference vector $r$; maximum number of trials
    considered $k_{max}$ and the error tolerance on $L_k^j$ in (??) and (??), denoted
    by $\varepsilon$.
Ensure: Input for the next trial $u_{k+1}$.
1: for $k = 0$ to $k_{max}$ do
2:     Input $u_k$ to the system to collect the data $x_k(t)$, $y_k(t)$, $e_k(t)$.
3:     Set $z_k(t) = \Psi(x_k(t))$, and collect the data set $\mathbf{X_1}$, $\mathbf{X_2}$, $U$.
4:     Compute (??) and (??) to obtain the matrices $(A_k, B_k, C_k)$.
5:     Solve (??) and (??) for $({L_k}^j, {P_k}^{j+1})$ (with starting values given above).
6:     Stop if $\|{L_k}^{j+1} - {L_k}^j\| \leq \varepsilon$, and ${L_k}^* = {L_k}^{j+1}$; otherwise, set $j = j + 1$
    and go to step 4.
7:     Update the input for the next trial $u_{k+1} = u_k + {L_k}^* e_k$.
8: end for
```

**Assumption 1** *Assume that for all* $k \in \mathbb{Z}_+$,

$$\|G_k\| \leq \beta_G, \|d_k\| \leq \beta_d,\ \|L_k\| \leq \beta_L,$$

$$\|r\| \leq \beta_r,\ \|u_0\| \leq \beta_{u_0},\ \|x_k(0)\| \leq \beta_{x_0},$$

*where* $\beta_G \geq 0, \beta_d \geq 0, \beta_L \geq 0, \beta_r \geq 0, \beta_{u_0} \geq 0, \beta_{x_0} \geq 0$ *are some finite bounds. Define* $\widetilde{G}_k = G_k + \Delta G_k$ *and* $\widetilde{d}_k = d_k + \Delta d_k$ *, for all* $k \in \mathbb{Z}_+$,

$$\|\widetilde{G}_k\| \leq \beta_{\widetilde{G}}, \|\widetilde{d}_k\| \leq \beta_{\widetilde{d}}$$

*where* $\beta_{\widetilde{G}} \geq 0$, $\beta_{\widetilde{d}} \geq 0$ *are some finite bounds.*

Given that the parameters of the system (31) vary with the trial number, convergence analysis for trial-variant systems is required(see, e.g., Reference [14, 13]).

As the boundedness of the control input is a fundamental prerequisite for the convergence of the tracking error, we derive a sufficient condition under the proposed control law to ensure that the input remains bounded across all trials. Based on the Assumption 1, the following theorem establishes the boundedness property of the design.

**Theorem 4.1** *Consider the lifted ILC system (31) under the ILC updating law (24), and Assumption 1 holds. If there exists constants* $C_1$ *and* $0 \leq \lambda_1 < 1$ *such that for all* $k \in \mathbb{Z}_+$, $j \in \mathbb{Z}_+$ *and* $k \geq j$

$$\|\prod_{i=j}^{k} (I - L_i \widetilde{G}_i)\| \leq C_1 \lambda_1^{k-j}. \tag{42}$$

*then the input $u_k$ and the output $y_k$ are bounded such that*

$$\sup_{k\in\mathbb{Z}_+}\|u_k\|\leq\beta_u,\ \sup_{k\in\mathbb{Z}_+}\|y_k\|\leq\beta_y, \tag{43}$$

*where* $\beta_u=\max_{k\in\mathbb{Z}_+}(\lambda_1^k\beta_{u_0}+(\frac{1-\lambda_1^k}{1-\lambda_1}C_1+1)\beta_L(\beta_r+\beta_{\widetilde{d}}))$ ,$\beta_y=\beta_{\widetilde{G}}\beta_u+\beta_{\widetilde{d}}$.

**Proof 1** *The input on trial $k+1$ can be written as*

$$\begin{aligned}u_{k+1}&=(I-L_k\widetilde{G}_k)u_k+L_k(r-\widetilde{d}_k)\\&=(I-L_k\widetilde{G}_k)(I-L_{k-1}\widetilde{G}_{k-1})u_{k-1}+(I-L_k\widetilde{G}_k)L_{k-1}(r-\widetilde{d}_{k-1})+L_k(r-\widetilde{d}_k)\\&=\prod_{i=0}^{k}(I-L_i\widetilde{G}_i)u_0+\sum_{j=0}^{k-1}\prod_{i=j+1}^{k}(I-L_i\widetilde{G}_i)L_j(r-\widetilde{d}_j)+L_k(r-\widetilde{d}_k)\end{aligned}. \tag{44}$$

*Taking norms on both sides of (44) gives*

$$\begin{aligned}\|u_{k+1}\|&\leq\|\prod_{i=0}^{k}(I-L_i\widetilde{G}_i)u_0\|+\|L_k(r-\widetilde{d}_k)\|+\|\sum_{j=0}^{k-1}(\prod_{i=j+1}^{k}(I-L_i\widetilde{G}_i))L_j(r-\widetilde{d}_j)\|\\&\leq\|\prod_{i=0}^{k}(I-L_i\widetilde{G}_i)\|\|u_0\|+\|L_k(r-\widetilde{d}_k)\|+\sum_{j=0}^{k-1}\|\prod_{i=j+1}^{k}(I-L_i\widetilde{G}_i)\|\|L_j(r-\widetilde{d}_j)\|,\\&\leq C\lambda^k\|u_0\|+\|L_k(r-\widetilde{d}_k)\|+\sum_{j=0}^{k-1}C\lambda^{k-j-1}\|L_j(r-\widetilde{d}_j)\|\end{aligned} \tag{45}$$

*where $L_k(r-\widetilde{d}_k)$ is bounded for all $k\in\mathbb{Z}_+$. Hence*

$$\begin{aligned}\|L_k(r-\widetilde{d}_k)\|&\leq\|L_k\|(\|r\|+\|\widetilde{d}_k\|)\\&\leq\beta_L(\beta_r+\beta_{\widetilde{d}})\end{aligned}, \tag{46}$$

*Provided that condition (42) holds, then (45) gives*

$$\begin{aligned}\|u_{k+1}\|&\leq C_1\lambda_1^k\|u_0\|+(\sum_{j=0}^{k-1}C_1\lambda_1^{k-j-1}+1)\beta_L(\beta_r+\beta_{\widetilde{d}})\\&\leq C_1\lambda_1^k\beta_{u_0}+\beta_L(\beta_r+\beta_{\widetilde{d}})+C_1\beta_L(\beta_r+\beta_{\widetilde{d}})(\lambda_1^0+\lambda_1^1+\cdots+\lambda_1^{k-1})\\&\leq C_1\lambda_1^k\beta_{u_0}+(\frac{1-\lambda_1^k}{1-\lambda_1}C_1+1)\beta_L(\beta_r+\beta_{\widetilde{d}})\\&\leq\beta_u\end{aligned}, \tag{47}$$

*and hence $u_k$ is bounded for all $k \in \mathbb{Z}_+$.*

*The final step in the proof is to use the boundedness of $u_k$ to prove that the output $y_k = \widetilde{G}_k u_k + \widetilde{d}_k$ is bounded. Taking the norm of $y_k$ leads to*

$$\begin{aligned} \|y_k\| &= \|\widetilde{G}_k u_k + \widetilde{d}_k\| \\ &\le \|\widetilde{G}_k\| \|u_k\| + \|\widetilde{d}_k\| \\ &\le \beta_{\widetilde{G}} \beta_u + \beta_{\widetilde{d}} \\ &= \beta_y \end{aligned}, \tag{48}$$

*and from (48), it follows that $y_k$ is bounded for all $k \in \mathbb{Z}_+$.*

**Remark 1** *The condition (42) in Theorems 4.1 can be replaced by $\|I - L_k \widetilde{G}_k\| \le \lambda_1, 0 \le \lambda_1 < 1, \forall k$.*

Theorem 4.1 gives conditions under which the system is bounded input ($u_k$) bounded output ($y_k$) stable. Hence the tracking error ($e_k$) is bounded, and next theorem is used to determine the convergence of the tracking error. In particular, routine manipulations enable $e_{k+1}$ can be written as

$$e_{k+1} = (I - \widetilde{G}_k L_k) e_k + (\widetilde{G}_k - \widetilde{G}_{k+1}) u_{k+1} + (\widetilde{d}_k - \widetilde{d}_{k+1}). \tag{49}$$

Then using (49), the following theorem gives the convergence properties of the tracking error $e_k$.

**Theorem 4.2** *Consider the system (31) when Assumption 1 holds. If (42) in Theorem 4.1 is satisfied then there exists constants $C_2$ and $0 \le \lambda_2 < 1$ such that for all $k \in \mathbb{Z}_+$, $j \in \mathbb{Z}_+$*

$$\|\prod_{i=j}^{k} (I - \widetilde{G}_i L_i)\| \le C_2 \lambda_2^{k-j}, \tag{50}$$

*where $k \ge j$, and the tracking error $e_k$ is bounded, i.e.,*

$$\sup_{k \in \mathbb{Z}_+} \|e_k\| \le \beta_e, \tag{51}$$

*where $\beta_e = \max\limits_{k \in \mathbb{Z}_+} (C_2 \lambda_2^k \beta_{e_0} + (2\beta_{\widetilde{G}} \beta_u + 2\beta_{\widetilde{d}})(\frac{1-\lambda_2^k}{1-\lambda_2} C_2 + 1))$ . Also the limiting property of the tracking error is*

$$\lim_{k \to \infty} \|e_k\| \le (2\beta_{\widetilde{G}} \beta_u + 2\beta_{\widetilde{d}})(\frac{1}{1-\lambda_2} C_2 + 1). \tag{52}$$

**Proof 2** *Using (49) and (31), $e_{k+1}$ can be written as*

$$\begin{aligned} e_{k+1} &= (I-\widetilde{G}_k L_k)e_k + (\widetilde{G}_k - \widetilde{G}_{k+1})u_{k+1} + (\widetilde{d}_k - \widetilde{d}_{k+1}) \\ &= \prod_{i=0}^{k}(I-\widetilde{G}_i L_i)e_0 + (\widetilde{G}_k - \widetilde{G}_{k+1})u_{k+1} + (\widetilde{d}_k - \widetilde{d}_{k+1}) \\ &\quad + \sum_{j=0}^{k-1}\prod_{i=j+1}^{k}(I-\widetilde{G}_i L_i)(\widetilde{G}_j - \widetilde{G}_{j+1})u_{j+1} \\ &\quad + \sum_{j=0}^{k-1}\prod_{i=j+1}^{k}(I-\widetilde{G}_i L_i)(\widetilde{d}_j - \widetilde{d}_{j+1}) \\ &= \prod_{i=0}^{k}(I-\widetilde{G}_i L_i)e_0 + \alpha_k + \sum_{j=0}^{k-1}\prod_{i=j+1}^{k}(I-\widetilde{G}_i L_i)\alpha_j \end{aligned} \quad . \tag{53}$$

*where* $\alpha_k = (\widetilde{G}_k - \widetilde{G}_{k+1})u_{k+1} + (\widetilde{d}_k - \widetilde{d}_{k+1})$.

*Taking the norm on both sides of (53) gives*

$$\begin{aligned} \|e_{k+1}\| &\leq \|\prod_{i=0}^{k}(I-\widetilde{G}_i L_i)e_0\| + \|\alpha_k\| + \sum_{j=0}^{k-1}\|\prod_{i=j+1}^{k}(I-\widetilde{G}_i L_i)\alpha_j\| \\ &\leq \|\prod_{i=0}^{k}(I-\widetilde{G}_i L_i)\|\|e_0\| + \|\alpha_k\| + \sum_{j=0}^{k-1}\|\prod_{i=j+1}^{k}(I-\widetilde{G}_i L_i)\|\|\alpha_j\|, \\ &\leq C_2\lambda_2^k\|e_0\| + \|\alpha_k\| + \sum_{j=0}^{k-1}C_2\lambda_2^{k-j-1}\|\alpha_j\| \end{aligned} \tag{54}$$

*where* $e_0 = r - \widetilde{G}_0 u_0 - \widetilde{d}_0$, *and*

$$\begin{aligned} \|e_0\| &= \|r - \widetilde{G}_0 u_0 - \widetilde{d}_0\| \\ &\leq \|r\| + \|\widetilde{G}_0\|\|u_0\| + \|\widetilde{d}_0\| \cdot \\ &\leq \beta_r + \beta_{\widetilde{G}}\beta_u + \beta_{\widetilde{d}} \end{aligned} \tag{55}$$

*For all* $k \in \mathbb{Z}_+$

$$\begin{aligned} \|\alpha_k\| &\leq \|(\widetilde{G}_k - \widetilde{G}_{k-1})u_{k+1}\| + \|(\widetilde{d}_k - \widetilde{d}_{k-1})\| \\ &\leq \|\widetilde{G}_k - \widetilde{G}_{k+1}\|\|u_{k+1}\| + \|(\widetilde{d}_k - \widetilde{d}_{k-1})\| \cdot \\ &\leq 2\beta_{\widetilde{G}}\beta_u + 2\beta_{\widetilde{d}} \end{aligned} \tag{56}$$

*Define $\beta_{e_0} = \beta_r + \beta_{\widetilde{G}}\beta_u + \beta_{\widetilde{d}}$. If condition (50) holds, then using (55) and (56) and, (54) can be written as*

$$\begin{aligned}\|e_{k+1}\| &\le C_2\lambda_2^k\beta_{e_0} + 2\beta_{\widetilde{G}}\beta_u + 2\beta_{\widetilde{d}} + (2C_2\beta_{\widetilde{G}}\beta_u + 2C_2\beta_{\widetilde{d}})(\lambda_2^0 + \lambda_2^1 + \cdots + \lambda_2^{k-1})\\ &\le C_2\lambda_2^k\beta_{e_0} + (2\beta_{\widetilde{G}}\beta_u + 2\beta_{\widetilde{d}})(\frac{1-\lambda_2^k}{1-\lambda_2}C_2 + 1) \qquad .\\ &\le \beta_e\end{aligned} \tag{57}$$

*Hence, $e_k$ is bounded, and (57) can be used to analyze the boundedness of $\|e_{k+1}\|$ as $k \to \infty$. Based on (57), and since condition (50) holds, it follows that if $k \to \infty$, then $\lambda_2^k \to 0$. Hence*

$$\lim_{k\to\infty} \|e_{k+1}\| \le (2\beta_{\widetilde{G}}\beta_u + 2\beta_{\widetilde{d}})(\frac{1}{1-\lambda_2}C_2 + 1). \tag{58}$$

Theorems 4.1 and 4.2 show that for robust ILC, if Assumption 1 holds, the inputs and outputs of system for all $k \in \mathbb{Z}_+$ are bounded, and the tracking error $e_k$ is bounded. Furthermore, as the trial number $k \to \infty$, the tracking error converges.

**Remark 2** *The condition (50) in Theorems 4.2 can be replaced by $\|I - \widetilde{G}_k L_k\| \le \lambda_2, 0 \le \lambda_2 < 1, \forall k$.*

# 5 Numerical case study

This section applies the new design to an example and also compares the results with those obtained for three other designs. The example is described by the following model

$$\begin{cases} x_k(t+1) &= \begin{bmatrix} \cos(x_k^{(1)}(t)) + 0.3x_k^{(1)}(t)x_k^{(2)}(t) \\ 0.4\sin(x_k^{(1)}(t)) + \cos(x_k^{(2)}(t)) + u_k(t) \end{bmatrix} \\ y_k(t) &= x_k^{(2)}(t) \end{cases},$$

where $x_k(t+1) = \begin{bmatrix} x_k^{(1)}(t+1) \\ x_k^{(2)}(t+1) \end{bmatrix}$. The reference trajectory is

$$y_r(t) = 0.8\sin(\frac{2\pi t}{50}) + 2\sin(\frac{2\pi t}{25}) + \sin(\frac{2\pi t}{10}),\ \ 0 \le t \le 50.$$

Next, the four designs are detailed.

For the new design of this paper, the design variable are chosen as: $Q = I$, $R = 5I$, $\gamma = 0.99$, $x_k(0) = 0$, ${P_k}^0 = 0$, ${L_k}^0 = 0$. The initial control input $u_0$ is a random signal distributed uniformly over the interval $[-0.5, 0.5]$.

In our simulations, we employed state itself (i.e.,$x^{(1)}$ and $x^{(2)}$) and thin plate spline (TPS) basis functions as the lifting functions in the EDMD formulation. TPS functions are known for their strong global approximation capabilities and smoothness, making them suitable for modeling systems with spatially distributed or smooth nonlinear dynamics. Compared to Gaussian radial basis functions, TPS is less sensitive to the selection of centers and bandwidth parameters, making it easier to implement. A typical TPS basis function is defined as:

$$\psi_i(x) = ||x - c_i||^2 \log(||x - c_i||). \tag{59}$$

where $c_i$ denotes the center point of the $i$th basis function, $c_i$ is chosen randomly with a uniform distribution over the unit box$[-1, 1]$, and $i \in [1, 10]$.

To compare performance, three other ILC designs are used: i) the data driven predictive design (DDPILC) [25], providing a direct comparison, ii) the P-type ILC design [15], providing a comparison with an ILC with no prediction in the trial to trial direction ($k$), and iii) a version of a data driven norm optimal ILC law (DDOILC) [5]. For ease of presentation in the cases of i) and iii) only the basics are given together with the simulation parameters used.

For the new design in this paper, the parameters: $Q = I$, $R = 5I$, $\gamma = 0.99$, $x_k(0) = 0$, ${P_k}^0 = 0$, ${L_k}^0 = 0$. The initial control input $u_0$ is a random signal distributed uniformly over the specified interval $[-0.5, 0.5]$.

The design in Reference [25], denoted by i), starts by expressing the input and output series as a set of algebraic functions, and the ILC law for implementation is of the form

$$u_{k+1}(t) = u_k(t) + U^T \Delta u_{k+1}^m(t), \tag{60}$$

where $U = \begin{bmatrix} I_n & 0 \end{bmatrix}$ ($I_n$ denotes the $n \times n$ identity matrix), and

$$\Delta u_{k+1}^m(t) = \frac{\gamma \hat{G}_{k+1}^{mT}(t) Q}{r + q||\hat{G}_{k+1}^{mT}(t)||^2} e_k(t+1), \tag{61}$$

where $q > 0$,$r > 0$,and $\gamma \in (0, 1]$ is introduced to make the control law more general. Also $\hat{G}_{k+1}^m(t)$ in 61 is the estimation of $G_{k+1}^m(t) = [G_{k+1}(t), G_{k+2}(t), \ldots, G_{k+m}(t)]$, where $G_k(t)$ is estimated as

$$\hat{G}_k(t) = \hat{G}_{k-1}(t) + \frac{\eta(\Delta y_{k-1}(t+1) - \hat{G}_{k-1}(t)\Delta u_{k-1}(t))\Delta u_{k-1}^T(t)}{\mu + ||\Delta_{k-1}(t)||^2}, \tag{62}$$

where $\eta \in (0, 2]$ and $\mu > 0$. The elements $G_{k+i}(t)$ are predicted using

$$\hat{G}_{k+i}(t) = \hat{\Xi}^T_{k+i-1}(t)\Gamma_k(t), \tag{63}$$

$$\hat{\Xi}_{k-1}(t) = [\hat{G}_{k-1}(t), \hat{G}_{k-2}(t), \dots, \hat{G}_{k-n_p}(t)], \tag{64}$$

$$\Gamma_k(t) = \Gamma_{k-1}(t) + \kappa\hat{\Xi}_{k-1}(t)[\nu I + \hat{\Xi}^T_{k-1}(t)\hat{\Xi}_{k-1}(t)]^{-1} \cdot [\hat{G}_k(t) - \hat{\Xi}^T_{k-1}(t)\Gamma_{k-1}(t)], \tag{65}$$

where $\kappa$ and $\nu$ are known positive constants, and $I$ is the identity matrix. The parameters used in the simulations are $q = 1$, $r = 5$, $\eta = 0.01$, $\mu = 5$, $\kappa = 0.01$, $\nu = 30$, $\gamma = 1$, $n_p = 3$,$m = 2$ and $\mu = 2$.

The ILC law under ii) above is

$$u_{k+1}(t) = u_k(t) + \alpha e_k(t), \tag{66}$$

where $\alpha$ is the learning gain, and in particular, $\alpha = 0.02$.

In the case of the ILC law under iii) above, the law is

$$u_k(t) = u_{k-1}(t) + \frac{\rho\hat{\Phi}^T_k e_{k-1}(t)}{\lambda + ||\hat{\Phi}_k||^2}, \tag{67}$$

where $\lambda > 0$ and $\rho > 0$ are scalars to be chosen, and $\hat{\Phi}_k$ is a lower triangular matrix, whose row entries, denoted by $\varphi_k(t)$ are estimated using

$$\hat{\varphi}_k(t) = \hat{\varphi}_{k-1}(t) + \frac{\eta(\Delta y_{k-1}(t+1) - \hat{\varphi}_{k-1}(t)\Delta u_{k-1}(t))\Delta u^T_{k-1}(t)}{\mu + ||\Delta_{k-1}(t)||^2}, \tag{68}$$

with a well defined starting sequence, and $\eta \in (0, 2)$ is a step size factor introduced to make the estimation algorithm more general and flexible. The parameters used in the simulations reported below are $\rho = 1.5$, $\lambda = 1.1$, $\eta = 1.5$, and $\mu = 2$.

The outputs plotted against the trial number for the new design in this paper are shown in Figure 2. From this plot, it can be observed that the output curves do not accurately track the reference trajectory on the initial trials. As the number of trials increases, however, the tracking accuracy improves.

Figure 3 plots the error on each trial against the trial number for the new design in this paper, where the tracking error is large in the initial trials. As the number of trials increases, the tracking error gradually decreases and approaches zero, and hence the design can achieve the control objective.

The convergence of quadratic performance index $J$ of the new design is shown in Figure 4. It can be seen that $J$ decreases with the number of trials to achieve optimal control.

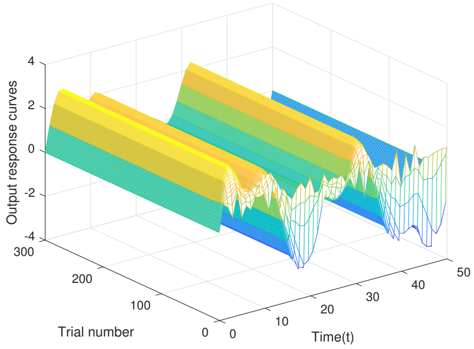

Figure 2: Output response curves for all trials.

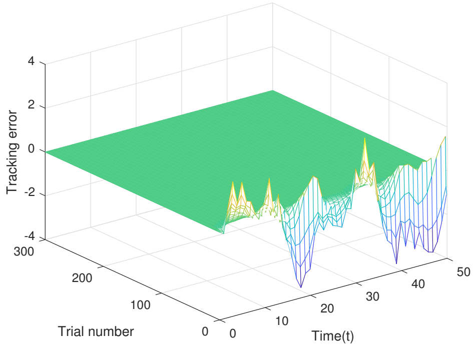

Figure 3: Tracking error curves for all trials.

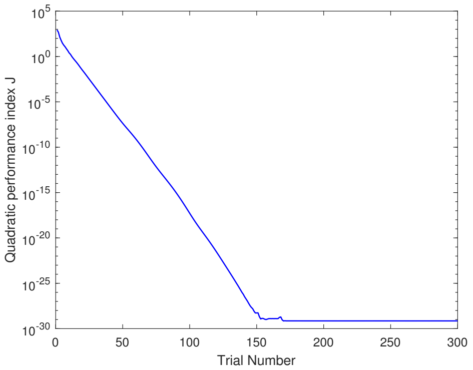

Figure 4: Convergence of the performance index $J$.

Figure 5 shows the modeling errors on each trial comparing the new design

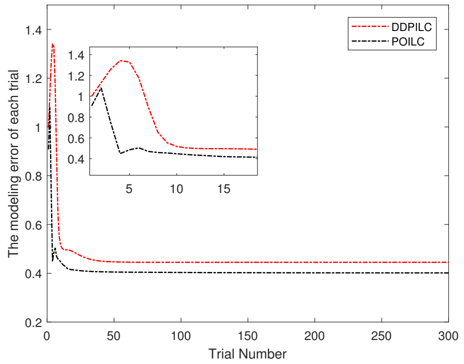

Figure 5: The modeling error plotted against the trial number.

with the alternative under i) above. The modeling error is computed as

$$\text{modeling error} = \frac{\sqrt{\sum\limits_{t=0}^{N-1} |y(t) - y_{\mathrm{m}}(t)|^2}}{\sqrt{\sum\limits_{t=0}^{N-1} |y(t)|^2}}, \tag{69}$$

where $y$ is the output of the original nonlinear system, and $y_m$ is the output of an approximately linear system with the same control input.

Figure 6 shows the root mean square tracking error, computed for trial $k$ as

$$\mathrm{RMSE}(k) = \sqrt{\frac{1}{N}\sum_{t=0}^{N-1} |e_k(t)|^2} \tag{70}$$

plotted against the trial number for all four designs considered

As observed from Figure 5 and Figure 6, the new design method achieves better control performance than the three alternatives considered. Compared with the P-type ILC scheme, the obtained control gain may not be optimal due to modeling errors, but it is sub-optimal. Therefore, the new design performs better than a fixed control gain P-type ILC. Compared with the DDOILC scheme, the new design uses predictive information about the inputs and outputs of future trials and can deliver better performance. Compared with the DDPILC design, the modeling error of both methods decreases with the increase in the number of trials, but the modeling error of the new design is smaller. Also, the new design has more prediction horizon than the DDPILC design, leading to better control performance.

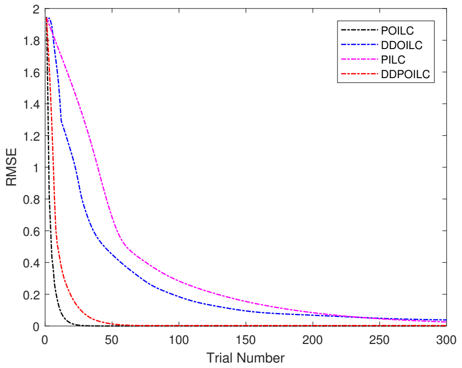

Figure 6: Root mean square tracking error curves plotted against the trial number for all four designs considered.

# 6 Conclusion

This paper has developed a new POILC design based on the Koopman operator for nonlinear systems. Based on the theory of this operator, a nonlinear dynamical system is lifted to a high dimensional space where the evolution of the nonlinear system is linear, and the approximate finite-dimensional value of the Koopman operator is computed using the EDMD method. Considering the modeling error of the approximate linear model with respect to the original nonlinear system, the POILC design for the nonlinear system is transformed into a design problem for a linear system with model uncertainty in the trial-to-trial direction, for which a convergence analysis, in the trial-to-trial direction, is developed. A numerical case study illustrates the design, including a detailed comparison against three alternative designs, one of which does also use predictive action in the trial-to-trial direction.

# Acknowledgments

This work was supported by the National Natural Science Foundation of China (62361136585), the National Science Centre in Poland (2020/39/B/ST7/01487), and 111 Project (B23008).

# Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

# References

[1] H S Ahn, Y Q Chen, and K L Moore. Iterative learning control: brief survey and categorization. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 37(6):1109–1121, 2007.

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

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

[4] Douglas A Bristow, Marina Tharayil, and Andrew G Alleyne. A survey of iterative learning control: A learning-based method for highperformance tracking control. *IEEE Control Syst. Mag*, 26(3):96–114, 2006.

[5] 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.

[6] Bing Chu, Andraes Rauh, Eric Aschemann, Haraldand Rogers, and David H Owens. Constrained iterative learning control for linear time-varying systems with experimental validation on a high-speed rack feeder. *IEEE Transactions on Control Systems Technology*, 30(5):1834–1846, 2022.

[7] Mingming Ha, Ding Wang, and Derong Liu. Novel discounted adaptive critic control designs with accelerated learning formulation. *IEEE Transactions on Cybernetics*, 54(5):3003–3016, 2023.

[8] Gary Hewer. An iterative technique for the computation of the steady state gains for the discrete optimal regulator. *IEEE Transactions on Automatic Control*, 16(4):382–384, 1971.

[9] Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

[10] Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the koopman operator. *Journal of Nonlinear Science*, 28:687–710, 2018.

[11] 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.

[12] Giorgos Mamakoukas, Maria L Castano, Xiaobo Tan, and Todd D Murphey. Derivative-based koopman operators for real-time control of robotic systems. *IEEE Transactions on Robotics*, 37(6):2173–2192, 2021.

[13] D. Meng and K. L Moore. Robust iterative learning control for non-repetitive uncertain systems. *IEEE Transactions on Automatic Control*, 16(2):907–913, 2016.

[14] Mikael Norrlöf and Svante Gunnarsson. Time and frequency domain convergence properties in iterative learning control. *International Journal of Control*, 75(14):1114–1126, 2002.

[15] James D Ratcliffe, Jari J Hätönen, Paul L Lewin, Eric Rogers, Thomas J Harte, and David H Owens. P-type iterative learning control for systems that contain resonance. *International Journal of adaptive control and signal processing*, 19(10):769–796, 2005.

[16] Eric Rogers, Bing Chu, Chris T Freeman, and Paul L Lewin. *Iterative learning control algorithms and experimental benchmarking*. Wiley, 2023.

[17] M. Bahadır Saltık, B. Jayawardhana, and A Cherukuri. Iterative learning and model predictive control for repetitive nonlinear systems via Koopman operator approximation. In *IEEE Conference on Decision and Control*, pages 3059–3065. IEEE, 2022.

[18] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[19] Lu Song and Shi Jingzhuo. Novel generalized predictive iterative learning speed controller for ultrasonic motors. *IEEE Access*, 8:29344–29353, 2020.

[20] Anton A Stoorvogel and Arie J T M Weeren. The discrete-time Riccati equation related to the $h_\infty$ control problem. *IEEE Transactions on Automatic Control*, 39(3):686–691, 1994.

[21] Ding Wang, Mingming Zhao, Mingming Ha, and Junfei Qiao. Stability and admissibility analysis for zero-sum games under general value iteration formulation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8707–8718, 2022.

[22] Liuping Wang, Chris T Freeman, and Eric Rogers. Predictive iterative learning control with experimental validation. *Control Engineering Practice*, 53:24–34, 2016.

[23] Youqing Wang, Donghua Zhou, and Furong Gao. Iterative learning model predictive control for multi-phase batch processes. *Journal of Process Control*, 18(6):543–557, 2008.

[24] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015.

[25] Qiongxia Yu and Zhongsheng Hou. Data-driven predictive iterative learning control for a class of multiple-input and multiple-output nonlinear systems. *Transactions of the Institute of Measurement and Control*, 38(3):266–281, 2016.