Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 1051–1056

Data-driven Norm Optimal Iterative Learning Control for Point-to-Point Tasks

Zheng Jiang * **Bin Chen** * **Bing Chu** *

* *School of Electronics and Computer Sciences, University of Southampton, Southampton, United Kingdom, (e-mail: {zj1y19, bc1n16, b.chu}@ soton.ac.uk)*

Abstract: Iterative learning control (ILC) is suitable for high-performance repetitive tasks since it learns from past trials to improve the tracking performance. Existing ILC designs often require a model, which can be difficult or expensive to obtain in practice. To address this problem, we recently developed a data-driven norm optimal ILC using the latest developments from data-driven control, namely, the Willems' fundamental lemma. In this paper, we show that the idea can also be extended to point-to-point ILC tasks that focus on tracking some intermediate points of the whole trial. We propose a novel data-driven point-to-point norm optimal ILC algorithm that can achieve the same performance as the model-based algorithm but without using an analytical model. The design requires the available data to be persistently exciting of a sufficiently high order. To relax this requirement, a receding horizon based algorithm and a trial partition based algorithm are further developed with well-defined, but different convergence properties. Numerical examples are given to illustrate the proposed algorithms' effectiveness.

Keywords: Iterative learning control, data-driven control, point-to-point tasks, linear systems.

1. INTRODUCTION

High-performance tracking control tasks, which require the system to work repetitively to track a desired reference trajectory with high precision, have found wide applications. The recent design applies iterative learning control (ILC) to achieve high-performance tracking control tasks since ILC can update the input by learning from its past trials' input and error information (that implicitly contains the model information) and hence it does not require a highly accurate model that can be difficult or expensive to obtain in practice (Bristow et al., 2006). Till now, it has found a great number of practical applications, e.g., robotics (Armstrong et al., 2021), stroke rehabilitation (Freeman et al., 2012) and additive manufacturing (Lim et al., 2017).

Existing ILC methods can be divided into model-free and model-based designs. Model-free ILC algorithms do not need a system model in the design, but parameters are required to be tuned to guarantee convergence, e.g., proportional-integral-derivative type ILC (Arimoto et al., 1984) and adaptive ILC (Tayebi, 2004). Model-based ILC algorithms, for example, inverse-based ILC (Harte et al., 2005), gradient-based ILC (Owens et al., 2009) and norm optimal ILC (NOILC) (Amann et al., 1996), generally have better tracking performance than model-free methods, but a system model which is not always easy to obtain in practice is required. Comprehensive reviews of ILC can be found in Bristow et al. (2006) and Owens (2016).

There is research on removing/relaxing the system model requirement in ILC design while achieving the convergence performance as that in model-based ILC designs. For example, Janssens et al. (2012) develops an algorithm to estimate Markov parameters by the input/output data, then model-based algorithms can be applied. In Bolder et al. (2018), online experiments are performed to obtain the response of the adjoint of the system. In de Rozario and Oomen (2019), a data-driven iterative inversion based control design in the frequency domain is proposed. Chi et al. (2019) estimates a linearised system model iteratively by an adaptive law. However, existing data-driven ILC algorithms need to do further analysis or experiments to explicitly or implicitly estimate the system model, which can be non-trivial and/or expensive in practice.

Recently, we developed a novel data-driven NOILC framework (Jiang and Chu, 2022) based on some latest developments in data-driven control, namely, the Willems' fundamental lemma (Willems et al., 2005), which directly uses the existing data to design the control input and hence avoids the use or the estimation of the system model. The resulting data-driven algorithms show the same convergence performance as the model-based NOILC algorithm, i.e., monotonic convergence of the tracking error norm to zero, which is appealing in practice.

Point-to-point tasks have wide applications in practice. They focus on the tracking behaviour on the intermediate time instants. For example, the 'pick' and 'place' tasks of a gantry robot only concern the perfect tracking at the points of 'pick' and 'place', other points are not that interesting. This feature creates more freedom in input choices but raises technical challenges for the design, as there is only error data from points of interest, rather than error data from an entire trajectory in conventional ILC design. In this paper, we further extend the data-driven idea in Jiang and Chu (2022) and design a data-driven point-to-point NOILC framework. We show that the proposed algorithm has the same tracking performance

10.1016/j.ifacol.2023.10.1703

as the model-based algorithm without using a system model. The design requires the existing data to be persistently exciting of a sufficiently high order which may not be easily satisfied in practice. Hence, two algorithms, i.e., receding horizon based implementation and the trial partition based implementation are proposed to relax the data assumption. Convergence properties of the proposed algorithms are analysed in detail and numerical examples are presented to illustrate their performance.

The paper is organised as follows. The system dynamics and the point-to-point NOILC are introduced in Section 2. The data-driven point-to-point NOILC framework is developed in Section 3. Two further extensions are proposed in Sections 4 and 5, respectively. Simulation examples are given in Section 6, and Section 7 summarises this paper.

Remark 1. The data-driven design approach used in this paper (i.e., that from Jiang and Chu (2022)) is very general. It can also be applied to other ILC design problems, e.g., for networked dynamical systems (Chen et al., 2023).

2. PROBLEM FORMULATION

This section will describe the system dynamics and formulate the point-to-point ILC problem.

2.1 System Dynamics

Consider the following discrete-time single-input single-output, linear time-invariant system in state-space form

$$\begin{aligned} x_k(t+1) &= Ax_k(t) + Bu_k(t), \quad x_k(0) = x_0, \\ y_k(t) &= Cx_k(t), \end{aligned} \tag{1}$$

where k is the trial index; $t \in [0, N]$ is time index in which N denotes the trial length; $u_k(t) \in \mathbb{R}$, $y_k(t) \in \mathbb{R}$ and $x_k(t) \in \mathbb{R}^n$ are the system input, output and state (where n is system order); A, B and C are system matrices with proper dimensions. For simplicity, we assume the system (1) is both controllable and observable. The system (1) is required to work within the time interval $[0, N]$ repetitively. At the time $N+1$, the time t is reset to 0 and the system state is reset to the identical initial condition $x_k(0) = x_0$, $\forall k \in \mathbb{N}$, then the next trial starts.

Assume the relative degree is unity, i.e. $CB \neq 0$, the lifted form system input and output are defined as (Owens, 2016)

$$\begin{aligned} u_k &= [u_k(0), u_k(1), \cdots, u_k(N-1)]^T, \\ y_k &= [y_k(1), y_k(2), \cdots, y_k(N)]^T. \end{aligned} \tag{2}$$

The system dynamics can then be represented as

$$y_k = Gu_k + d, \tag{3}$$

where G is the system matrix defined as

$$G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{bmatrix}, \tag{4}$$

and d is the system initial response given by

$$d = \left[CAx_0 \ CA^2x_0 \ CA^3x_0 \ \cdots \ CA^Nx_0\right]^T. \tag{5}$$

For traditional tracking problems, the system is required to track a reference defined at every time instants over the whole trial. By contrast, point-to-point tracking problems focus on M intermediate points of the whole trial, i.e., the intermediate points (or tracking points) at time t_1 to t_M are required to be tracked. Define the point-to-point reference, output and tracking error as

$$\begin{aligned} r^P &= [r(t_1) \ r(t_2) \ \cdots \ r(t_M)]^T, \\ y_k^P &= [y_k(t_1) \ y_k(t_2) \ \cdots \ y_k(t_M)]^T, \ e_k^P = r^P - y_k^P. \end{aligned} \tag{6}$$

Note the point-to-point input-output relationship is

$$y_k^P = FGu_k + Fd, \tag{7}$$

where F is defined as

$$F = \left[f_{t_1}^T \quad f_{t_2}^T \quad \cdots \quad f_{t_M}^T\right]^T \in \mathbb{R}^{M \times N}, \tag{8}$$

in which $f_{t_i} \in \mathbb{R}^N$ $(1 \leq i \leq M)$ is the standard basis in $\mathbb{R}^N$ where the t_i-th element in f_{t_i} is 1 and the rest are 0. Without loss of generality, r is replaced by $r - d$, such that $d = 0$. Equations (3) and (7) becomes

$$y_k = Gu_k \quad \& \quad y_k^P = FGu_k. \tag{9}$$

Then the point-to-point ILC design problem can be stated as finding a control updating law

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

such that the point-to-point error converges to 0, i.e.,

$$\lim_{k \to \infty} e_k^P = 0. \tag{11}$$

2.2 Point-to-point Norm Optimal ILC

The point-to-point NOILC was proposed in Owens et al. (2013) that updates the input at each trial by solving

$$\begin{aligned} u_{k+1} &= \arg\min_{u_{k+1}} \left\|e_{k+1}^P\right\|_Q^2 + \|u_{k+1} - u_k\|_R^2, \\ &\text{s.t.} \quad e_{k+1}^P = r^P - y_{k+1}^P, \ y_{k+1}^P = FGu_{k+1}. \end{aligned} \tag{12}$$

where the associated norms are defined as

$$\left\|e_{k+1}^P\right\|_Q = \sqrt{{e_{k+1}^P}^T Q e_{k+1}^P},$$

$$\|u_{k+1} - u_k\|_R = \sqrt{(u_{k+1} - u_k)^T R (u_{k+1} - u_k)}$$

in which Q and R are positive definite weighting matrices. The solution of (12) is

$$u_{k+1} = u_k + (G^T F^T Q F G + R)^{-1} G^T F^T Q e_k^P. \tag{13}$$

The algorithm has some very nice properties, including monotonic convergence in the tracking error norm to zero and convergence of input to the minimum energy solution (Owens et al., 2013). However, the system model G is needed in (13) that may not be easy to obtain in practice. In this work, we will propose a novel data-driven framework to remove this model requirement.

3. DATA-DRIVEN POINT-TO-POINT NOILC

In this section, we will develop the data-driven point-to-point NOILC algorithm.

3.1 Preliminary Results on Data-driven Control

We first introduce some key results, i.e., the Willems' fundamental lemma, of the data-driven control. Define the system trajectory w with J sample length as

$$w = \left[u^T \quad y^T\right]^T = \text{col}(u, y) \in \mathbb{R}^{2J}. \tag{14}$$

All trajectories with J sample length generated by (1) form a subspace $\mathcal{G}_J$, which is defined as

$$\mathcal{G}_J := \left\{\left[u^T \quad y^T\right]^T \in \mathbb{R}^{2J} \mid \exists x(t) \in \mathbb{R}^n, \text{ such that } x(t+1) = Ax(t) + Bu(t), \ y(t) = Cx(t)\right\}.$$

Given a signal $l = [l(0)\; l(1)\; \cdots\; l(J-1)]^T \in \mathbb{R}^J$, its Hankel matrix form is given by

$$\mathcal{H}_t(l) = \begin{bmatrix} l(0) & l(1) \cdots & l(J-t) \\ l(1) & l(2) \cdots & l(J-t+1) \\ \vdots & \vdots \ddots & \vdots \\ l(t-1) & l(t) \cdots & l(J-1) \end{bmatrix}. \quad (15)$$

If $\text{rank}(\mathcal{H}_t(l)) = t$, l is persistently exciting of order t. Next, we present the key result of the data-driven control, which is known as the Willems' fundamental lemma.

Theorem 2. (Willems et al., 2005) Consider a controllable system (1), a J sample long trajectory $w_d = \text{col}(u_d, y_d)$ generated by (1). If the input u_d is persistently exciting of order $t+n$, then any t samples long trajectory of (1) can be written as a linear combination of the columns of $\mathcal{H}_t(w_d)$ and any linear combination $\mathcal{H}_t(w_d)g$, where $g \in \mathbb{R}^{J-t+1}$, is a trajectory of $\mathcal{G}_t$, i.e.,

$$\text{col span}(\mathcal{H}_t(w_d)) = \mathcal{G}_t, \quad (16)$$

in which col span$(\cdot)$ denotes the column span of the matrix.

Theorem 2 shows that the system behaviour can be expressed by the existing data that satisfies the persistently exciting assumption. An important application of *Theorem 2* is the data-driven simulation, which can be stated as given a system initial trajectory $w(1:n) = \text{col}(u(1:n), y(1:n))$ and input $u(n+1:n+N)$, to find the system output $y(n+1:n+N)$.

We can define the following matrices and their partitions

$$U = \mathcal{H}_{n+N}(u_d) = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \; Y = \mathcal{H}_{n+N}(y_d) = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}. \quad (17)$$

The blocks U_p, $Y_p \in \mathbb{R}^{n\times(J-n-N+1)}$ (p denotes 'past') are used to calculate the initial conditions while the blocks U_f, $Y_f \in \mathbb{R}^{N\times(J-n-N+1)}$ (f denotes 'future') are used to calculate the system response. By the Willems' fundamental lemma, there exists some g such that

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u(1:n) \\ y(1:n) \\ u(n+1:n+N) \\ y(n+1:n+N) \end{bmatrix}. \quad (18)$$

Then, calculate the solution of

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g = \begin{bmatrix} u(1:n) \\ y(1:n) \\ u(n+1:n+N) \end{bmatrix}, \quad (19)$$

we can get the response $y(n+1:n+N)$ by

$$y(n+1:n+N) = Y_f g. \quad (20)$$

3.2 A Data-driven Point-to-point NOILC Algorithm

With the above definitions and results, we propose the following data-driven point-to-point NOILC algorithm:

Algorithm 1. Given a trajectory of the system (1) $w_d = \text{col}(u_d, y_d)$ where u_d is persistently exciting of order $N+2n$, a reference r^P, weighting matrices Q and R, past input u_k and past error e_k^P. The input u_{k+1} is generated by iteratively solving the following optimisation problem

$$u_{k+1} = \arg\min_{u_{k+1}} \left\|e_{k+1}^P\right\|_Q^2 + \|u_{k+1} - u_k\|_R^2$$
$$\text{s.t.} \quad \begin{bmatrix} U_p^T & Y_p^T & U_f^T & Y_f^T \end{bmatrix}^T g = \begin{bmatrix} \mathbf{0}_{1,n} & \mathbf{0}_{1,n} & u_{k+1}^T & y_{k+1}^T \end{bmatrix}^T$$
$$y_{k+1}^P = F y_{k+1}, \; e_{k+1}^P = r^P - y_{k+1}^P$$

The solution is given by

$$u_{k+1} = u_k + \begin{bmatrix} I \\ \mathbf{0}_{N,M} \end{bmatrix}^T \bar{F} W_0 (W_0^T \bar{F}^T S \bar{F} W_0)^\dagger W_0^T \bar{F}^T S \begin{bmatrix} \mathbf{0}_{N,1} \\ e_k^P \end{bmatrix}$$

where $\bar{F} = \begin{bmatrix} I & F^T \end{bmatrix}^T$, $\mathbf{0}_{m,n}$ denotes an $m \times n$ zero matrix, $\dagger$ denotes the Moore-Penrose pseudo-inverse, weighting matrix $S = \text{diag}(R, Q)$, the zero initial condition response matrix W_0 can be calculated by the following steps

(1) Calculate the solution g of the following equation

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g = \begin{bmatrix} \mathbf{0}_{n,T-N+1} \\ \mathbf{0}_{n,T-N+1} \\ \mathcal{H}_N(u_d) \end{bmatrix}. \quad (21)$$

(2) The result obtained is then used to calculate

$$Y_f g = Y_0. \quad (22)$$

(3) Combine the input Hankel matrix and the above equation to get the initial condition response matrix

$$W_0 = \begin{bmatrix} \mathcal{H}_N(u_d) \\ Y_0 \end{bmatrix}. \quad (23)$$

The derivations of *Algorithm 1* and (21) to (23) use the idea of data-driven linear quadratic tracking that projects the trajectory on the zero initial condition subbehaviour (Refer to Markovsky and Rapisarda (2008) for details).

The above algorithm has the following properties:

Theorem 3. The input and the tracking performance of data-driven point-to-point NOILC are identical to the model-based point-to-point NOILC. Consequently, *Algorithm 1* guarantees monotonic convergence of the tracking error norm to zero, i.e.,

$$\left\|e_{k+1}^P\right\|_Q \le \left\|e_k^P\right\|_Q \;\&\; \lim_{k\to\infty} e_k^P = 0. \quad (24)$$

In addition, *Algorithm 1* converges to solution u_s^* for the following optimisation problem

$$u_s^* = \arg\min_u \left\{\|u - u_0\|_R^2 | r^P = FGu\right\}. \quad (25)$$

Note the minimum input energy is obtained when $u_0 = 0$.

Proof. The proof is omitted here due to space reasons.

Theorem 3 shows that *Algorithm 1* has identical tracking performance to the model-based algorithm without using the system model. However, the existing input u_d is assumed to be persistently exciting of order $N+2n$, which may not easily be satisfied when N is large. In the next section, we will develop a receding horizon based point-to-point NOILC algorithm to relax this assumption.

4. DATA-DRIVEN RECEDING HORIZON POINT-TO-POINT NOILC

This section develops a data-driven receding horizon point-to-point NOILC algorithm, with its properties analysed.

4.1 Algorithm Description

The idea of the data-driven receding horizon point-to-point NOILC algorithm is shown in Fig. 1. Instead of finding the input over the whole interval in one go, the receding horizon design, at each time instant ih ($0 \le i \le p-1$, where p is the number of intervals), finds the input over a much smaller prediction horizon h (or the shrunk

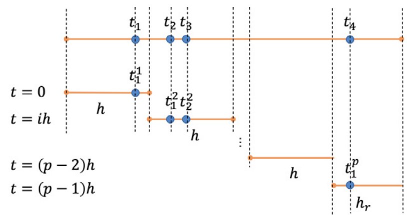

Fig. 1. Illustration of *Algorithm 2*

prediction horizon $h_r = N \bmod h$, where mod denotes the modulo operator, for the last time interval), for which the data only need to be persistently exciting of order $h + 2n$ instead of $N + 2n$ as in the previous setting.

For the data-driven receding horizon point-to-point NOILC algorithm, the interval input and output are defined as

$$u_{k,i} = u_k(ih : ih + h - 1), \quad y_{k,i} = y_k(ih + 1 : ih + h),$$

where i denotes the interval index; h denotes the horizon (or h_r on the last interval horizon to prevent the time $ih + h_r$ exceed the trial length N). The point-to-point interval output is defined as

$$y^P_{k,i} = \left[y_k(ih + t^i_1)\ y_k(ih + t^i_2)\ \cdots\ y_k(ih + t^i_{M_i})\right]^T \in \mathbb{R}^{M_i},$$

in which t^i_j $(1 \le j \le M_i)$ are the intermediate tracking time points falling into the interval (M_i denotes the number of intermediate points in the interval i). Point-to-point reference r^P_i and error $e^P_{k,i}$ are defined in the same manner. If no points are contained, then $y^P_{k,i}$, $e^P_{k,i}$ and r^P_i are set as 0. The relationship between the interval output $y_{k,i}$ and point-to-point interval output $y^P_{k,i}$ is given by

$$y^P_{k,i} = F_i y_{k,i}, \tag{26}$$

where $F_i \in \mathbb{R}^{M_i \times h}$ is defined similarly as in (8). Then, the data-driven receding horizon based point-to-point NOILC algorithm is designed as in *Algorithm 2*.

Algorithm 2. Given a trajectory of the system (1) $w_d = \mathrm{col}(u_d, y_d)$ where u_d is persistently exciting of order $h+2n$, reference r^P, weighting matrices Q and R, previous trial input u_k, previous trial error e^P_k, the input u_{k+1} can be updated by the following steps.

(1) Set $i = 0$.

(2) At time $t = ih$, solve the following problem

$$\begin{aligned} u_{k+1,i} = &\arg \min_{u_{k+1,i}} \left\|e^P_{k+1,i}\right\|^2_Q + \|u_{k+1,i} - u_{k,i}\|^2_R \\ \text{s.t.} &\begin{bmatrix} U_p \\ Y_p \\ U^h_f \\ Y^h_f \end{bmatrix} g = \begin{bmatrix} u^{ini}_{k+1,i} \\ y^{ini}_{k+1,i} \\ u_{k+1,i} \\ y_{k+1,i} \end{bmatrix} \\ & y^P_{k+1,i} = F_i y_{k+1,i},\ e^P_{k+1,i} = r^P_i - y^P_{k+1,i} \end{aligned} \tag{27}$$

where $U^h_f,\ Y^h_f \in \mathbb{R}^{n\times(T-n-h+1)}$ are defined similarly as (17). The interval initial sequences directly use the calculation result of the last interval, which is given by

$$u^{ini}_{k+1,i} = u_{k+1}(ih{-}n : ih{-}1),\ \ y^{ini}_{k+1,i} = y_{k+1}(ih{-}n{+}1 : ih).$$

(3) Apply $u_{k+1,i}$ to the system from time ih to $ih+h-1$.

(4) Set $i \leftarrow i+1$ until the end of the current trial $i = p-1$.

The derivation of *Algorithm 2* uses the similar idea of *Algorithm 1* and the solution of (27) is given by

$$\begin{aligned} u_{k+1,i} = u_{k,i} + &\begin{bmatrix} I \\ \mathbf{0}_{h,M_i} \end{bmatrix}^T \bar{F}_i W_{0h}(W^T_{0h}\bar{F}^T_i S \bar{F}_i W_{0h})^{\dagger} \\ &\times W^T_{0h}\bar{F}_i S \begin{bmatrix} \mathbf{0}_{h,1} \\ e^P_{k,i} + F_i(d_{k,i} - d_{k+1,i}) \end{bmatrix}, \end{aligned} \tag{28}$$

where $\bar{F}_i = \left[I\ F^T_i\right]^T$, W_{0h} is the interval zero initial condition response matrix. The response $d_{k+1,i}$ is calculated by (18) to (20) (Set $u^{ini}_{k+1,i}$, $y^{ini}_{k+1,i}$ as the initial trajectory and zero sequence as the input). If the last time interval's horizon is shrunk (i.e., $h_r \neq 0$), the corresponding W_{0r} and $d_{k+1,r}$ should be calculated in a similar way.

In Equation (27), the length of $u_{k+1,i}$ is h. Thus the input u_d is assumed to be persistently exciting of order $h + 2n$, which has significantly relaxed the assumption in *Algorithm 1* (which is $N + 2n$ as the trial length N is usually much bigger compared to the system order n).

Remark 2. Algorithm 2 solves the problem every h time steps and then applies the input $u_{k+1,i}$ (over these h time steps) to the system (instead of just the first one), which is different from the conventional receding horizon control.

4.2 Convergence Analysis

The convergence properties of *Algorithm 2* are described in the following theorem.

Theorem 4. Algorithm 2 guarantees the tracking error norm asymptotically converges to zero, i.e., $\lim_{k\to\infty} e^P_k = 0$. If Q and R are scalar weightings, i.e., $Q = qI$, $R = rI$, where $q, r > 0 \in \mathbb{R}$, the proposed algorithm achieves monotonic convergence if and only if

$$\overline{\sigma}((I + \frac{q}{r} F G \tilde{G}^T F^T)^{-1}) < 1, \tag{29}$$

for which a sufficient condition is given by

$$\frac{q}{r} > \frac{2}{\underline{\sigma}(G\tilde{G}^T)}, \tag{30}$$

where $\underline{\sigma}(\cdot)$ is the minimum singular value, $\overline{\sigma}(\cdot)$ denotes the maximum singular value, $\tilde{G} = \mathrm{diag}(G_h, G_h, \cdots, G_h, G_{h_r})$ and G_h, G_{h_r} are defined in a similar way as in (4).

Proof. The proof is omitted here due to space reasons.

Algorithm 2 has relaxed the persistently exciting assumption. It can guarantee the asymptotic convergence of the tracking error norm, but monotonic convergence is only achieved under certain conditions. To overcome this issue, in the next section, we will develop a novel algorithm to simultaneously relax the persistently exciting assumption and maintain the monotonic convergence.

5. DATA-DRIVEN TRIAL PARTITION BASED POINT-TO-POINT NOILC

In this section, we will develop the data-driven trial partition based point-to-point NOILC algorithm.

5.1 Point-to-point NOILC Based on Trial Partition

The idea of the algorithm is shown in Fig. 2. In this algorithm, the trial will be partitioned for relaxing the persistently exciting assumption. The decision variables $w_{k,i} \in \mathbb{R}^{2(n+h)}$, $i = 1, \cdots, p-1$ and $w_{k,p} \in \mathbb{R}^{2(n+h_r)}$

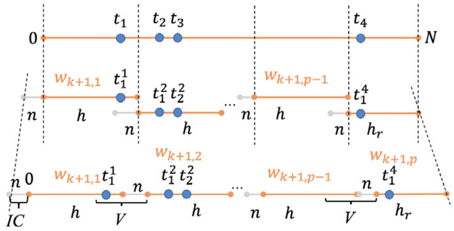

Fig. 2. Illustration of *Algorithm 3*

(where p is the number of intervals, h is the horizon and h_r is the shrunk horizon) of the new formulation contain both input and output, are defined as

$$w_{k+1,i} = \begin{bmatrix} u_{k+1}(ih-n : ih+h-1) \\ y_{k+1}(ih-n+1 : ih+h) \end{bmatrix}, \quad w_{k+1,p} = \begin{bmatrix} u_{k+1}((p-1)h-n : N-1) \\ y_{k+1}((p-1)h-n+1 : N) \end{bmatrix}, \tag{31}$$

and the whole trajectory $\boldsymbol{w}_{k+1}$ is given by

$$\boldsymbol{w}_{k+1} = \begin{bmatrix} w_{k+1,1}^T & w_{k+1,2}^T & \cdots & w_{k+1,p}^T \end{bmatrix}^T. \tag{32}$$

In *Algorithm 2*, one interval only focuses on its own optimisation problem and thus the monotonic convergence is lost. In this section, the trajectory of each interval has to be 'considered' with others by adding constraints and then the global optimum can be achieved. The idea is formally presented in the following algorithm.

Algorithm 3. Given a trajectory of the system (1) $w_d = \mathrm{col}(u_d, y_d)$ where u_d is persistently exciting of order $h+2n$, reference r^P, weighting matrices Q and R, past trajectory $\boldsymbol{w}_k$, the input u_{k+1} can be obtained by the following steps.

(1) Solving the following optimisation problem for $\boldsymbol{w}_{k+1}$

$$\begin{aligned} \underset{\boldsymbol{w}_{k+1}}{\text{minimise}} \quad & \left\|e_{k+1}^P\right\|_Q^2 + \left\|\boldsymbol{F^i}(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k)\right\|_R^2 \\ \text{s.t.} \quad & \begin{bmatrix} \boldsymbol{IC}^T & \boldsymbol{V}^T & I \end{bmatrix}^T \boldsymbol{w}_{k+1} = \begin{bmatrix} 0 & 0 & \boldsymbol{W} g_{k+1}^T \end{bmatrix}^T \\ & e_{k+1}^P = r^P - y_{k+1}^P, \; y_{k+1}^P = \boldsymbol{F^o} \boldsymbol{w}_{k+1} \end{aligned} \tag{33}$$

where $\boldsymbol{F^i}$ and $\boldsymbol{F^o}$ are the input and output selection matrix that selects the input and output information from each interval respectively. $\boldsymbol{IC}$ is used to constrain the initial trajectory to zero, which defines as

$$\boldsymbol{IC} = \begin{bmatrix} I_{2n} & \mathbf{0}_{2n,2(pn+N-n)} \end{bmatrix}, \tag{34}$$

$\boldsymbol{V}$ puts a constraint between two consecutive intervals, i.e., the last l samples' trajectory of the previous interval needs to be the 'past' trajectory of the current interval.

$$\boldsymbol{V} = \left[\begin{array}{cc|cc|cc|c} \mathbf{0}_{2n,2h} & I_{2n} & -I_{2n} & \mathbf{0}_{2n,2h} & 0 & \cdots & \cdots \\ 0 & \cdots & \mathbf{0}_{2n,2h} & I_{2n} & -I_{2n} & \mathbf{0}_{2n,2h} & \ddots \end{array}\right],$$

$\boldsymbol{W} g_{k+1}$ denotes the data-driven input/output relationship for $\boldsymbol{w}_{k+1}$ and $\boldsymbol{W}$ is given by

$$\boldsymbol{W} = \mathrm{diag}(W_{dn}, W_{dn}, \cdots, W_{dn}, W_{dnr}), \tag{35}$$

where $W_{dn} = \mathcal{H}_{n+h}(w_d)$ and $W_{dnr} = \mathcal{H}_{n+h_r}(w_d)$.

(2) The input is given by

$$u_{k+1} = \boldsymbol{F^i} \boldsymbol{w}_{k+1}. \tag{36}$$

Note the above algorithm has an analytical solution

$$u_{k+1} = \boldsymbol{F^i} \boldsymbol{W} \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} H & M^T \\ M & 0 \end{bmatrix}^\dagger \begin{bmatrix} -f \\ 0 \end{bmatrix}, \tag{37}$$

where

$$H = \boldsymbol{W}^T \boldsymbol{F^o}^T Q \boldsymbol{F^o} \boldsymbol{W} + \boldsymbol{W}^T \boldsymbol{F^i}^T R \boldsymbol{F^i} \boldsymbol{W},$$

$$M = \begin{bmatrix} \boldsymbol{IC}\boldsymbol{W}^T & \boldsymbol{V}\boldsymbol{W}^T \end{bmatrix}^T,$$

$$f = -\boldsymbol{W}^T \boldsymbol{F^o}^T Q \boldsymbol{r}^P - \boldsymbol{W} \boldsymbol{F^i}^T R \boldsymbol{F^i} \boldsymbol{w}_k.$$

The derivation of *Algorithm 3* is omitted here for brevity.

5.2 Convergence Properties

Algorithm 3 has the following properties:

Theorem 5. Algorithm 3 has the same convergence properties as the model-based point-to-point NOILC algorithm as given in Theorem 3.

Proof. The proof is omitted here due to space reasons.

By partitioning the trial into small intervals, the data-driven trial partition based point-to-point NOILC algorithm has relaxed the persistently exciting assumption on the existing input to the order of $h + 2n$. Compared with *Algorithm 2*, *Algorithm 3* maintains the monotonic convergence properties as well as obtains the minimum energy solution when $u_0 = 0$.

6. SIMULATION EXAMPLES

In this section, we will perform simulations to verify the effectiveness of the proposed algorithms. A model of the gantry robot system (Ratcliffe et al., 2006) is used. The Z-axis, which is modelled as a 3rd order SISO LTI system, is used. The transfer function is

$$H(z) = \frac{3.6482 \times 10^{-4}(z^2 + 0.09791z + 0.005951)}{(z-1)(z^2 + 0.005922z + 0.451 \times 10^{-4})}. \tag{38}$$

The initial state is set to zero. The trial length is $N = 200$. The reference is given by

$$r^P(t) = 0.005 \sin(\frac{1}{50}\pi t - \frac{\pi}{2}) + 0.005, \quad t = 1, 75, 125, 150.$$

The tracking performance of proposed algorithms and the effect of horizon h are shown in Fig. 3. We fix $Q = I$ and $R = 3 \times 10^{-6} I$. The horizon is chosen as $3, 20, 200$ for *Algorithms 2* and *3*. The figure shows that the identical tracking performance of *Algorithm 1* and all choices of horizon h in *Algorithm 3* to the model-based point-to-point NOILC algorithm which verifies *Theorems 3* and *5*. It also shows that all choices of the horizon in *Algorithm 2* lead to the perfect tracking which verifies *Theorem 4*. Besides, a larger horizon has a faster convergence speed. When the horizon equals the trial length, i.e., $h = 200$, it can recover *Algorithm 1*.

Next, we compare the input energy cost of each algorithm. We use the same setting as the previous simulation. As shown in Fig. 4, the input generated by *Algorithms 1* and *3* obtain the minimum energy solution, i.e., $\|u\|_R = 0.0050$. On the other hand, for *Algorithm 2*, the minimum input energy cost can not be generally achieved except when horizon $h = N$, which recovers *Algorithm 1*.

We conduct simulations to explore the effect of the weighting matrices Q, R of proposed algorithms. The results indicate that a larger ratio of Q to R will have a faster convergence speed. This is as expected since the same observation can be found in the model-based point-to-point NOILC design (Owens et al., 2013). For space reasons, the results are omitted.

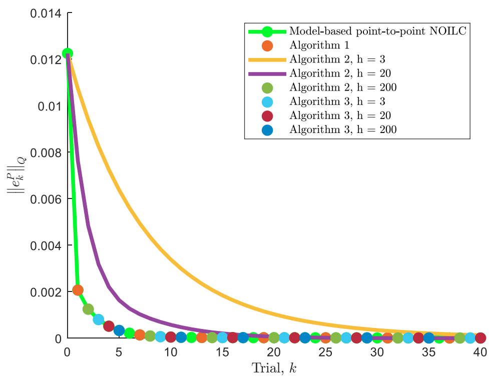

Fig. 3. Convergence of tracking error norm for different h

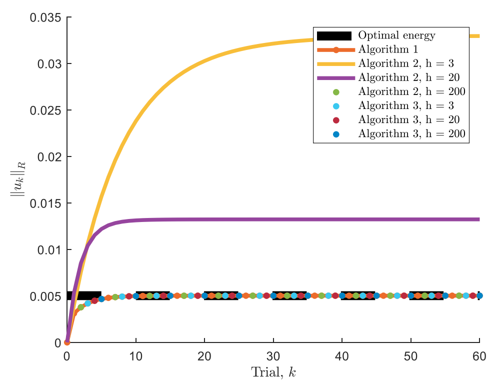

Fig. 4. Input energy cost of different algorithms

7. CONCLUSION

Model-based ILC algorithms tend to have good convergence properties but require a system model, which may be difficult or expensive to get in practice. To solve this issue, our previous work has developed a data-driven framework for high performance tracking tasks. However, it cannot be applied to point-to-point tasks that have a range of practical applications. To address this limitation, we develop a data-driven point-to-point NOILC framework. The identical performance of the proposed data-driven algorithm and the model-based algorithm is proved rigorously. To relax the existing data persistently exciting assumption, we further develop a receding horizon based and a trial-partition based point-to-point NOILC algorithms. Simulations are then performed to illustrate the effectiveness of the proposed designs. Future research needs to consider the robustness issue and the experimental verification.

REFERENCES

Amann, N., Owens, D.H., and Rogers, E. (1996). Iterative learning control using optimal feedback and feedforward actions. *International Journal of Control*, 65(2), 277–293.

Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering operation of robots by learning. *Journal of Robotic systems*, 1(2), 123–140.

Armstrong, A.A., Wagoner Johnson, A.J., and Alleyne, A.G. (2021). An improved approach to iterative learning control for uncertain systems. *IEEE Transactions on Control Systems Technology*, 29(2), 546–555.

Bolder, J., Kleinendorst, S., and Oomen, T. (2018). Data-driven multivariable ILC: Enhanced performance by eliminating L and Q filters. *International Journal of Robust and Nonlinear Control*, 28(12), 3728–3751.

Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3), 96–114.

Chen, B., Jiang, Z., and Chu, B. (2023). Distributed data-driven iterative learning control for consensus tracking. In *IFAC World Congress 2023, to appear 2023*. Elsevier.

Chi, R., Hou, Z., Jin, S., and Huang, B. (2019). An improved data-driven point-to-point ILC using additional on-line control inputs with experimental verification. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(4), 687–696.

de Rozario, R. and Oomen, T. (2019). Data-driven iterative inversion-based control: Achieving robustness through nonlinear learning. *Automatica*, 107, 342–352.

Freeman, C.T., Rogers, E., Hughes, A.M., Burridge, J.H., and Meadmore, K.L. (2012). Iterative learning control in health care: Electrical stimulation and robotic-assisted upper-limb stroke rehabilitation. *IEEE Control Systems Magazine*, 32(1), 18–43.

Harte, T.J., Hätönen, J.J., and Owens, D.H. (2005). Discrete-time inverse model-based iterative learning control: Stability, monotonicity and robustness. *International Journal of Control*, 78(8), 577–586.

Janssens, P., Pipeleers, G., and Swevers, J. (2012). A data-driven constrained norm-optimal iterative learning control framework for LTI systems. *IEEE Transactions on Control Systems Technology*, 21(2), 546–551.

Jiang, Z. and Chu, B. (2022). Norm optimal iterative learning control: A data-driven approach. *IFAC-PapersOnLine*, 55(12), 482–487.

Lim, I., Hoelzle, D.J., and Barton, K. (2017). A multi-objective iterative learning control approach for additive manufacturing applications. *Control Engineering Practice*, 64, 74–87.

Markovsky, I. and Rapisarda, P. (2008). Data-driven simulation and control. *International Journal of Control*, 81(12), 1946–1959.

Owens, D.H. (2016). *Iterative Learning Control: An Optimization Paradigm*. Springer London.

Owens, D.H., Freeman, C.T., and Van Dinh, T. (2013). Norm-optimal iterative learning control with intermediate point weighting: Theory, algorithms, and experimental evaluation. *IEEE Transactions on Control Systems Technology*, 21(3), 999–1007.

Owens, D.H., Hätönen, J.J., and Daley, S. (2009). Robust monotone gradient-based discrete-time iterative learning control. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 19(6), 634–661.

Ratcliffe, J.D., Lewin, P.L., Rogers, E., Hätönen, J.J., and Owens, D.H. (2006). Norm-optimal iterative learning control applied to gantry robots for automation applications. *IEEE Transactions on Robotics*, 22(6), 1303–1307.

Tayebi, A. (2004). Adaptive iterative learning control for robot manipulators. *Automatica*, 40(7), 1195–1203.

Willems, J.C., Rapisarda, P., Markovsky, I., and De Moor, B.L. (2005). A note on persistency of excitation. *Systems & Control Letters*, 54(4), 325–329.