Journal of the Franklin Institute 363 (2026) 108414

Contents lists available at ScienceDirect

Journal of the Franklin Institute

journal homepage: www.elsevier.com/locate/fi

Fast data-driven iterative learning control for linear system with output disturbance

Jia Wang [a,*], Leander Hemelhof [a], Ivan Markovsky [b,c], Panagiotis Patrinos [a]

[a] Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, Kasteelpark Arenberg 10, 3001, Leuven, Belgium
[b] Credible Data-driven Models Group, International Centre for Numerical Methods in Engineering (CIMNE), Gran Capità, 08034, Barcelona, Spain
[c] Catalan Institution for Research and Advanced Studies (ICREA), Pg. Lluis Companys 23, 08010, Barcelona, Spain

ARTICLE INFO

Keywords:
Data-driven control
Iterative learning control
Convergence analysis

ABSTRACT

This paper studies data-driven iterative learning control (ILC) for linear time-invariant (LTI) systems with unknown dynamics, output disturbances and input box-constraints. Our main contributions are: 1) using a non-parametric data-driven representation of the system dynamics, for dealing with the unknown system dynamics in the context of ILC, 2) design of a fast ILC method for dealing with output disturbances, model uncertainty and input constraints. A complete design method is given in this paper, which consists of the data-driven representation, controller formulation, acceleration strategy and convergence analysis. A batch of numerical experiments and a case study on a high-precision robotic motion system are given in the end to show the effectiveness of the proposed method.

1. Introduction

Iterative learning control (ILC) was proposed for repetitive control tasks and has been widely used to improve the tracking accuracy in many industries, for example printing systems [1], atomic force microscopes [2] and continuously stirred tank reactors [3]. By using the previous tracking errors and control signals, ILC updates the current control signal to steer the system output to be as close as possible to the reference trajectory over a fixed time interval.

The existing solutions of ILC mainly focus on Lyapunov function-based adaptive ILC [4], contraction mapping-based proportional derivative ILC [5] and optimality-based ILC [6]. Optimality-based ILC is the most popular choice for real-world applications since it can reject the undesirable large transient behavior [3]. However, it has to use the model information in the algorithm design, e.g., the impulse response or state-space representation, hence, it is classified as a model-based control method [7]. In the case that the nominal model is not given orfirst-principle modelling is expensive to use, system identification is required. One of the most powerful methods is subspace identification [8], which has been used successfully in the ILC scenario [9,10], but only the white Gaussian disturbance is considered in the existing literature. In practice, the system output can be perturbed with structured disturbances such as the sine wave [11] and periodic disturbances [12]. Therefore, the data-driven representation method that can handle the structured disturbance effectively is worth investigating. Recently, Willems et al.'s fundamental lemma [13] has received a great attention in the control community, which offers a non-parametric data-driven representation of the system dynamics based on available input-output measurements [14,15], and has been successfully used in many applications [16,17]. Authors in the recent work [18] propose

* Corresponding author.
E-mail addresses: jia.wang@kuleuven.be (J. Wang), leander.hemelhof@kuleuven.be (L. Hemelhof), imarkovsky@cimne.upc.edu (I. Markovsky), panos.patrinos@esat.kuleuven.be (P. Patrinos).

https://doi.org/10.1016/j.jfranklin.2026.108414
Received 19 March 2025; Received in revised form 10 November 2025; Accepted 6 January 2026
Available online 11 January 2026
0016-0032/© 2026 The Franklin Institute. Published by Elsevier Inc. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

a novel data-driven ILC for linear time-invariant (LTI) systems with the aid of Willems et al.'s fundamental lemma, which is then extended to point-to-point ILC tasks [19]. In [20], a modified Willems et al.'s fundamental lemma is proposed to relax assumptions on the controllability and persistency of excitation, and then combined with the optimality-based ILC for LTI systems. However, the methods above assume no output disturbances. Inspired by [18], we apply a non-parametric data-driven representation to the optimality-based ILC under output disturbances, which can be promising to handle structured disturbances in the repetitive control process, due to its non-parametric nature. This constitutes the first motivation of this work.

On the other hand, the convergence speed is one of the most important performance indicators for the ILC. In recent years, there are many attempts to design ILC algorithms with a faster convergence speed. In [21,22], the classical gradient-based ILC is designed and robust monotonic convergence under multiplicative uncertainty is established. In [23], an accelerated version is proposed, which uses successive projection and a momentum term in the control signal update. In [24], a Quasi-Newton ILC method is proposed to improve the convergence speed and convergence is proved. Moreover, the accelerated gradient method [25] is combined with the gradient-based ILC in [26], and a faster convergence speed is achieved. However, the above mentioned accelerated methods assume to know the exact model and no output disturbances, which is impractical in many applications. Therefore, it is important to develop a fast ILC method based on the available, but inexact information. In [27], an inexact accelerated gradient method is proposed, which is an extension of the exact method in [28]. Inspired by [27], we attempt to combine the inexact accelerated method with the classical optimality-based ILC, and present the convergence rate under an inexact model and output disturbances, which is the second motivation of this paper.

Motivated by the above discussions, non-parametric data-driven representation based fast ILC is investigated in this paper, which aims to improve the ILC convergence speed and robustness under unknown and inexact information. The main contributions are summed up as follows:

- Non-parametric data-driven ILC is developed, which can deal efficiently with structured disturbances.
- The fast ILC approach with inexact information is developed. Moreover, the convergence rate of applying the inexact Nesterov accelerated gradient method into the ILC scenario is provided.
- A hybrid ILC approach is proposed with an empirical but effective switching condition between the fast and classical ILC, which inherits the advantages of these two methods, i.e., faster speed and better asymptotic performance.

The remainder of this paper is organized as follows. Section 2 gives the problem statement. Section 3 formulates the data-driven representation. In Section 4, fast ILC methods with inexact information are developed. Numerical experiments and a case study are given to verify the effectiveness of the proposed ILC framework in Section 5. Finally, Section 6 concludes this paper.

2. Problem statement

Let us start from the controllable single-input single-output (SISO) LTI system:

$$\begin{cases} x_j(k+1) = Ax_j(k) + Bu_j(k) \\ y_j(k) = Cx_j(k), \end{cases} \tag{1}$$

where j denotes the trial index, and $k \in \{0 \dots N\}$ denotes the time index. N is the data length in one trial. $x_j(k) \in \mathbb{R}^n$, $u_j(k) \in Q := [\underline{u}, \overline{u}] \subset \mathbb{R}$ and $y_j(k) \in \mathbb{R}$ represent the system state, input and output, respectively. $\underline{u}$ and $\overline{u}$ are the lower and upper bound on $u_j(k)$. A, B, C are real but unknown matrices with appropriate dimensions. The initial state is the same for all trials, i.e., $x_j(0) = x_0$ and assumed to be unknown.

Taking into account (1) with $k \in \{0 \dots N\}$, the relation between input and output can be characterized as follows.

$$\begin{aligned} y_j(1) &= Cx_j(1) \\ &= CAx_j(0) + CBu_j(0) \\ y_j(2) &= Cx_j(2) \\ &= CAx_j(1) + CBu_j(1) \\ &= CA^2x_j(0) + CABu_j(0) + CBu_j(1) \\ &\vdots \\ y_j(N) &= Cx_j(N) \\ &= CAx_j(N-1) + CBu_j(N-1) \\ &= CA^Nx_j(0) + CA^{N-1}Bu_j(0) + CA^{N-2}Bu_j(1) + \cdots + CBu_j(N-1). \end{aligned}$$

Then, the input/output mapping in the trial-domain can be written compactly as

$$y_j = Gu_j + c, \tag{2}$$

where

$$u_j := \begin{bmatrix} u_j(0) & u_j(1) & \cdots & u_j(N-1) \end{bmatrix}^\top \in \mathbb{R}^N \tag{3a}$$

$$y_j := \begin{bmatrix} y_j(1) & y_j(2) & \cdots & y_j(N) \end{bmatrix}^\top \in \mathbb{R}^N \tag{3b}$$

and

$$G = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{N-1}B & \cdots & CAB & CB \end{bmatrix} \in \mathbb{R}^{N\times N}, \quad c = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} x_0 \in \mathbb{R}^N \tag{4}$$

with $u_j \in \mathcal{Q}$ and $\mathcal{Q} = Q \times Q \times \cdots \times Q$ is the Cartesian product.

In this paper, we consider the SISO LTI system (1) in the presence of output disturbances. We therefore introduce the following definition.

Definition 1. The output disturbance at the kth time index of the jth trial is denoted by $d_j(k)$.

According to Definition 1, the available measurement of the system output is affected by the disturbance, i.e.,

$$\tilde{y}_j := y_j + d_j, \tag{5}$$

where

$$d_j := \begin{bmatrix} d_j(1) & d_j(2) & \cdots & d_j(N) \end{bmatrix}^\top \in \mathbb{R}^N.$$

The following assumption is given to guarantee the invertibility of G.

Assumption 1. The system (1) has relative degree 1, i.e., $CB \neq 0$.

To begin with, we introduce the basic idea of ILC in the case where G is known and there is no output disturbance, i.e., $d_j = 0$. Specifically, for a given reference trajectory $r = \begin{bmatrix} r(1) & \cdots & r(N) \end{bmatrix}^\top \in \mathbb{R}^N$, the goal is to determine the optimal input trajectory $u^* := \begin{bmatrix} u^*(0) & \cdots & u^*(N-1) \end{bmatrix}^\top \in \mathbb{R}^N$, which is obtained as the solution to the following constrained least-squares problem:

$$\underset{u\in\mathcal{Q}}{\text{minimize}} \; J(u) := \tfrac{1}{2}\|r - (Gu + c)\|^2, \tag{6}$$

where the dependence on the trial index j is omitted in (6), so that the expression $u \in \mathbb{R}^N$ refers to the variable itself rather than its iterative solution. The ILC law designed from the iterative solution of (6) is referred to as *optimality-based ILC*; see [6,7]. Specifically, given the simple structure of the box-constraint set $\mathcal{Q}$, a natural and effective way to solve (6) iteratively (trial by trial) is to use the projected gradient method:

$$u_{j+1} = \Pi_{\mathcal{Q}}\big(u_j - \rho\nabla J(u_j)\big), \tag{7}$$

in which $\Pi_{\mathcal{Q}}(\cdot)$ is the Euclidean projection on the set $\mathcal{Q}$ and the gradient of the cost function is

$$\nabla J(u_j) = -G^\top\big(r - (Gu_j + c)\big). \tag{8}$$

Moreover, the parameter $\rho \in (0, 2/L)$ serves as the step-size, where $L := \lambda_{\max}(G^\top G)$ is the Lipschitz constant and $\lambda_{\max}(\cdot)$ denotes the maximal eigenvalue. In the unconstrained case, the solution is given by $u^* = (G^\top G)^{-1}G^\top(r-c) = G^{-1}(r-c)$, since G is invertible for a SISO system under Assumption 1. The iterative process (7) is necessary since the matrix G can be ill-conditioned and thus sensitive to system inversion [26]. Finally, it should be emphasized that the iteration index j in (7) also corresponds to the trial index in the ILC framework.

In practice, however, the exact representation matrix G is typically unknown and the output channel is affected by measurement disturbances. Consequently, system identification and denoising techniques are required to estimate G from the noisy offline data. It should be emphasized that, in this case, only an inexact representation matrix can be obtained, which is denoted by $\tilde{G}$. Furthermore, during the online process, only the inexact gradient $\tilde{\nabla}J(u_j) := -\tilde{G}^\top(r - \tilde{y}_j)$ can be computed, since the available measurement is $\tilde{y}_j$ rather than y_j. On the other hand, fast convergence of the learning controller is strongly required in many practical real-time applications [24]. Therefore, motivated by the above considerations, the goal of this paper is to design a fast learning algorithm that generates an input sequence $\{u_j\}$ to rapidly reduce the tracking error using only the inexact information $\tilde{G}$ and $\tilde{\nabla}J(u_j)$.

3. Non-parametric data-driven representation

In this paper, a non-parametric data-driven representation of the system dynamics is used. Some basic formulations are given in the following subsection; see [15] for details. The noise filtering technique is introduced in the second subsection in order to obtain a more precise representation matrix.

3.1. Basic formulation

We begin the introduction of the basic formulation with the case where no output disturbances are present, let $w_{\mathrm{d}} := \begin{bmatrix} w_{\mathrm{d}}(0) & \cdots & w_{\mathrm{d}}(T) \end{bmatrix}^\top \in \mathbb{R}^{2(T+1)}$ be a $(T+1)$-sample long offline data trajectory of the SISO LTI system (1), in which $w_{\mathrm{d}}(i) =$

$\begin{bmatrix} u_{\mathrm{d}}(i) & y_{\mathrm{d}}(i) \end{bmatrix}^{\top}$ and the subscript 'd' denotes 'data'. Let $K \in \{0, \dots, T\}$, the $2(K+1) \times (T-K+1)$-dimensional *Hankel matrix* for the data trajectory is formed by

$$H_{K+1}(w_{\mathrm{d}}) := \begin{bmatrix} w_{\mathrm{d}}(0) & w_{\mathrm{d}}(1) & \cdots & w_{\mathrm{d}}(T-K) \\ w_{\mathrm{d}}(1) & w_{\mathrm{d}}(2) & \cdots & w_{\mathrm{d}}(T-K+1) \\ \vdots & \vdots & \ddots & \vdots \\ w_{\mathrm{d}}(K) & w_{\mathrm{d}}(K+1) & \cdots & w_{\mathrm{d}}(T) \end{bmatrix}. \tag{9}$$

Although $H_{K+1}(w_{\mathrm{d}})$ is well defined for all $K \in \{0, \dots, T\}$, we require $H_{K+1}(w_{\mathrm{d}})$ to have more columns than rows, which implies $K \leq \lfloor \frac{T-1}{3} \rfloor$ and $\lfloor a \rfloor$ is the largest integer smaller than or equal to a. In the following, we set $K = T_{\mathrm{ini}} + N$, where $T_{\mathrm{ini}} \geq l$ denotes the initial trajectory length and l is the lag of system (1).

Let $u_{\mathrm{d}} := \begin{bmatrix} u_{\mathrm{d}}(0) & \cdots & u_{\mathrm{d}}(T) \end{bmatrix}^{\top} \in \mathbb{R}^{T+1}$ and $y_{\mathrm{d}} := \begin{bmatrix} y_{\mathrm{d}}(0) & \cdots & y_{\mathrm{d}}(T) \end{bmatrix}^{\top} \in \mathbb{R}^{T+1}$, permuting and partitioning $H_{K+1}(w_{\mathrm{d}})$ lead to

$$H_{K+1}(u_{\mathrm{d}}) = \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ U_{\mathrm{b}} \end{bmatrix} \quad \text{and} \quad H_{K+1}(y_{\mathrm{d}}) = \begin{bmatrix} Y_{\mathrm{b}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix}, \tag{10}$$

where $U_{\mathrm{p}}, Y_{\mathrm{p}} \in \mathbb{R}^{T_{\mathrm{ini}} \times (T-K+1)}$, $U_{\mathrm{f}}, Y_{\mathrm{f}} \in \mathbb{R}^{N \times (T-K+1)}$ and $U_{\mathrm{b}}, Y_{\mathrm{b}} \in \mathbb{R}^{1 \times (T-K+1)}$. Then, let us define the following matrix:

$$\mathcal{H}_K(w_{\mathrm{d}}) := \begin{bmatrix} U_{\mathrm{p}}^{\top} & Y_{\mathrm{p}}^{\top} & U_{\mathrm{f}}^{\top} & Y_{\mathrm{f}}^{\top} \end{bmatrix}^{\top} \in \mathbb{R}^{2K \times (T-K+1)},$$

which plays an important role in our data-driven ILC algorithm. Note that the lifted model (4) introduces a time shift in trajectories u_j and y_j, as shown in (3). Consequently, $u_{\mathrm{d}}(T)$ and $y_{\mathrm{d}}(0)$ should be excluded from the data trajectory. Equivalently, this corresponds to removing the blocks U_{b} and Y_{b} in (10) when constructing the matrix $\mathcal{H}_K(w_{\mathrm{d}})$.

Let the input data trajectory $\begin{bmatrix} u_{\mathrm{d}}(0) & \cdots & u_{\mathrm{d}}(T-1) \end{bmatrix}^{\top}$ be *persistently exciting* of order $K+n$, then according to the *fundamental lemma* ([15] Lemma 2) and ([13]), there exists a vector $g \in \mathbb{R}^{T-K+1}$ such that

$$\mathcal{H}_K(w_{\mathrm{d}})g = \begin{bmatrix} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} \\ u_{\mathrm{f}} \\ y_{\mathrm{f}} \end{bmatrix} \tag{11}$$

for every trajectory $(u_{\mathrm{ini}}, y_{\mathrm{ini}}, u_{\mathrm{f}}, y_{\mathrm{f}})$ of system (1). Here, $u_{\mathrm{ini}}, y_{\mathrm{ini}} \in \mathbb{R}^{T_{\mathrm{ini}}}$ represent the initial condition, while the *future trajectory* $u_{\mathrm{f}}, y_{\mathrm{f}} \in \mathbb{R}^{N}$ evolves according to the lifted system (2), i.e., $y_{\mathrm{f}} = Gu_{\mathrm{f}} + c(u_{\mathrm{ini}}, y_{\mathrm{ini}})$, where $c(u_{\mathrm{ini}}, y_{\mathrm{ini}})$ denotes the initial response associated with u_{ini} and y_{ini}. It is worth noting that the original fundamental lemma in [13] appears as a special case of ([15], Lemma 2) when the system is described in an input/state representation; see ([15], Corollary 3) for a detailed discussion. The method proposed in this paper is developed directly from input-output data. Indeed, data-driven approaches of this kind have been successfully applied in various applications, for example [16,29].

Since the system is assumed to be controllable, the initial input trajectory u_{ini} determines the state x_0 as

$$x_0 = \begin{bmatrix} A^{T_{\mathrm{ini}}-1}B & A^{T_{\mathrm{ini}}-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u_{\mathrm{ini}}(0) \\ u_{\mathrm{ini}}(1) \\ \vdots \\ u_{\mathrm{ini}}(T_{\mathrm{ini}}-1) \end{bmatrix}, \tag{12}$$

which drives the system from the origin to x_0. The corresponding output trajectory is $y_{\mathrm{ini}} = \begin{bmatrix} y_{\mathrm{ini}}(1) & \cdots & y_{\mathrm{ini}}(T_{\mathrm{ini}}) \end{bmatrix}^{\top}$. Throughout this paper, the initial trajectory $(u_{\mathrm{ini}}, y_{\mathrm{ini}})$ is assumed to be unknown.

According to the *generalized persistency of excitation condition* in ([30], Corollary 5), for $K \geq l$, the range of $\mathcal{H}_K(w_{\mathrm{d}})$ captures all possible trajectories of (2) if and only if $\mathrm{rank}\big(\mathcal{H}_K(w_{\mathrm{d}})\big) = K + n$. In the presence of output disturbances, however, the Hankel matrix $\mathcal{H}_K(w_{\mathrm{d}})$ attains full row rank, thereby violating this rank condition. Consequently, as discussed in [30], the system behavior can no longer be exactly represented. To address this issue, we introduce a noise filtering technique in the following subsection.

3.2. Noise filtering

In this part, a heuristic method based on the low rank approximation is used to denoise the perturbed Hankel matrix. For clarity of presentation, we present the noise filtering method based on the matrix with output disturbances as follows:

$$U := \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix} \quad \text{and} \quad \tilde{Y} := \begin{bmatrix} \tilde{Y}_{\mathrm{p}} \\ \tilde{Y}_{\mathrm{f}} \end{bmatrix}, \tag{13}$$

where the blocks U_{p} and U_{f} are the same as in (10). In contrast, although $\tilde{Y}_{\mathrm{p}}$ and $\tilde{Y}_{\mathrm{f}}$ have the same structure as in (10), they are constructed from noisy data. Specifically, let $d_{\mathrm{d}} := \begin{bmatrix} d_{\mathrm{d}}(0) & \cdots & d_{\mathrm{d}}(T) \end{bmatrix}^{\top} \in \mathbb{R}^{T+1}$ and $\tilde{y}_{\mathrm{d}} := y_{\mathrm{d}} + d_{\mathrm{d}}$, then $\tilde{Y}_{\mathrm{p}}$ and $\tilde{Y}_{\mathrm{f}}$ are formed from $\tilde{y}_{\mathrm{d}}$ by applying (9) and (10).

To denoise the perturbed part $\tilde{Y}$, we follow the approach in [31]. The key idea of the noise filtering is to reduce the rank of $\begin{bmatrix} U^\top & \tilde{Y}^\top \end{bmatrix}$ from $2K$ to $K+n$, while keeping U unchanged, since there is no input disturbance. To this end, let us first apply the *QR decomposition* to $U^\top$ as follows:

$$U^\top = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}. \tag{14}$$

The QR decomposition factors a matrix into the product of an orthogonal matrix and an upper triangular matrix. Here, $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \in \mathbb{R}^{(T-K+1)\times(T-K+1)}$ as an orthogonal matrix, i.e., $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}^{-1} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}^\top$. Moreover, $R_1 \in \mathbb{R}^{K\times K}$ is an upper triangular matrix, while $R_2 \in \mathbb{R}^{(T-2K+1)\times K}$ is the zero matrix $\mathbf{0}$. The structure in (14) yields the orthogonal matrix $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$, which plays a key role in the denoising procedure. Specifically, left-multiplying $\begin{bmatrix} U^\top & \tilde{Y}^\top \end{bmatrix}$ by $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}^\top$ gives

$$\begin{bmatrix} Q_1^\top \\ Q_2^\top \end{bmatrix} \begin{bmatrix} U^\top & \tilde{Y}^\top \end{bmatrix} = \begin{bmatrix} R_1 & D_1 \\ \mathbf{0} & D_2 \end{bmatrix}, \tag{15}$$

which implies that the first block row $\begin{bmatrix} R_1 & D_1 \end{bmatrix}$ in (15) has full row rank, since R_1 is an upper triangular matrix. It is important to note that multiplication by an orthogonal matrix preserves the rank. Hence, we have

$$\operatorname{rank}\left(\begin{bmatrix} U \\ \tilde{Y} \end{bmatrix} \right) = \operatorname{rank}\left(\begin{bmatrix} R_1 & D_1 \\ \mathbf{0} & D_2 \end{bmatrix} \right).$$

Therefore, the task boils down to reducing the rank of $\begin{bmatrix} \mathbf{0} & D_2 \end{bmatrix}$ from K to n. To this end, we apply the *singular value decomposition (SVD)* to the matrix D_2, i.e., $D_2 = \mathcal{U}\Sigma\mathcal{V}^\top$. Since SVD is a standard tool for rank reduction, a rank-n approximation of the full rank matrix D_2 is then obtained by retaining the n largest singular values:

$$\bar{D}_2 = \sum_{i=1}^{n} \sigma_i \mathcal{U}_i \mathcal{V}_i^\top, \tag{16}$$

where σ_i denotes the ith largest singular value in Σ, and $\mathcal{U}_i$ and $\mathcal{V}_i$ are the corresponding ith columns of $\mathcal{U}$ and $\mathcal{V}$. By replacing D_2 to $\bar{D}_2$ in (15), we obtain

$$\operatorname{rank}\left(\begin{bmatrix} R_1 & D_1 \\ \mathbf{0} & \bar{D}_2 \end{bmatrix} \right) = \operatorname{rank}\left(\begin{bmatrix} Q_1^\top \\ Q_2^\top \end{bmatrix} \begin{bmatrix} U^\top & \tilde{Y}^\top \end{bmatrix} \right) = K + n,$$

where $\bar{Y}$ denotes the filtered output matrix and is computed as

$$\bar{Y} = \left(\begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} D_1 \\ \bar{D}_2 \end{bmatrix} \right)^\top.$$

In this way, we have $\operatorname{rank}\left(\begin{bmatrix} U^\top & \tilde{Y}^\top \end{bmatrix} \right) = K + n$. By defining

$$\bar{Y} := \begin{bmatrix} \bar{Y}_\mathrm{p} \\ \bar{Y}_\mathrm{f} \end{bmatrix} \quad \text{and} \quad \mathcal{H}_K(\bar{w}_\mathrm{d}) := \begin{bmatrix} U_\mathrm{p}^\top & \bar{Y}_\mathrm{p}^\top & U_\mathrm{f}^\top & \bar{Y}_\mathrm{f}^\top \end{bmatrix}^\top,$$

where $\bar{w}_\mathrm{d}$ is the filtered data trajectory corresponding to U and $\bar{Y}$, the method introduced in the above subsection can be used. Specifically, there exists a vector $\bar{g} \in \mathbb{R}^{T-K+1}$ such that

$$\mathcal{H}_K(\bar{w}_\mathrm{d})\bar{g} = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \\ u_j \\ y_j \end{bmatrix} \tag{17}$$

for the lifted system (2), in which the subscript 'j' denotes the trial index. Therefore, the vector $\bar{g}$ can be written as

$$\bar{g} = \begin{bmatrix} U_\mathrm{p} \\ \bar{Y}_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix}^\dagger \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \\ u_j \end{bmatrix} = \Phi u_j + \Psi \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \end{bmatrix},$$

where $\dagger$ denotes the pseudo-inverse, the matrices $\Phi \in \mathbb{R}^{(T-K+1)\times N}$ and $\Psi \in \mathbb{R}^{(T-K+1)\times 2T_\mathrm{ini}}$ are the partitioning of $\left(\begin{bmatrix} U_\mathrm{p}^\top & \bar{Y}_\mathrm{p}^\top & U_\mathrm{f}^\top \end{bmatrix}^\top \right)^\dagger$ corresponding to u_j and $\begin{bmatrix} u_\mathrm{ini}^\top & y_\mathrm{ini}^\top \end{bmatrix}^\top$. Then, the output trajectory in the jth trial is

$$y_j = \bar{Y}_\mathrm{f}\bar{g} = \bar{Y}_\mathrm{f}\Phi u_j + \bar{Y}_\mathrm{f}\Psi \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \end{bmatrix}. \tag{18}$$

The formulation (18) has a similar structure to (2), i.e., $\bar{Y}_\mathrm{f}\Phi$ corresponds to the representation matrix G and $\bar{Y}_\mathrm{f}\Psi\begin{bmatrix} u_\mathrm{ini}^\top & y_\mathrm{ini}^\top \end{bmatrix}^\top$ corresponds to the initial response. Furthermore, the data-driven representation matrix should have the same lower triangular structure

as G due to the causality. Therefore, a projection operator $\mathcal{P}_0(\cdot)$ is applied on $\bar{Y}_f\Phi$ in order to impose elements above the diagonal as 0. In this way, we obtain the data-driven representation from the noisy data:

$$\tilde{G} := \mathcal{P}_0(\bar{Y}_f\Phi). \tag{19}$$

The inexact representation (19) can be used to evaluated the function value and gradient. In the next section, the above inexact information is applied in the classical and fast gradient methods.

4. ILC design

In the case of no offline and online disturbances, the function value and gradient of (6) can be evaluated exactly under a given control trajectory u_j:

$$u_j \to \begin{cases} J(u_j) = \frac{1}{2}\|\varepsilon_j\|_2^2 \\ \nabla J(u_j) = -G^\top \varepsilon_j, \end{cases} \tag{20}$$

where ε_j is the tracking error without disturbances, i.e., $\varepsilon_j := r - y_j = r - (Gu_j + c)$. However, in practice, only inexact information is available due to the disturbance:

$$u_j \to \begin{cases} \tilde{J}(u_j) = \frac{1}{2}\|e_j\|_2^2 \\ \tilde{\nabla} J(u_j) = -\tilde{G}^\top e_j, \end{cases} \tag{21}$$

where $e_j := r - \tilde{y}_j = r - (Gu_j + c + d_j)$. Therefore, the goal of the ILC algorithm is to approximate the optimal input trajectory u^* with the increase of the trial number. To this end, the following assumption is introduced to bound the disturbance.

Assumption 2. Each element of the unknown disturbance trajectory is bounded, i.e., $\exists\ \bar{d} > 0$ such that $|d_j(k)| \le \bar{d},\ \forall k, j \ge 0$.

This assumption is commonly used in ILC or data-driven control to ensure the robustness for resisting the worst-case errors [1]. The physical meaning of the l_∞ norm of the disturbance is related to sensor errors, for example an accuracy range of $\pm 0.005\Omega$ for a pressure sensor. Then, the Euclidean norm of d_j is determined by the data length N and bounded by $\|d_j\|_2 \le \sqrt{N}\bar{d}$. Notice that only finite N is considered in this work since it is more common in practical applications. Under Assumption 2, the following lemma bounds the maximal error between exact and inexact information.

Lemma 1. *Let Assumption 2 hold, the difference between the pairs $(J(u_j), \nabla J(u_j))$ and $(\tilde{J}(u_j), \tilde{\nabla} J(u_j))$ is bounded by following equations:*

$$|J(u_j) - \tilde{J}(u_j)| \le \Delta_1, \tag{22a}$$

$$\|\nabla J(u_j) - \tilde{\nabla} J(u_j)\|_2 \le \Delta_2, \tag{22b}$$

where Δ_1 and Δ_2 are defined as

$$\Delta_1 := \sqrt{2\bar{J}N}\bar{d} + \frac{1}{2}N\bar{d}^2, \tag{23a}$$

$$\Delta_2 := \left\|\tilde{G} - G\right\|_2 \sqrt{2\bar{J}} + \left\|\tilde{G}\right\|_2 \sqrt{N}\bar{d}, \tag{23b}$$

with $\bar{J} := \max_{u\in\mathcal{Q}} J(u)$.

Proof. Considering the function values in (20) and (21), the difference is

$$\begin{aligned} J(u_j) - \tilde{J}(u_j) &= \frac{1}{2}\|r - (Gu_j + c)\|_2^2 - \frac{1}{2}\|r - (Gu_j + c + d_j)\|_2^2 \\ &= \langle r - (Gu_j + c), d_j\rangle - \frac{1}{2}\|d_j\|_2^2. \end{aligned}$$

Since $\mathcal{Q}$ is a convex and compact set, and $J(u)$ is a convex function, the value $\bar{J} := \max_{u\in\mathcal{Q}} J(u)$ is finite. Then, according to the Cauchy-Schwarz inequality, we have

$$\begin{aligned} |J(u_j) - \tilde{J}(u_j)| &\le \left|\langle r - (Gu_j + c), d_j\rangle\right| + \frac{1}{2}\|d_j\|_2^2 \\ &\le \|r - (Gu_j + c)\|_2\|d_j\|_2 + \frac{1}{2}N\bar{d}^2 \\ &\le \sqrt{2\bar{J}N}\bar{d} + \frac{1}{2}N\bar{d}^2. \end{aligned}$$

On the other hand, let us consider the difference of the gradient in (20) and (21):

$$\begin{aligned} \|\nabla J(u_j) - \tilde{\nabla} J(u_j)\|_2 &= \left\|\left(\tilde{G}^\top - G^\top\right)\left(r - (Gu_j + c)\right) - \tilde{G}^\top d_j\right\|_2 \\ &\le \left\|\left(\tilde{G}^\top - G^\top\right)\left(r - (Gu_j + c)\right)\right\|_2 + \left\|\tilde{G}^\top d_j\right\|_2 \\ &\le \left\|\tilde{G}^\top - G^\top\right\|_2 \left\|r - (Gu_j + c)\right\|_2 + \left\|\tilde{G}^\top d_j\right\|_2 \\ &\le \left\|\tilde{G} - G\right\|_2 \sqrt{2\bar{J}} + \left\|\tilde{G}\right\|_2 \sqrt{N}\bar{d}. \end{aligned}$$

In this way, the bounds in (22a) and (22b) are obtained. □

The following lemma provides two inequalities between the exact and inexact information, which is crucial for the convergence proof of our proposed ILC algorithm.

Lemma 2. *Let Assumption 2 hold and consider the functions:*

$$\begin{cases} J_{\delta,\bar{L}}(u_j) := \tilde{J}(u_j) - \Delta_1 - \Delta_2 D \\ G_{\delta,\bar{L}}(u_j) := \tilde{\nabla} J(u_j) \end{cases} \tag{24}$$

with $D := \sup_{a,b\in\mathcal{Q}} \|a-b\|_2$, $\tilde{L} := \lambda_{\max}(\tilde{G}^\top \tilde{G})$, $\Delta_3 := |L - \tilde{L}|$ *and* $\delta := 2\Delta_1 + 2\Delta_2 D + \frac{\Delta_3}{2} D^2$. *Then, the following inequalities hold for any* $u_j \in \mathcal{Q}$*:*

$$0 \le J(u_j') - \Big(J_{\delta,\bar{L}}(u_j) + \langle G_{\delta,\bar{L}}(u_j), u_j' - u_j\rangle\Big) \le \frac{\tilde{L}}{2}\|u_j' - u_j\|_2^2 + \delta, \quad \forall u_j' \in \mathcal{Q}. \tag{25}$$

Proof. Let us prove the left inequality first. Specifically, with the aid of Lemma 1, the convexity of $J(u_j)$ and the Cauchy-Schwarz inequality, we have

$$\begin{aligned} J(u_j') &\ge J(u_j) + \langle \nabla J(u_j), u_j' - u_j\rangle \\ &\overset{22a}{\ge} \tilde{J}(u_j) - \Delta_1 + \langle \nabla J(u_j) - \tilde{\nabla} J(u_j) + \tilde{\nabla} J(u_j), u_j' - u_j\rangle \\ &\ge \tilde{J}(u_j) - \Delta_1 + \langle \tilde{\nabla} J(u_j), u_j' - u_j\rangle - \|\nabla J(u_j) - \tilde{\nabla} J(u_j)\|_2 \|u_j' - u_j\|_2 \\ &\overset{22b}{\ge} \tilde{J}(u_j) - \Delta_1 + \langle \tilde{\nabla} J(u_j), u_j' - u_j\rangle - \Delta_2 D \end{aligned}$$

for all $u_j, u_j' \in \mathcal{Q}$. Second, let us prove the right inequality in (25) as follows:

$$\begin{aligned} J(u_j') &\le J(u_j) + \langle \nabla J(u_j), u_j' - u_j\rangle + \frac{L}{2}\|u_j' - u_j\|_2^2 \\ &\overset{22a}{\le} \tilde{J}(u_j) + \Delta_1 + \langle \nabla J(u_j) - \tilde{\nabla} J(u_j) + \tilde{\nabla} J(u_j), u_j' - u_j\rangle + \frac{L}{2}\|u_j' - u_j\|_2^2 \\ &\le \tilde{J}(u_j) + \Delta_1 + \langle \tilde{\nabla} J(u_j), u_j' - u_j\rangle + \|\nabla J(u_j) - \tilde{\nabla} J(u_j)\|_2 \|u_j' - u_j\|_2 + \frac{L}{2}\|u_j' - u_j\|_2^2 \\ &\overset{22b}{\le} \tilde{J}(u_j) + \Delta_1 + \langle \tilde{\nabla} J(u_j), u_j' - u_j\rangle + \Delta_2 D + \frac{L}{2}\|u_j' - u_j\|_2^2 \\ &\le \tilde{J}(u_j) + \Delta_1 + \langle \tilde{\nabla} J(u_j), u_j' - u_j\rangle + \Delta_2 D + \frac{\tilde{L} + \Delta_3}{2}\|u_j' - u_j\|_2^2 \\ &\le J_{\delta,\bar{L}}(u_j) + \langle G_{\delta,\bar{L}}(u_j), u_j' - u_j\rangle + \frac{\tilde{L}}{2}\|u_j' - u_j\|_2^2 + \underbrace{2\Delta_1 + 2\Delta_2 D + \frac{\Delta_3}{2} D^2}_{\delta :=} \end{aligned}$$

for all $u_j, u_j' \in \mathcal{Q}$. In this way, we obtain (25). □

Remark 1. After the offline data is collected, the inexact representation $\tilde{G}$ is fixed, which implies that the parameters in Lemmas 1 and 2 are time-invariant. Moreover, Δ_1, Δ_2 and Δ_3 do not have to be known, as they only appear in the convergence proof.

Applying the inexact gradient in (7), we have the following iterative algorithm:

$$u_{j+1} = \arg\min_{u\in\mathcal{Q}} \left\{ \langle \tilde{\nabla} J(u_j), u - u_j\rangle + \tfrac{\bar{L}}{2}\|u - u_j\|_2^2 \right\} \tag{26a}$$

$$\Rightarrow u_{j+1} = \Pi_{\mathcal{Q}}\Big(u_j + \tfrac{1}{\bar{L}}\tilde{G}^\top e_j\Big). \tag{26b}$$

The ILC update law (26) can be summarized in Algorithm 1 and the convergence is shown in Theorem 1.

Theorem 1. *Let Assumptions 1 and 2 hold, consider the classical data-driven ILC in Algorithm 1, we have the following convergence result:*

$$\sum_{j=0}^{M-1} \big(J(u_{j+1}) - J(u^*)\big) \le \frac{\tilde{L}}{2}\|u^* - u_0\|^2 + \delta M. \tag{27}$$

Proof. The first-order optimality condition of (26a) implies

$$\langle \tilde{\nabla} J(u_j) + \tilde{L}(u_{j+1} - u_j), u_{j+1}' - u_{j+1}\rangle \ge 0, \quad \forall u_{j+1}' \in \mathcal{Q}.$$

Moreover, note that

$$\begin{aligned} \|u_{j+1} - u^*\|^2 &= \|u_{j+1} - u_j\|^2 + \|u_j - u^*\|^2 + 2\langle u_{j+1} - u_j, u_j - u^*\rangle \\ &= -\|u_{j+1} - u_j\|^2 + \|u_j - u^*\|^2 + 2\langle u_{j+1} - u_j, u_{j+1} - u^*\rangle. \end{aligned}$$

Algorithm 1 Classical data-driven iterative learning control.

Input: Data-driven representation $\tilde{G}$, Lipschitz constant $\tilde{L}$, reference trajectory r, initial trajectory u_0 and maximal iteration number M.
Output: ILC sequence $\{u_j\}$.
1: **for** $j = 0 : M-1$ **do**
2: Measure the perturbed system output $\tilde{y}_j$.
3: Compute the tracking error $e_j = r - \tilde{y}_j$.
4: Compute the inexact gradient $\tilde{\nabla} J(u_j) = -\tilde{G}^\top e_j$.
5: Compute
$u_{j+1} = \arg\min_{u\in Q} \left\{ \langle \tilde{\nabla} J(u_j), u - u_j\rangle + \frac{\tilde{L}}{2}\|u - u_j\|_2^2 \right\}$
6: Apply u_{j+1} to the system.
7: **end for**

Then, let $u'_{j+1} = u^*$ and in the light of Lemma 2, we have

$$\begin{aligned}\|u_{j+1} - u^*\|^2 &\le -\|u_{j+1} - u_j\|^2 + \|u_j - u^*\|^2 + \tfrac{2}{\tilde{L}}\langle \tilde{\nabla} J(u_j), u^* - u_{j+1}\rangle \\ &= \|u_j - u^*\|^2 + \tfrac{2}{\tilde{L}}\langle \tilde{\nabla} J(u_j), u^* - u_j\rangle - \tfrac{2}{\tilde{L}}\left(\langle \tilde{\nabla} J(u_j), u_{j+1} - u_j\rangle + \tfrac{\tilde{L}}{2}\|u_{j+1} - u_j\|^2\right) \\ &\le \|u_j - u^*\|^2 + \tfrac{2}{\tilde{L}}\left(J(u^*) - J_{\delta,\tilde{L}}(u_j)\right) - \tfrac{2}{\tilde{L}}\left(J(u_{j+1}) - J_{\delta,\tilde{L}}(u_j) - \delta\right).\end{aligned}$$

By summing the above inequalities from $j = 0, \dots, M-1$, we obtain (27). □

As discussed in ([27], Section 6.2), the inequality (27) implies the following convergence rate:

$$J(u_j) - J(u^*) \le \frac{\tilde{L}\|u^* - u_0\|^2}{4j} + \delta. \tag{28}$$

Moreover, we use the fast gradient method with the inexact information to solve the ILC problem, which is summarized in Algorithm 2 and the convergence result is given in Theorem 2.

Algorithm 2 Fast data-driven iterative learning control.

Input: Data-driven representation $\tilde{G}$, Lipschitz constant $\tilde{L}$, reference trajectory r, initial trajectory u_0 and maximal iteration number M.
Output: ILC sequence $\{u_j\}$.
1: **for** $j = 0 : M-1$ **do**
2: Measure the perturbed system output $\tilde{y}_j$.
3: Compute the tracking error $e_j = r - \tilde{y}_j$.
4: Compute the inexact gradient $\tilde{\nabla} J(u_j) = -\tilde{G}^\top e_j$.
5: Compute
$\mu_j = \arg\min_{\mu\in Q} \left\{ \langle \tilde{\nabla} J(u_j), \mu - u_j\rangle + \frac{\tilde{L}}{2}\|\mu - u_j\|_2^2 \right\}$
6: Compute
$v_j = \arg\min_{v\in Q} \left\{ \frac{1}{2}\|v - u_0\|_2^2 + \sum_{i=0}^{j} \frac{i+1}{2\tilde{L}}\langle \tilde{\nabla} J(u_i), v - u_i\rangle \right\}$
7: Compute $u_{j+1} = \frac{2}{j+3}v_j + \frac{j+1}{j+3}\mu_j$
8: Apply u_{j+1} to the system.
9: **end for**

Theorem 2. *Let Assumptions 1 and 2 hold, consider the fast data-driven ILC in Algorithm 2, we have the following convergence result:*

$$J(\mu_j) - J(u^*) \le \frac{2\tilde{L}\|u^* - u_0\|^2}{(j+1)(j+2)} + \frac{j+3}{3}\delta. \tag{29}$$

Proof. With the aid of Lemma 2, Theorem 4 in [27] implies that

$$\begin{aligned}\frac{J(\mu_j)}{2\tilde{L}}\sum_{i=0}^{j}(i+1) &\le \min_{v\in Q}\left\{ \tfrac{1}{2}\|v - u_0\|^2 + \textstyle\sum_{i=0}^{j}\frac{i+1}{2\tilde{L}}\left(J_{\delta,\tilde{L}}(u_i) + \langle G_{\delta,\tilde{L}}(u_i), v - u_i\rangle\right)\right\} \\ &\quad + \frac{\delta}{2\tilde{L}}\sum_{i=0}^{j}\sum_{k=0}^{i}(k+1).\end{aligned}$$

The minimization in the right hand side follows the inequality:

$$\begin{aligned}
&\min_{v\in Q}\left\{\tfrac{1}{2}\|v-u_0\|^2+\textstyle\sum_{i=0}^{j}\tfrac{i+1}{2\tilde{L}}\left(J_{\delta,\tilde{L}}(u_i)+\langle G_{\delta,\tilde{L}}(u_i),v-u_i\rangle\right)\right\}\\
&\le \frac{1}{2}\|u^*-u_0\|^2+\sum_{i=0}^{j}\frac{i+1}{2\tilde{L}}\left(J_{\delta,\tilde{L}}(u_i)+\langle G_{\delta,\tilde{L}}(u_i),u^*-u_i\rangle\right)\\
&\overset{25}{\le} \frac{1}{2}\|u^*-u_0\|^2+\sum_{i=0}^{j}\frac{i+1}{2\tilde{L}}J(u^*)\\
&= \frac{1}{2}\|u^*-u_0\|^2+\frac{(j+1)(j+2)}{4\tilde{L}}J(u^*).
\end{aligned}$$

Then, one obtains the result:

$$\frac{(j+1)(j+2)}{4\tilde{L}}J(\mu_j)\le\frac{1}{2}\|u^*-u_0\|^2+\frac{(j+1)(j+2)}{4\tilde{L}}J(u^*)+\delta\sum_{i=0}^{j}\frac{(i+1)(i+2)}{4\tilde{L}},$$

which leads to

$$J(\mu_j)-J(u^*)\le\frac{2\tilde{L}\|u^*-u_0\|^2}{(j+1)(j+2)}+\frac{\delta}{6}\frac{(j+1)(j+2)(2j+6)}{(j+1)(j+2)}.$$

In this way, we have the convergence rate (29). □

Note that the result (29) characterizes the convergence of the algorithm in terms of μ_j, rather than u_{j+1}. However, as the iterations increase, μ_j and u_{j+1} tend to coincide, since $u_{j+1}=\frac{2}{j+3}v_j+\frac{j+1}{j+3}\mu_j$. Therefore, the result (29) can also reflect the convergence of u_{j+1}. On the other hand, the result (28) shows that there is no error accumulation and the upper bound of $J(u_j)-J(u^*)$ converges to δ with the rate $\mathcal{O}(1/j)$. In contrast, the first term in (29) decreases with $\mathcal{O}(1/j^2)$, but the second term increases with $\mathcal{O}(j)$, which implies that the fast ILC is asymptotically divergent. However, the divergent term can only dominate the convergence rate after a few iterations. Therefore, the fast ILC is superior to the classical ILC at the early iteration stage.

Based on this observation, a natural way to improve the convergence performance is to combine the fast and classical ILC methods, which is called hybrid ILC in this paper. Specifically, the fast ILC is used first in order to decrease the tracking error rapidly, and then the algorithm is switched to the classical ILC for a better asymptotic performance. Assuming that the algorithm is switched at the iteration s, it is easy to see that the hybrid ILC has the same convergence result as the fast ILC when $j\le s$, and after $j>s$, the classical ILC convergence result initialized from u_s is inherited. There are two merits of applying this hybrid method: (i), the classical ILC can start from u_s early. The classical ILC usually takes more iterations to calculate the same input u_s compared to the fast ILC. (ii), the divergence effect of the fast ILC is removed. The input u_s can be improved further by using the classical ILC, rather than oscillating due to the error accumulation.

Algorithm 3 Hybrid data-driven iterative learning control.

Input: Data-driven representation $\tilde{G}$, Lipschitz constant $\tilde{L}$, reference trajectory r, initial trajectory u_0, maximal iteration number M, moving horizon length W and initial switching flag $SF=0$.
Output: ILC sequence $\{u_j\}$.
for $j=0:M-1$ **do**
  Measure the perturbed system output $\tilde{y}_j$.
  Compute the tracking error $e_j=r-\tilde{y}_j$.
  Compute the inexact gradient $\tilde{\nabla}J(u_j)=-\tilde{G}^\top e_j$.
  if $j\ge W$ and $SF=0$ **then**
    Compute the averaged tracking error within the current time window $[j-W+1,j]$: $\bar{e}_j=\frac{1}{W}\sum_{i=j-W+1}^{j}\|e_i\|$
    Compute the averaged tracking error within the previous time window $[j-W,j-1]$: $\bar{e}_{j-1}=\frac{1}{W}\sum_{i=j-W}^{j-1}\|e_i\|$
    if $\bar{e}_j>\bar{e}_{j-1}$ **then**
      Set $SF=1$.
    end if
  end if
  if $SF=0$ **then**
    Invoke steps 5 − 7 in Algorithm 2 (Fast ILC).
  else if $SF=1$ **then**
    Invoke step 5 in Algorithm 1 (Classical ILC).
  end if
  Apply u_{j+1} to the system.
end for

The switching iteration s is determined by some switching conditions, which is checked at each iteration. In this paper, we propose an empirical switching condition, which is simple but efficient in experiments. Specifically, we calculate the average tracking error norm $\bar{e}_j$ at the jth iteration within the time window $[j-W+1, j]$, in which the window length W is determined by the user, and then compare with the previous average tracking error norm $\bar{e}_{j-1}$. If $\bar{e}_j > \bar{e}_{j-1}$, then the algorithm switches to the classical ILC. The proposed hybrid ILC method is summarized in Algorithm 3. Lines $5-11$ summarize the detection of the switching condition. In line 13, the fast ILC is used to decrease the tracking error rapidly at the early iteration stage. Line 15 shows that the classical ILC is used for a better asymptotic tracking performance after the switching.

Remark 2. The switching condition of the hybrid ILC is determined by the user, we only provide a simple one in Algorithm 3. The essence is to observe the tracking error e_j, if e_j becomes oscillating or starts to diverge, for example around the step 50 in Fig. 3(a), then one should switch to the classical ILC in order to reduce the tracking error further.

5. Illustrative examples

5.1. Implementation

In this section, we test the proposed methods in Algorithms 1–3 on a batch of systems. We also test different data-driven representations in (2): the proposed non-parametric data-driven representation $\tilde{G}_{\mathrm{DD}}$ and the parametric representation obtained by *subspace identification* $\tilde{G}_{\mathrm{SI}}$, in which DD and SI denote the non-parametric direct data-driven method obtained by the fundamental lemma and the parametric representation obtained by subspace identification, respectively.

All experiments are conducted using MATLAB R2018a on a machine equipped with a 9th-generation Intel Core i7 processor. The MATLAB function `n4sid`, which provides a practical implementation of subspace identification [32], is used to obtain $\tilde{G}_{\mathrm{SI}}$. This function requires the data trajectories u_{d}, $\tilde{y}_{\mathrm{d}}$, and the system order n as inputs.

To begin with, we briefly review the concept of the subspace identification method and the implementation of `n4sid` for completeness; see [8,32,33] for more details. Given the offline data trajectories $u_{\mathrm{d}} = \begin{bmatrix} u_{\mathrm{d}}(0) & \cdots & u_{\mathrm{d}}(T) \end{bmatrix}^\top$ and $\tilde{y}_{\mathrm{d}} = \begin{bmatrix} \tilde{y}_{\mathrm{d}}(0) & \cdots & \tilde{y}_{\mathrm{d}}(T) \end{bmatrix}^\top$, the Hankel matrices are constructed in the same manner as in (13):

$$H_{N_{\mathrm{p}}+N_{\mathrm{f}}}(u_{\mathrm{d}}) = \begin{bmatrix} U_{\mathrm{SI,p}} \\ U_{\mathrm{SI,f}} \end{bmatrix} \quad \text{and} \quad H_{N_{\mathrm{p}}+N_{\mathrm{f}}}(\tilde{y}_{\mathrm{d}}) = \begin{bmatrix} Y_{\mathrm{SI,p}} \\ Y_{\mathrm{SI,f}} \end{bmatrix}, \tag{30}$$

where $U_{\mathrm{SI,p}}, Y_{\mathrm{SI,p}} \in \mathbb{R}^{N_{\mathrm{p}} \times (T-N_{\mathrm{p}}-N_{\mathrm{f}}+2)}$ and $U_{\mathrm{SI,f}}, Y_{\mathrm{SI,f}} \in \mathbb{R}^{N_{\mathrm{f}} \times (T-N_{\mathrm{p}}-N_{\mathrm{f}}+2)}$. Note that the number of block rows N_{p} and N_{f} are selected automatically by `n4sid`. Based on which, we have the relation:

$$Y_{\mathrm{SI,f}} = \Gamma X_{\mathrm{SI,p}} + \Theta U_{\mathrm{SI,f}} + D_{\mathrm{SI,f}}, \tag{31}$$

where

$$\Gamma = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N_{\mathrm{f}}-1} \end{bmatrix} \quad \text{and} \quad \Theta = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ CB & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ CA^{N_{\mathrm{f}}-2}B & \cdots & CB & 0 \end{bmatrix}. \tag{32}$$

The matrix $X_{\mathrm{SI,p}} = \begin{bmatrix} x(N_{\mathrm{p}}) & \cdots & x(T-N_{\mathrm{f}}+1) \end{bmatrix}$ collects the system states, while $D_{\mathrm{SI,f}}$ has the same structure as $Y_{\mathrm{SI,f}}$ but is constructed from the disturbance trajectory. Both $X_{\mathrm{SI,p}}$ and $D_{\mathrm{SI,f}}$ are unknown.

The key idea of the subspace identification method is to project the future outputs $Y_{\mathrm{SI,f}}$ onto the row space of the past data $[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top$ along the future inputs $U_{\mathrm{SI,f}}$. To achieve this, the concept of the *oblique projection* plays a central role. Specifically, given matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, the oblique projection of $\mathbf{A}$ onto the row space of $\mathbf{B}$ along the row space of $\mathbf{C}$ is defined as $\mathbf{A}\Pi_{\mathbf{B}|\mathbf{C}}$, where

$$\Pi_{\mathbf{B}|\mathbf{C}} = \mathbf{B}^\top \left(\mathbf{B}\Pi_{\mathbf{C}}^\perp \mathbf{B}^\top\right)^\dagger \mathbf{B}\Pi_{\mathbf{C}}^\perp \quad \text{and} \quad \Pi_{\mathbf{C}}^\perp = I - \mathbf{C}^\top \left(\mathbf{C}\mathbf{C}^\top\right)^\dagger \mathbf{C}.$$

With this definition, we obtain the following result:

$$\begin{aligned} \Omega &:= Y_{\mathrm{SI,f}} \Pi_{[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top | U_{\mathrm{SI,f}}} \\ &= \left(\Gamma X_{\mathrm{SI,p}} + \Theta U_{\mathrm{SI,f}} + D_{\mathrm{SI,f}}\right) \Pi_{[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top | U_{\mathrm{SI,f}}} \\ &= \Gamma X_{\mathrm{SI,p}} \Pi_{[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top | U_{\mathrm{SI,f}}} + \Theta U_{\mathrm{SI,f}} \Pi_{[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top | U_{\mathrm{SI,f}}} + D_{\mathrm{SI,f}} \Pi_{[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top | U_{\mathrm{SI,f}}} \\ &\approx \Gamma X_{\mathrm{SI,p}}, \end{aligned}$$

in which the relation $U_{\mathrm{SI,f}} \Pi_{[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top | U_{\mathrm{SI,f}}} = 0$ follows directly from the definition of the oblique projection. The approximation $D_{\mathrm{SI,f}} \Pi_{[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top | U_{\mathrm{SI,f}}} \approx 0$ holds since the disturbance is assumed to be white and uncorrelated with both past inputs and outputs. Moreover, $X_{\mathrm{SI,p}} \Pi_{[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top | U_{\mathrm{SI,f}}} \approx X_{\mathrm{SI,p}}$ because the row space of $X_{\mathrm{SI,p}}$ is contained in the row space of $[U_{\mathrm{SI,p}}^\top \; Y_{\mathrm{SI,p}}^\top]^\top$. Then, applying the SVD to the resulting matrix Ω yields $\Omega = \mathcal{U}\Sigma\mathcal{V}^\top$ with

$$\mathcal{U} = \begin{bmatrix} \mathcal{U}_1 & \mathcal{U}_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \quad \text{and} \quad \mathcal{V} = \begin{bmatrix} \mathcal{V}_1 & \mathcal{V}_2 \end{bmatrix},$$

in which $\mathcal{U}_1 \in \mathbb{R}^{N_f \times n}$, $\Sigma_1 \in \mathbb{R}^{n \times n}$ and $\mathcal{V}_1 \in \mathbb{R}^{(T-N_p-N_f+2)\times n}$. The function `n4sid` uses the following canonical way to estimate Γ and $X_{SI,p}$:

$$\tilde{\Gamma} := \mathcal{U}_1 \Sigma_1^{1/2} \quad \text{and} \quad \tilde{X}_{SI,p} := \Sigma_1^{1/2} \mathcal{V}_1^\top.$$

The structure in (32) implies that the matrix C can be estimated from the first row of $\tilde{\Gamma}$, denoted by $\tilde{C}$. Moreover, (32) implies the relation

$$\underbrace{\begin{bmatrix} C \\ \vdots \\ CA^{N_f-2} \end{bmatrix}}_{\Gamma_1} A = \underbrace{\begin{bmatrix} CA \\ \vdots \\ CA^{N_f-1} \end{bmatrix}}_{\Gamma_2}.$$

Hence, the matrix A can be estimated as $\tilde{A} := \tilde{\Gamma}_1^\dagger \tilde{\Gamma}_2$, where $\tilde{\Gamma}_1$ is obtained from $\tilde{\Gamma}$ by removing its last row, and $\tilde{\Gamma}_2$ by removing its first row. Finally, the matrix B is estimated via the least-squares problem:

$$\tilde{B} := \arg\min_B \left\| (\tilde{X}_{SI,p})^+ - \tilde{A}\tilde{X}_{SI,p} - BU_{SI,p} \right\|,$$

which has the closed-form solution:

$$\tilde{B} = (U_{SI,p} U_{SI,p}^\top)^\dagger U_{SI,p} (\tilde{X}_{SI,p}^+ - \tilde{A}\tilde{X}_{SI,p})^\top.$$

In this way, the estimated matrices $(\tilde{A}, \tilde{B}, \tilde{C})$ are used to construct the data-driven representation $\tilde{G}_{SI}$. Clearly, the subspace identification method has certain limitations. As discussed above, it requires the disturbance to be white and uncorrelated, making it unsuitable for cases with, e.g., structured disturbances. A detailed comparison is provided in the following subsections.

5.2. Toy example

In this part, the performance of different methods are evaluated on the system:

$$H(z) = \frac{0.7836z^3 + 0.7732z^2 + 0.1936z + 0.009937}{z^4 + 1.778z^3 + 0.9869z^2 + 0.2007z + 0.0205},$$

which is controllable, minimum-phase and of relative degree 1. Each element of the invariant reference signal and initial state x_0 are randomly generated from the interval $[-10, 10]$. The box-constraint set $\mathcal{Q}$ for each input element is selected from $[-20, 20]$, with a guarantee that the optimal input trajectory $u^* = G^{-1}(r - c)$ is included (G and c are defined in (4)), which leads to $J(u^*) = 0$. The proposed method has two parts: data-driven representation (offline part) and iterative learning control (online part). The offline data length is $T = 100$ and the initial trajectory length $T_{ini} = 4$, which is equal to the system lag. The trial length is $N = 20$, the maximal iteration number is $M = 500$ and the time window length is $W = 20$ for the online part.

Three kinds of output disturbances are considered: (a), the uniformly distributed disturbance with the bound $\bar{d}$. (b), the sine disturbance with the bound $\bar{d}$, 100rad/s angular frequency and random phases. (c), the white Gaussian disturbance with 0 mean and σ standard deviation. The disturbance data length is T for the offline part and N for the online part. The bound is $\bar{d} = 3$ for the first two disturbances, and the standard deviation is $\sigma = 1$ for the last disturbance. Although the white Gaussian disturbance can be unbounded, around 99.73% of samples belong to the interval $[-3\sigma, 3\sigma]$ according to the three-σ rule.

In order to evaluate the offline performance, we discuss the relative error generated by the data-driven representations $\tilde{G}_{DD}$ and $\tilde{G}_{SI}$. In particular, a feasible input trajectory u is randomly generated after the offline part, and then used to calculate the relative error $\|y - \tilde{y}\|_2 / \|y\|_2$ by $y = Gu$ (exact output), and $\tilde{y}_{DD} = \tilde{G}_{DD}u$ or $\tilde{y}_{SI} = \tilde{G}_{SI}u$, in which the initial response is assumed to be 0. As for the online performance, we use the exact tracking error, i.e., ε_j in (20), and the input error $u_j - u^*$ as the indicators.

We first compare the performance of the different methods in the disturbance-free case, i.e., $\bar{d} = 0$. In this case, we have $\tilde{G}_{DD} = \tilde{G}_{SI} = G$ and $\Delta_1 = \Delta_2 = \Delta_3 = \delta = 0$. Therefore, the relative errors of the two data-driven representations are 0. Applying such $\tilde{G}_{DD}$ and $\tilde{G}_{SI}$ to the online control, the tracking error $\|\varepsilon_j\|_2$ is depicted in Fig. 1, which shows that the fast ILC converges much faster than the classical ILC. Note that the hybrid ILC is not used in the disturbance-free case, since there is no divergence term in (29), hence no switching mechanism is required.

In the disturbance case, the relative errors are shown in Table 1, which implies that the direct data-driven method yields larger relative errors compared to the system identification method in the cases of uniform and white Gaussian disturbance. This is to be expected, given that the subspace identification is designed to exploit the full knowledge of the state-space representation behind the data with white Gaussian disturbance [8]. On the other hand, the direct data-driven method shows better performance in the case of structured disturbance, exhibiting much smaller relative errors compared to SI. One possible explanation is that the non-parametric nature of the direct data-driven method is able to better capture the additional system dynamics implicitly responsible for the structured disturbance.

Second, to compare the data-driven ILC with the model-based ILC, we apply G, $\tilde{G}_{DD}$ and $\tilde{G}_{SI}$ to both the classical and fast ILC methods, assuming that the true model G is available. The tracking error $\|\varepsilon_j\|_2$ is shown in Fig. 2, where the maximum number of iterations is limited to $M = 100$ for clarity of presentation. Since the data-driven representations $\tilde{G}_{DD}$ and $\tilde{G}_{SI}$ are approximations of the true model G, the ILC methods using G achieve better performance than those using $\tilde{G}_{DD}$ and $\tilde{G}_{SI}$ under all three types of disturbances. Moreover, the results in Fig. 2 are consistent with Table 1, which shows that the direct data-driven method performs comparably to the model-based method in the presence of structured disturbances.

Table 1
Relative error of two representation methods.

	Uniform	Sine	Gaussian
DD	0.2629	**1.0179e-04**	0.2007
SI	**0.0975**	0.1152	**0.0760**

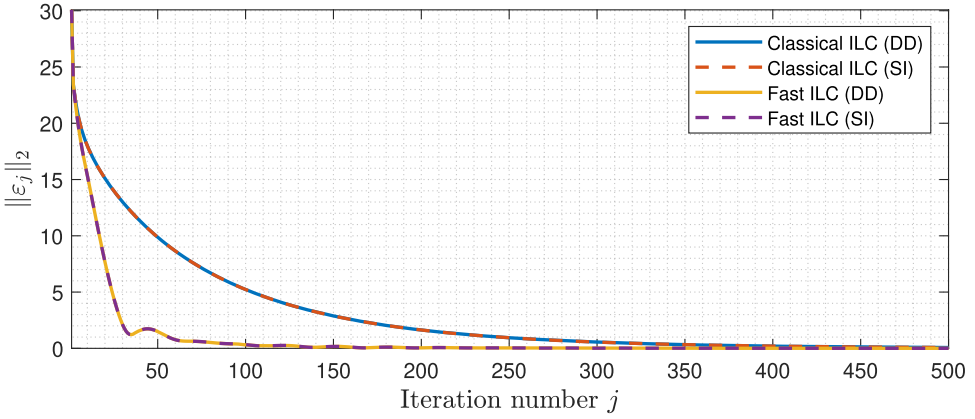
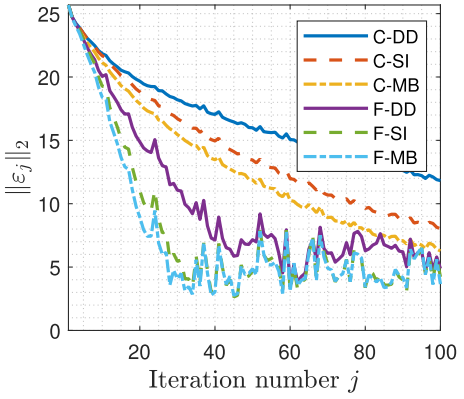
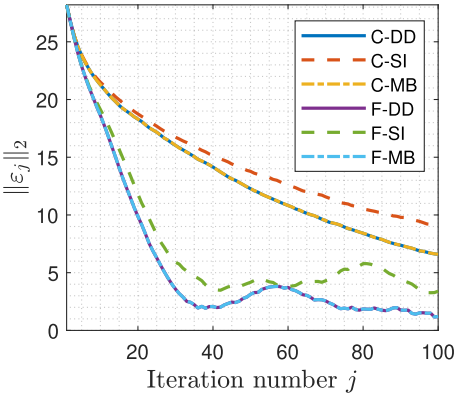


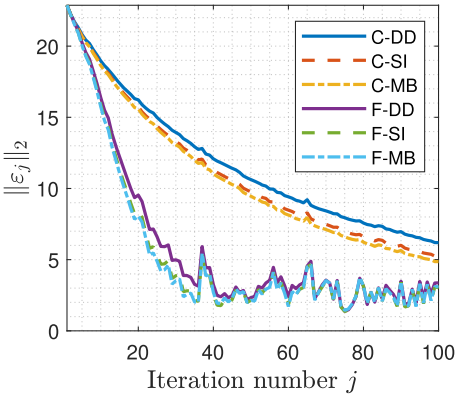
Fig. 1. Tracking error in the disturbance-free case.



(a) Uniformly distributed random disturbance.

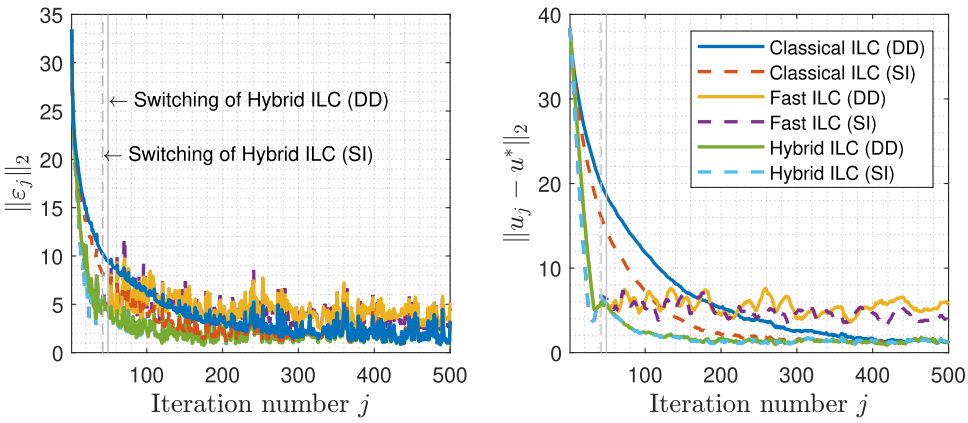


(b) Sine disturbance.

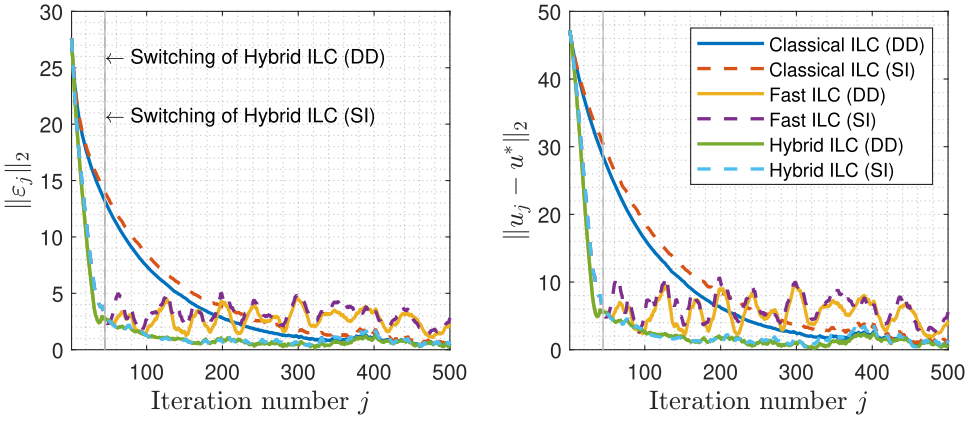


(c) White Gaussian disturbance.

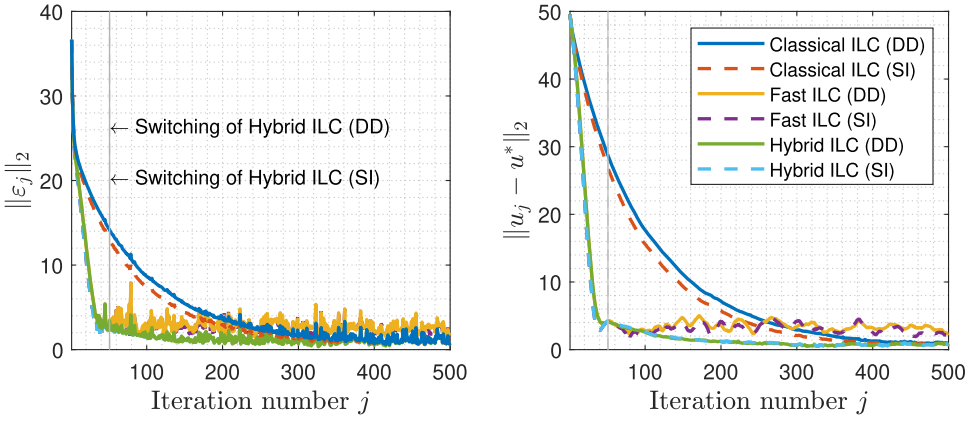
Fig. 2. Compared to the model-based ILC in the disturbance case (C: Classical, F: Fast, MB: Model-Based).



(a) Uniformly distributed random disturbance.



(b) Sine disturbance.



(c) White Gaussian disturbance.

Fig. 3. Tracking and input errors in the disturbance case.

Finally, we apply $\tilde{G}_{\mathrm{DD}}$ and $\tilde{G}_{\mathrm{SI}}$ to three ILC methods: classical, fast and hybrid ILC. The tracking error $\|\varepsilon_j\|_2$ and input error $\|u_j - u^*\|_2$ are depicted in Fig. 3, which indicates that the hybrid ILC has the same fast convergence speed as the fast ILC at the early iteration stage, and has an excellent asymptotic performance after switching. In the other words, the hybrid ILC achieves equally good asymptotic performance as the classical ILC using $100 - 200$ iterations less. However, in the cases of uniform and white Gaussian disturbance, each ILC method with the non-parametric representation cannot outperform the parametric representation. As discussed above, the direct data-driven method has a limitation in handling the unstructured disturbance. On the other hand, as expected, the

ILC method with the non-parametric representation outperforms the parametric representation in the case of sine disturbance because of the high representation accuracy; see Table 1. In this way, the effectiveness of the non-parametric data-driven representation and the hybrid ILC are illustrated.

5.3. Batch of random systems for the structured disturbance

The above toy example shows that the direct data-driven method is promising when it comes to handling structured disturbances. In this subsection, we test the different methods on a batch of randomly generated systems affected by sine disturbances, and discuss the statistical performance. We use 100 random systems of order $n = 4$, generated using the MATLAB function `drss`. All systems are controllable, minimum-phase and of relative degree 1. The disturbance setting, the trial length N, the maximal iteration number M and the time window length W are the same as in the previous subsection. The input box-constraint set $\mathcal{Q}$ and the reference r are randomly generated for each system.

The average relative errors yielded by the non-parametric and parametric representations are 0.0277 and 0.0681, which implies that the direct data-driven method can obtain smaller relative errors in most of experiments compared to the subspace identification method. Fig. 4(a) shows that three ILC methods equipped with the non-parametric representation are more robust to sine disturbances on average when it has a large bound $\bar{d}$, and then a better control performance can be achieved. On the other hand, noting that the early iteration stage of an ILC process can reflect the convergence speed and the final stage can reflect the steady-state tracking error, we discuss the performance of different ILC methods at these two stages in the following. We select the average tracking errors at the two stages as indicators and the stage data length as 50, then the early stage performance indicator is $\frac{1}{50}\sum_{j=1}^{50}\|\epsilon_j\|_2$ and the final stage performance indicator is $\frac{1}{50}\sum_{j=451}^{500}\|\epsilon_j\|_2$. A smaller indicator value implies a better performance: faster convergence speed or smaller steady-state tracking error. The box plots that counted all the random experiments are given as Fig. 4(b). The left box plot shows that the hybrid ILC has better statistical performance at the early stage. Compared to the parametric data-driven representation, the ILC methods with non-parametric representation yield smaller errors, which confirm that the non-parametric representation is more robust to the large structured disturbance. Since the fast ILC is sensitive to disturbances (tracking errors may oscillate significantly, see Fig. 3(b)), the 0th, 25th and 50th percentiles of the fast ILC are larger than the classical ILC. This drawback is improved by the hybrid ILC. The right box plot in Fig. 4(b) implies that smaller tracking errors at the final stage can be obtained by the hybrid ILC, the reason is that the hybrid ILC achieves a similar asymptotic performance to the classical ILC in advance and then the tracking error can be reduced further. It is worth noting that the tracking error may be decreasing very slowly if only using the classical ILC (the 75th and 100th percentiles can be even larger than the fast ILC), which implies the necessity of a switching mechanism.

The algorithm performance on a small disturbance bound is also discussed. We choose $\bar{d} = 0.5$, the non-parametric and parametric representations yield similar average relative errors as 0.0046 and 0.0061, respectively. Fig. 5(a) implies that each ILC method can provide a similar performance by using the two data-driven representations. Fig. 5(b) also shows the similarity of the two representations in the small structured disturbance case. Furthermore, the advantages of the hybrid ILC are indicated continually. Combined the result in Figs. 4 and 5, we conclude that the non-parametric representation is more robust to the large structured disturbance on average and performs similarly to the parametric representation with the small structured disturbance. We also suggest that the hybrid ILC should always be used, which can achieve a faster tracking error decrease at the early stage and obtain a better asymptotic performance after the switching.

Remark 3. The discussion in this subsection is based on the statistical performance, there also exist some systems that the non-parametric representation cannot outperform the parametric one. The system structure under which direct data-driven method can provide a superior performance with the structured disturbance will be studied in the future work.

Remark 4. The comparison between the non-parametric and parametric representations is based on the same order n, i.e., using n in (16). The essence of structured disturbances is an additional system dynamics of order m, hence, using the order $n + m$ in the function `n4sid` can perform equally well if not better than the direct data-driven method. In practice, however, the additional order m is usually unknown, but the system order n is known a priori, for example the robotic motion system in the next subsection. Therefore, the direct data-driven method can be more practical in some applications.

5.4. Simulation on the robotic motion system

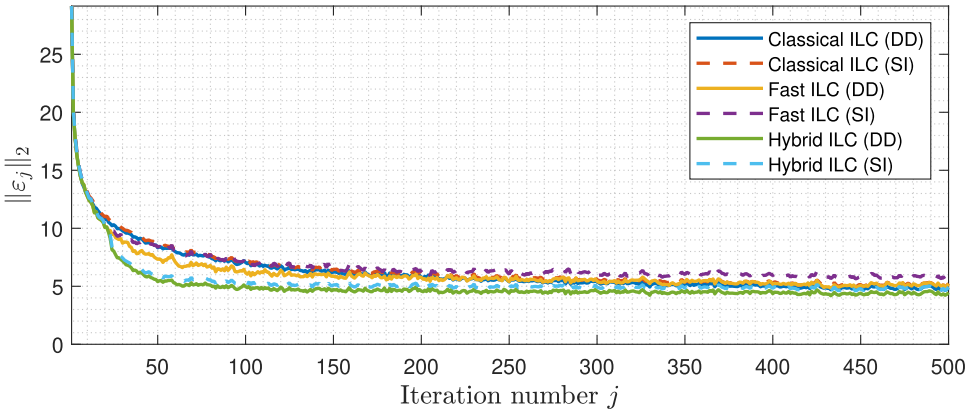
In this subsection, we provide a case study on a high-precision two-axis robotic motion system; see [34] for details. Since the X and Y axes are decoupled, we only consider the Y-axis for simplicity and the feedforward control of the X-axis can be deployed in a similar way. The Y-axis model is

$$G_Y(s) = \frac{3500\left(\frac{1}{88.5^2}s^2 + \frac{0.1}{88.5}s + 1\right)}{s^2\left(\frac{1}{89.5^2}s^2 + \frac{0.1}{89.5}s + 1\right)} \quad \left(\frac{\text{mm}}{\text{V}}\right)$$

and the feedback controller is

$$C_Y(s) = \frac{10(s + 1.9)(s + 100)}{(s + 1.15)(s + 200)} \quad \left(\frac{\text{V}}{\text{mm}}\right),$$

in which the tracking error dynamics is stabilized with a feedback controller $C_Y(s)$ and the ILC is used to generate the feedforward input. The control structure is shown in Fig. 6.



(a) Average tracking errors of 100 random experiments.

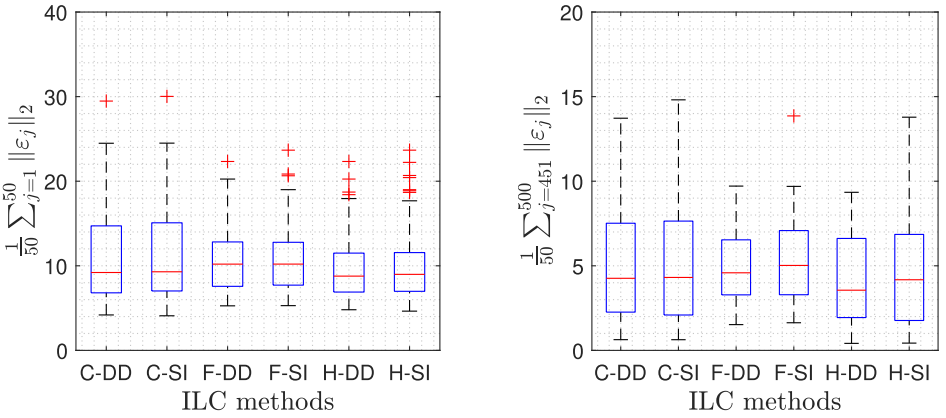


Fig. 4. Performance of different ILC methods with the sine disturbance in the case of $\bar{d} = 3$.

The transfer function from reference r and feedforward u to position y is

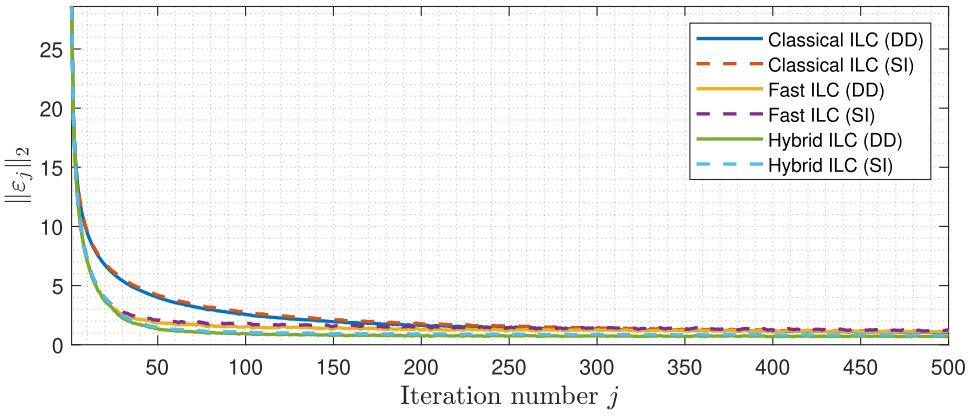
$$y(s) = \frac{G_Y}{1 + G_Y C_Y} u(s) + \frac{G_Y C_Y}{1 + G_Y C_Y} r(s),$$

in which the sampling rate is set to be 100Hz. For the offline part, the data trajectories with the length $T = 100$ are collected. The length of the initial trajectory is $T_{\text{ini}} = 12$, which is equal to the lag of the closed-loop system. The data length in one trial is $N = 10$. The box-constraint on $u_j(k)$ is 0V $\sim$ 10V. To start the iteration, each element in the initial trajectory u_0 is selected as the center of the box-constraint, i.e., $u_0(k) = 5$ for $k \in \{0 \dots N-1\}$. Each element of the invariant reference r is randomly generated from 0.77mm $\sim$ 0.81mm. The initial condition is randomly generated and invariant in each trial. The maximal iteration number is fixed as $M = 20$ and the time window length is $W = 5$.

Seven methods are compared: the classical, fast and hybrid ILC equipped with the non-parametric and parametric representations, and the feedback controller C_Y (without the feedforward). In the disturbance-free case, the tracking error $\|\varepsilon_j\|_2$ is shown in Fig. 7. Except for the similar conclusion to the above subsections, we also observe that a better tracking performance is obtained by the ILC methods, compared to only using the feedback controller, which illustrates the necessity of the feedforward in real applications.

In the disturbance case, since the required control accuracy is high, we choose the disturbance level as $\bar{d} = 0.1$ and the standard deviation of white Gaussian disturbance as $\sigma = 0.1$. The performance is illustrated by Fig. 8. The left column of Fig. 8 is the tracking errors under different disturbances. As expected for the structured disturbance, the non-parametric and parametric representations perform similarly since the disturbance bound is small. In the cases of the uniform and white Gaussian disturbance, the parametric representation performs better than the non-parametric one. Moreover, the hybrid ILC performs similarly with the fast ILC because of the minor disturbance, which leads to a small δ and benefits the convergence from the first term in (29). This similarity also implies that the hybrid ILC is not worse than the classical and fast ILC under the small disturbance. In fact, the performance improvement introduced by the hybrid ILC is almost a free lunch, the hybrid ILC requires the calculation, comparison and storage of the averaged tracking error norm, and these operations are cheap in the modern digital controller. The right column of Fig. 8 shows the control convergence of the hybrid ILC with the non-parametric representation: u_j converges to u^* (dotted line) with the increase of j. In this way, Fig. 8 confirms the conclusions of previous subsections and the effectiveness of the proposed method is verified.

Remark 5. The choice of the maximal iteration number M depends on the specific application. For example, the batch reactor in [7] uses $M = 300$, the robotic testbed system in [35] uses $M = 100$ and the nanopositioning system in [36] uses $M = 60$. Thus, the value



(a) Average tracking errors of 100 random experiments.

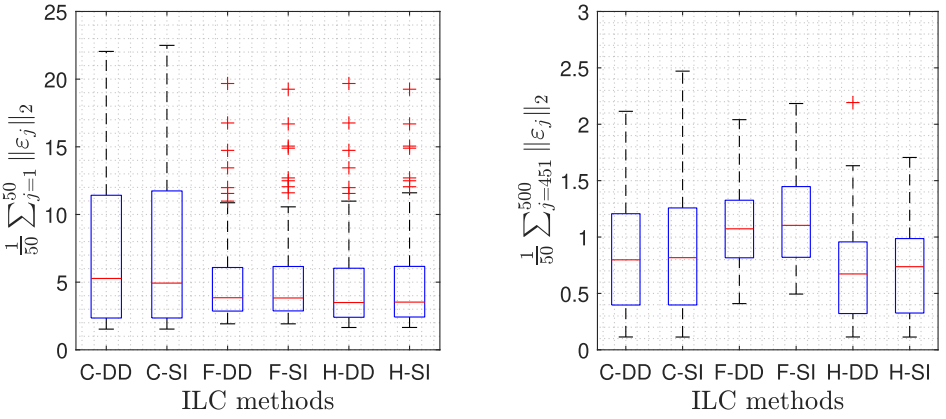


Fig. 5. Performance of different ILC methods with the sine disturbance in the case of $\bar{d} = 0.5$.

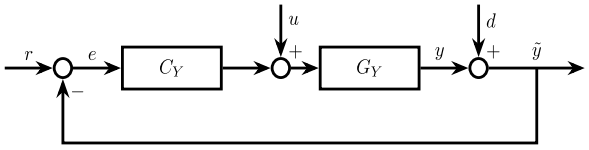


Fig. 6. Control structure of the precision motion system.

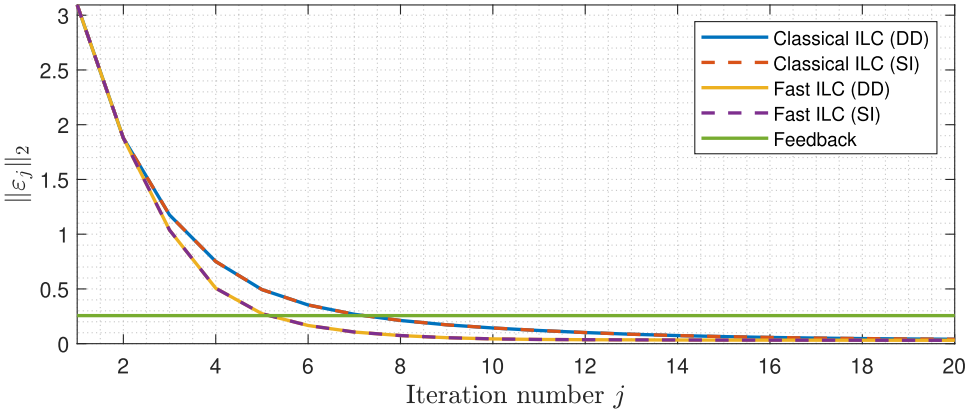
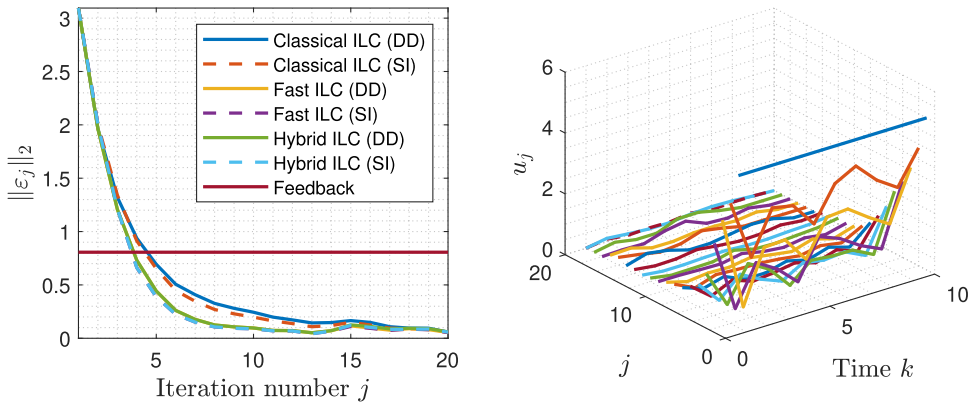
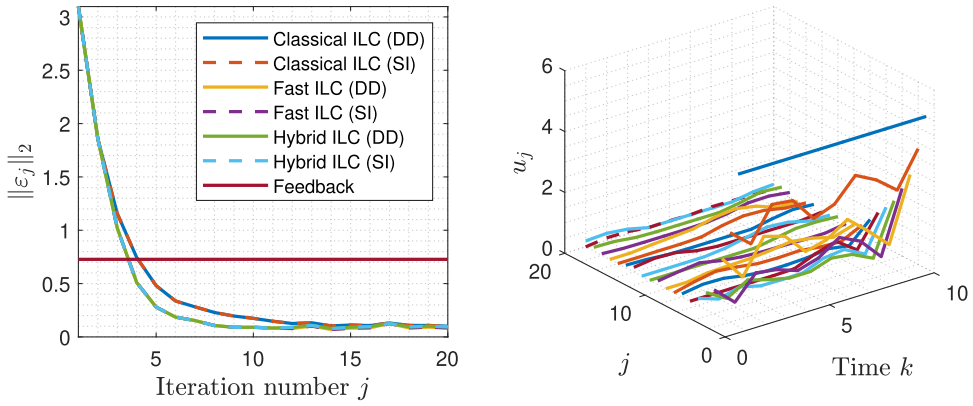


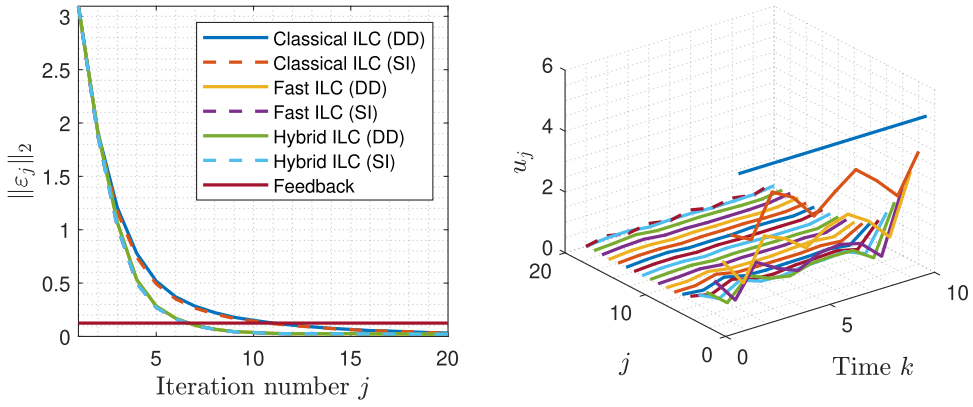
Fig. 7. Tracking error in the disturbance-free case.



(a) Uniformly distributed random disturbance.



(b) Sine disturbance.



(c) White Gaussian disturbance.

Fig. 8. Tracking error and control convergence in the disturbance case.

of M is typically determined by practical requirements. On the other hand, the time window length W in Algorithm 3 should be chosen with respect to M. If M is small, the available data for computing the average tracking error norm $\bar{e}_j$ is limited, so W should also be kept small. Conversely, if M is large or the system dynamics are complex, it is reasonable to apply the fast ILC over a longer window, i.e., W should be correspondingly larger.

6. Conclusion

This work presents a fast data-driven ILC method for linear systems with unknown dynamics under the presence of bounded output disturbances and input box-constraints. A method based on the non-parametric data-driven representation of the system dynamics is

compared to a subspace identification method. The non-parametric data-driven method shows a promise in dealing with structured disturbances. On the other hand, since the data-driven representation and the online measurement are inexact, we design the ILC based on the inexact classical and fast gradient methods with proven convergence results. Moreover, we propose a hybrid ILC method which has a faster convergence speed and a better asymptotic performance. Currently, only output disturbances are considered in this paper, while model uncertainty (disturbance in the state equation of (1)) will be addressed in future work. In addition, since the data-driven representation remains fixed after the offline phase, updating it using online data constitutes another direction for future research. Furthermore, motivated by the discussion in Remark 5, the development of an adaptive strategy for tuning W will also be investigated in the future.

CRediT authorship contribution statement

Jia Wang: Writing – original draft, Methodology; **Leander Hemelhof:** Writing – original draft, Methodology; **Ivan Markovsky:** Supervision, Methodology; **Panagiotis Patrinos:** Supervision, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Ivan Markovsky is an ICREA professor at the International Centre for Numerical Methods in Engineering, Barcelona. The research leading to these results has received funding from: Fond for Scientific Research Vlaanderen (FWO) projects G033822N, G081222N, G0A0920N. European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 953348.

References

[1] R. De Rozario, T. Oomen, Data-driven iterative inversion-based control: achieving robustness through nonlinear learning, Automatica 107 (2019) 342–352.
[2] S. Tien, Q. Zou, S. Devasia, Iterative control of dynamics-coupling-caused errors in piezoscanners during high-speed AFM operation, IEEE Trans. Control Syst. Technol. 13 (6) (2005) 921–931.
[3] R. Chi, Z. Hou, S. Jin, B. Huang, Computationally efficient data-driven higher order optimal iterative learning control, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 5971–5980.
[4] A. Tayebi, Adaptive iterative learning control for robot manipulators, Automatica 7 (2004) 1195–1203.
[5] K.H. Park, A study on the robustness of a PID-type iterative learning controller against initial state error, Int. J. Syst. Sci. 30 (1) (1999) 49–59.
[6] K.L. Barton, A.G. Alleyne, A norm optimal approach to time-varying ILC with application to a multi-axis robotic testbed, IEEE Trans. Control Syst. Technol. 19 (1) (2011) 166–180.
[7] R. Chi, Z. Hou, B. Huang, S. Jin, A unified data-driven design framework of optimality-based generalized iterative learning control, Comput. Chem. Eng. 77 (2015) 10–23.
[8] P. Van Overschee, B. De Moor, Subspace Identification for Linear Systems: Theory, Implementation, Applications, Kluwer Academic Publishers Group, The Netherlanders, 2012.
[9] G. Nijsse, M. Verhaegen, N.J. Doelman, A new subspace based approach to iterative learning control, in: 2001 European Control Conference (ECC), Porto, Portugal, 2001, pp. 3375–3380.
[10] A. Janczak, E. Rogers, Z. Cai, Subspace identification of process dynamics for iterative learning control, in: Proceedings of the 8th International Workshop on Multidimensional Systems, Erlangen, Germany, 2013, pp. 39–44.
[11] I. Markovsky, T. Liu, A. Takeda, Data-driven structured noise filtering via common dynamics estimation, IEEE Trans. Signal Process. 68 (2020) 3064–3073.
[12] M. Bodson, J.S. Jensen, S.C. Douglas, Active noise control for periodic disturbances, IEEE Trans. Control Syst. Technol. 9 (1) (2001) 200–205.
[13] J.C. Willems, P. Rapisarda, I. Markovsky, B. De Moor, A note on persistency of excitation, Syst. Control Lett. 54 (4) (2005) 325-329.
[14] I. Markovsky, P. Rapisarda, Data-driven simulation and control, Int. J. Control 81 (12) (2008) 1946–1959.
[15] I. Markovsky, F. Dörfler, Behavioral systems theory in data-driven analysis, signal processing, and control, Annu. Rev. Control 52 (2021) 42–64.
[16] E. Elokda, J. Coulson, P.N. Beuchat, J. Lygeros, F. Dörfler, Data-enabled predictive control for quadcopters, Int. J. Robust Nonlinear Control 31 (2021) 8916–8936.
[17] P.G. Carlet, A. Favato, S. Bolognani, F. Dörfler, Data-driven predictive current control for synchronous motor drives, in: 2020 IEEE Energy Conversion Congress and Exposition (ECCE), Detroit, MI, USA, 2020, pp. 213–218.
[18] Z. Jiang, B. Chu, Norm optimal iterative learning control: a data-driven approach, IFAC-PapersOnLine 55 (12) (2022) 482–487.
[19] Z. Jiang, B. Chen, B. Chu, Data-driven norm optimal iterative learning control for point-to-point tasks, IFAC-PapersOnLine 56 (2) (2023) 1051–1056.
[20] C. Wang, D. Meng, Data-based optimization control for learning systems, IEEE Trans. Circuits Syst. II Express Briefs 70 (7) (2023) 2560–2564.
[21] T.J. Harte, J.J. Hätönen, D.H. Owens, Discrete-time inverse model-based iterative learning control: stability, monotonicity and robustness, Int. J. Control 78 (8) (2005) 577–586.
[22] J.D. Ratcliffe, J.J. Hätönen, P.L. Lewin, E. Rogers, D.H. Owens, Robustness analysis of an adjoint optimal iterative learning controller with experimental verification, Int. J. Robust Nonlinear Control 18 (2008) 1089–1113.
[23] B. Chu, D.H. Owens, Accelerated norm-optimal iterative learning control algorithms using successive projection, Int. J. Control 82 (8) (2009) 1469–1484.
[24] Y. Geng, X. Ruan, Quasi-Newton-type optimized iterative learning control for discrete linear time invariant systems, Control Theory Technol. 13 (2015) 256–265.
[25] Y. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, Dokl. Akad. Nauk 27 (2) (1983) 372–376.
[26] P. Gu, S. Tian, Y.Q. Chen, Iterative learning control based on nesterov accelerated gradient method, IEEE Access 7 (2019) 115836–115842.
[27] O. Devolder, F. Glineur, Y. Nesterov, First-order methods of smooth convex optimization with inexact oracle, Math. Program. 146 (2014) 37–75.
[28] Y. Nesterov, Smooth minimization of non-smooth functions, Math. Program. 103 (2005) 127–152.
[29] J. Coulson, J. Lygeros, F. Dörfler, Data-enabled predictive control: in the shallows of the DeePC, in: 2019 18th European Control Conference (ECC), Naples, Italy, 2019, pp. 307–312.
[30] I. Markovsky, F. Dörfler, Identifiability in the behavioral setting, IEEE Trans. Automat. Control 68 (3) (2023) 1667–1677.
[31] J.W. Demmel, The smallest perturbation of a submatrix which lowers the rank and constrained total least squares problems, SIAM J. Numer. Anal. 24 (1) (1987) 199–206.

[32] P. Van Overschee, B. De Moor, N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems, Automatica 30 (1) (1994) 75–93.
[33] S.J. Qin, An overview of subspace identification, Comput. Chem. Eng. 30 (2006) 1502–1513.
[34] D.A. Bristow, A.G. Alleyne, A high precision motion control system with application to microscale robotic deposition, IEEE Trans. Control Syst. Technol. 14 (6) (2006) 1008–1020.
[35] K.L. Barton, A.G. Alleyne, A cross-coupled iterative learning control design for precision motion control, IEEE Trans. Control Syst. Technol. 16 (6) (2008) 1218–1231.
[36] G. Parmar, K. Barton, S. Awtar, Large dynamic range nanopositioning using iterative learning control, Precis. Eng. 38 (1) (2014) 48–56.