Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-12 (2022) 482–487

Norm Optimal Iterative Learning Control: A Data-Driven Approach

Zheng Jiang * **Bing Chu** *

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

Abstract: Iterative learning control (ILC) is a control design method that can improve the tracking performance for systems working in a repetitive manner by learning from the previous iterations. Norm optimal ILC is a well known ILC design with appealing convergence properties, e.g. monotonic error norm convergence. However, it requires an explicit system model in the design, which can be difficult or expensive to obtain in practice. To address this problem, this paper proposes a data-driven norm optimal ILC design exploiting recent development in data-driven control. A receding horizon implementation of the design is further developed to relax the requirement on data. Convergence properties of the design are analysed rigorously and simulation examples are presented to demonstrate the effectiveness of the method.

Keywords: Iterative learning control, data-driven control, convergence analysis, control design, simulation.

1. INTRODUCTION

Many practical applications require the system to have high tracking performance and to work in a repetitive manner. Iterative learning control (ILC) can learn from data (i.e., input and error) generated in past iterations to achieve high tracking performance even if there is no accurate model available. Originating from Arimoto et al. (1984), it now has attracted intensive research in different applications such as robotics (Armstrong et al. (2021)), stroke rehabilitation (Freeman et al. (2012)), additive manufacturing (Lim et al. (2017)) and broiler production (Johansen et al. (2021)).

Generally, ILC designs can be divided into model-free methods and model-based methods according to whether the model information is used. Model-free ILC designs, such as proportional-integral-derivative (PID)-type ILC (Arimoto et al. (1984)), adaptive ILC (Tayebi (2004)), phase-lead ILC (Wang and Longman (1996)) do not require the system model, but the parameters needed to be carefully tuned to ensure good convergence. On the other hand, model-based ILC algorithms, for example, inverse-based ILC (Harte et al. (2005)), gradient ILC (Owens et al. (2009)), norm optimal ILC (Amann et al. (1996)), use the system model in the input updating law. Generally, model-based methods tend to have better performance. However, the system model can be difficult or very expensive to obtain in practice. For more comprehensive review of ILC, please refer to Owens (2016), Bristow et al. (2006).

There have been various attempts to relax/remove the model information requirement in model-based ILC design while still maintaining the good convergence properties. As examples, in Janssens et al. (2012), the input and output data are used to estimate the system impulse response which is then used in model-based ILC design. Blanken et al. (2020) proposes a basis function method to iteratively learn the inverse system which can be used in the norm optimal ILC design. Bolder et al. (2018) utilises an online experiment on the plant to get the response of the adjoint system which replaces the model based calculation. However, the preceding algorithms need to explicitly or implicitly identify the system model which can be non-trivial, or extra experiments on the plant between trials that may not be desirable in practice.

To address these limitations, this paper develops a model-free norm optimal ILC (NOILC) framework based on the recently developments in data-driven control theory (Markovsky and Rapisarda (2008); De Persis and Tesi (2019)). We first develop a model-free data-driven NOILC algorithm using a well-known result in data-driven control called the Willems' fundamental lemma (Willems et al. (2005)). We show that the proposed algorithm can achieve the same tracking performance as model-based NOILC design with rigorous proof. This data-driven NOILC algorithm has a persistently exciting requirement of the input signal which may be difficult to satisfy in some real applications. To further relax the requirement, a novel data-driven ILC algorithm in a receding horizon manner is then proposed and the convergence properties are analysed rigorously.

The rest of the paper is organized as follows. The system dynamics are described and norm optimal iterative learning control algorithm is introduced in Section 2. A data-driven NOILC framework is presented in Section 3. The further developed data-driven receding horizon based NOILC is given in Section 4. Simulation examples based on the gantry robot system are provided to demonstrate the algorithms' performance in Section 5. Finally, conclusions and future work are given in Section 6.

10.1016/j.ifacol.2022.07.358

2. PROBLEM FORMULATION

In this section, the system dynamics are first described. Then the NOILC design problem is introduced.

2.1 System Dynamics

Consider the following single-input single-output (SISO) discrete time linear time-invariant (LTI) system

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

where k is the iteration (or trial) number, $t \in [0, N]$ is the time index, N is the trial length, $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}$ and $y_k(t) \in \mathbb{R}$ are the system state (n is the system order), input and output respectively. A, B and C are system matrices of proper dimensions. This system works repetitively to track a reference sequence $r(t)$. After t reaches N, a new iteration begins: t will be reset to 0 and the system state will be reset to the same initial condition, i.e., $x_k(0) = x_0$.

Assume that the relative degree is unity ($CB \neq 0$), the lifted form input and output relationship of (1) at iteration k can be described as (Hätönen et al. (2004))

$$y_k = Gu_k + d_k \tag{2}$$

where the input and output sequence u_k and y_k are defined as

$$\begin{aligned} u_k &= [u_k(0), u_k(1), \cdots, u_k(N-1)]^T \\ y_k &= [y_k(1), y_k(2), \cdots, y_k(N)]^T. \end{aligned} \tag{3}$$

The input sequence u_k and output sequence y_k are defined in the Hilbert space $\mathcal{U} = \mathbb{R}^N$ and $\mathcal{Y} = \mathbb{R}^N$ respectively with the inner products

$$\langle u, v\rangle_R = u^T Rv, \quad \langle y, v\rangle_Q = y^T Qv, \tag{4}$$

and the associated norms

$$\|u\|_R = \sqrt{u^T Ru}, \quad \|y\|_Q = \sqrt{y^T Qy}, \tag{5}$$

where $R \in \mathbb{R}^{N\times N}$ and $Q \in \mathbb{R}^{N\times N}$ are positive definite weighting matrices.

The system model G in matrix form can be represented as

$$G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{bmatrix}, \tag{6}$$

and d is the system initial response given by

$$d = \mathcal{O}x_0, \tag{7}$$

where the observability matrix $\mathcal{O}(A, C)$ is defined as

$$\mathcal{O}(A, C) = \mathrm{col}(CA, CA^2, \cdots, CA^N), \tag{8}$$

in which $\mathrm{col}(a, b) := \begin{bmatrix} a \\ b \end{bmatrix}$. Besides, the reference r in vector form can be defined as

$$r = [r(1), r(2), \cdots, r(N)]^T \tag{9}$$

and the tracking error e_k at iteration k is given by

$$e_k = r - y_k = r - Gu_k - d. \tag{10}$$

2.2 Norm Optimal Iterative Learning Control

ILC Design Problem: The ILC design problem is to find a control updating law

$$u_{k+1} = f(u_k, e_k) \tag{11}$$

such that the system tracking error can asymptotically converge to 0, i.e.

$$\lim_{k\to\infty} e_k = 0. \tag{12}$$

Note that in the model-based ILC design, the system model information, i.e. G, is involved in the control updating law (11). With the system model, these designs could have appealing tracking performance. For example, the well-known NOILC algorithm (Amann et al. (1996)), which can guarantee monotonic convergence of the tracking error norm, is introduced next.

The NOILC algorithm minimises the following cost function to update the input u_{k+1} for iteration $k+1$

$$\begin{aligned} u_{k+1} = &\arg\min_{u_{k+1}} \|e_{k+1}\|_Q^2 + \|u_{k+1} - u_k\|_R^2, \\ &\text{s.t.} \quad e_{k+1} = r - Gu_{k+1} - d. \end{aligned} \tag{13}$$

By calculating the derivative of (13) and setting it to zero, the optimal solution is

$$u_{k+1} = u_k + (G^T QG + R)^{-1} G^T Qe_k. \tag{14}$$

The algorithm has nice convergence properties, as shown in the following theorem.

Theorem 1. (Amann et al. (1996)) The NOILC algorithm can achieve monotonic convergence in the tracking error norm to zero, i.e.

$$\|e_{k+1}\|_Q \leq \|e_k\|_Q, \quad \lim_{k\to\infty} e_k = 0. \tag{15}$$

By utilizing system model information G, NOILC can achieve monotonic convergence of the tracking error norm. However, the model information is not always easy to obtain in practice. In what follows, we will develop a model-free data-driven control law that does not require a system model but still has appealing convergence properties as model-based NOILC algorithm.

3. DATA-DRIVEN NORM OPTIMAL ITERATIVE LEARNING CONTROL

In this section, some preliminary results from the data-driven control are first introduced. Then the data-driven NOILC algorithm is developed with the convergence rigorously analysed.

3.1 Preliminary Results on Data-Driven Control

In this subsection, we review a key result in data-driven control, namely, the Willems' fundamental lemma (Willems et al. (2005)) which is the foundation of our data-driven NOILC algorithm. We first introduce some necessary definitions.

Given T length system input u and output y, the trajectory w is defined as

$$w := \begin{bmatrix} u \\ y \end{bmatrix}. \tag{16}$$

All the T length trajectories $w := \mathrm{col}(u, y)$ generated by G form a subspace $\mathcal{G}_T$ of $\mathcal{W}$, and is given below

$$\begin{aligned} \mathcal{G}_T := \Big\{ &\begin{bmatrix} u \\ y \end{bmatrix} \in \mathbb{R}^{2T} | \exists x(t) \in \mathbb{R}^n, \\ &\text{such that } x(t+1) = Ax(t) + Bu(t) \\ &\text{and } y(t) = Cx(t) \Big\} \end{aligned} \tag{17}$$

Hankel matrix $\mathcal{H}_{t1}(x)$ of a signal $x \in \mathbb{R}^T$ is defined as

$$\mathcal{H}_{t1}(x) = \begin{bmatrix} x(0) & x(1) & \cdots & x(T-t_1) \\ x(1) & x(2) & \cdots & x(T-t_1+1) \\ \vdots & \vdots & \ddots & \vdots \\ x(t_1-1) & x(t_1) & \cdots & x(T-1) \end{bmatrix}. \quad (18)$$

The signal x is called persistently exciting of order t_1 if the Hankel matrix (18) has full row rank.

With the above definitions, we can define the following the input and output Hankel matrices and partitions

$$U = \mathcal{H}_{T_{ini}+N}(u) = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \quad Y = \mathcal{H}_{T_{ini}+N}(y) = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}, \quad (19)$$

where T_{ini} is an integer representing the length of the initial input/output response (defining the initial condition of the system). The blocks $U_p, Y_p \in \mathbb{R}^{T_{ini}\times(T-T_{ini}-N+1)}$ (p denotes 'past') are used to calculate the initial conditions while the blocks U_f, $Y_f \in \mathbb{R}^{N\times(T-T_{ini}-N+1)}$ (f denotes 'future') are used to calculate the system response.

We can now present the Willems' fundamental lemma:

Theorem 2. (Willems et al. (2005)) Consider the system (1) and assume it is controllable. Given a T sample long data trajectory $w_d := \text{col}(u_d, y_d)$ generated by (1). If the system input u_d is persistently exciting of order $t+n$, then any t samples long trajectory w of $\mathcal{G}$ 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 $g \in \mathbb{R}^{T-t+1}$, is a trajectory of $\mathcal{G}_t$, i.e.

$$\text{col span}(\mathcal{H}_N(w_d)) = \mathcal{G}_t, \quad (20)$$

in which col span$(\cdot)$ denotes the column span of the matrix and $\mathcal{G}_t$ is defined a similar way as in (17).

Using the Willems' fundamental lemma, data-driven simulation and control design are proposed in Markovsky and Rapisarda (2008) without using system model. Based on these results, we develop a data-driven NOILC algorithm to remove requirement on the system model which will be introduced next.

3.2 Algorithm Description

The proposed data-driven NOILC algorithm is described as follows:

Algorithm 1. Given T samples long trajectory data w_d for system (1). Assume u_d is persistently exciting of order $N+n$, and T_{ini} is no smaller than the observability index of (1). Then the input u_{k+1} can be calculated by minimising the following cost function

$$\begin{aligned} u_{k+1} = &\arg\min_{u_{k+1}} \|e_{k+1}\|_Q^2 + \|u_{k+1}-u_k\|_R^2 \\ \text{s.t.} \quad &\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} \mathbf{0}_{T_{ini},1} \\ \mathbf{0}_{T_{ini},1} \\ u_{k+1}-u_k \\ y_{k+1}-y_k \end{bmatrix}, \end{aligned} \quad (21)$$

where $\mathbf{0}_{m,n}$ denotes $m \times n$ zero matrix. Compare to the model based NOILC (13), the system model is represented by a data-driven representation (21) using *Theorem 2* as a linear combination of the existing data.

The data-driven NOILC solution is given by

$$u_{k+1} = u_k + \begin{bmatrix} I \\ \mathbf{0}_{N,N} \end{bmatrix}^T W_0(W_0^T S W_0)^\dagger W_0^T S \begin{bmatrix} \mathbf{0}_{N,1} \\ e_k \end{bmatrix}, \quad (22)$$

where the superscription † denotes the pseudo-inverse and W_0 is calculated as follows.

(1) Calculate the solution g_r of the following equation

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g_r = \begin{bmatrix} \mathbf{0}_{T_{ini},T-N+1} \\ \mathbf{0}_{T_{ini},T-N+1} \\ \mathcal{H}_N(u_d) \end{bmatrix} \quad (23)$$

(2) The result obtained is then used to calculate

$$Y_f g_r = Y_0 \quad (24)$$

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

$$W_0 = \begin{bmatrix} \mathcal{H}_N(u_d) \\ Y_0 \end{bmatrix} \quad (25)$$

The detailed derivations of (22) to (25) use results from Markovsky and Rapisarda (2008) and can be found in Appendix A (in the proof of *Theorem 3*).

3.3 Convergence Analysis

The performance of the proposed data-driven NOILC can be described as the following theorem:

Theorem 3. The tracking performance of data-driven NOILC is identical to the model-based NOILC (i.e., solution u_{k+1} in (22) is identical to solution (14)). Consequently, the proposed algorithm can guarantee monotonic convergence of the tracking error norm.

Proof. See the Appendix A.

From the above theorem, it can be seen that the proposed algorithm has the nice convergence properties of the model-based NOILC since their tracking performance are identical. Moreover, the proposed algorithm does not need the system model information which is required in the model-based NOILC. Instead, it needs the past trajectory data. It is worth noting that for the assumption of *Algorithm 1* to hold, the length of data has to be sufficiently long, i.e., $T \geq 2(T_{ini}+N+n)-1$. When the trial length is long, the assumption of persistently exciting of the system input may not easily to satisfied. To further relax this requirement, we proposed the receding horizon based data-driven NOILC in the next section.

4. RECEDING HORIZON BASED DATA-DRIVEN NOILC

In this section, we developed a data-driven ILC algorithm in a receding horizon manner. The convergence analysis with rigorous proof is presented.

4.1 Algorithm Description

Algorithm 2. Given T samples long trajectory w_d for system (1). Assume u_d is persistently exciting of order $h+n$ (where h is the prediction horizon) and T_{ini} is no smaller than the observability index of (1). Then the input $u_{k+1,t}$ can be calculated by minimising the following cost function

$$u_{k+1,t} = \arg\min_{u_{k+1,t}} \|e_{k+1,t}\|_Q^2 + \|u_{k+1,t} - u_{k,t}\|_R^2 \quad \text{s.t.} \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{k+1,t}^{ini} \\ y_{k+1,t}^{ini} \\ u_{k+1,t} \\ y_{k+1,t} \end{bmatrix}, \tag{26}$$

where

$$\begin{aligned} u_{k+1,t}^{ini} &= [u_{k+1}(t-T_{ini})\ u_{k+1}(t-T_{ini}+1)\ \cdots \\ &\quad \cdots\ u_{k+1}(t-1)]^T \in \mathbb{R}^{T_{ini}} \\ y_{k+1,t}^{ini} &= [y_{k+1}(t-T_{ini}+1)\ y_{k+1}(t-T_{ini}+2)\ \cdots \\ &\quad \cdots\ y_{k+1}(t)]^T \in \mathbb{R}^{T_{ini}} \end{aligned} \tag{27}$$

are initial input and initial output at iteration $k+1$, time t, and their associated norms are defined in (5).

$$\begin{aligned} u_{k+1,t} &= [u_{k+1}(t)\ u_{k+1}(t+1)\ \cdots\ u_{k+1}(t+h-1)]^T \in \mathbb{R}^h \\ y_{k+1,t} &= [y_{k+1}(t+1)\ y_{k+1}(t+2)\ \cdots\ y_{k+1}(t+h)]^T \in \mathbb{R}^h \\ e_{k+1,t} &= [e_{k+1}(t+1)\ e_{k+1}(t+2)\ \cdots\ e_{k+1}(t+h)]^T \in \mathbb{R}^h \end{aligned} \tag{28}$$

are input sequence, output sequence and error sequence at iteration $k+1$, time t respectively, and their associated norms are defined a similar way as in (5).

The data-driven receding horizon based NOILC solution is given by

$$\begin{aligned} u_{k+1,t} = &u_{k,t} + \begin{bmatrix} I \\ \mathbf{0}_{h,h} \end{bmatrix}^T W_{0h}(W_{0h}^T S W_{0h})^\dagger W_{0h}^T \\ &\times S \begin{bmatrix} \mathbf{0}_{h,1} \\ e_{k,t} + d_{k,t} - d_{k+1,t} \end{bmatrix}, \end{aligned} \tag{29}$$

where $S = \begin{bmatrix} R & \\ & Q \end{bmatrix}$ is the positive definite weighting matrix.

W_{0h} is calculated by the same steps as (23) to (25) , $d_{k+1,t}$ is the system free response at iteration $k+1$, time t which is calculated as follows (by *Theorem 2*) .

(1) At iteration k, time t, solve the equation

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g_r = \begin{bmatrix} u_{k+1,t}^{ini} \\ y_{k+1,t}^{ini} \\ \mathbf{0}_{h,1} \end{bmatrix} \tag{30}$$

(2) Compute the free response by

$$d_{k+1,t} = Y_f g_r \tag{31}$$

Note that if the horizon needs to be shrunk (i.e., $h_r \neq 0$ where $h_r = N \bmod h$), the data-driven receding horizon based NOILC solution at time $N - h_r$ is given by

$$\begin{aligned} u_{k+1,N-h_r} = &u_{k,N-h_r} + \begin{bmatrix} I \\ \mathbf{0}_{h_r,h_r} \end{bmatrix}^T W_{0r}(W_{0r}^T S W_{0r})^\dagger \\ &\times W_{0r}^T S \begin{bmatrix} \mathbf{0}_{h_r,1} \\ e_{k,N-h_r} + d_{k,N-h_r} - d_{k+1,N-h_r} \end{bmatrix}, \end{aligned} \tag{32}$$

where W_{0r} is calculated by the same steps as (23) to (25), $d_{k+1,N-h_r}$ is calculated by the same steps as (30) to (31).

Remark 1. In *Algorithm 2*, the input u_d is assumed to be persistently exciting of order $h+n$, which has significantly relax the assumption in *Algorithm 1* (which is $N+n$ as the trial length N is usually much bigger compared to the system order n).

Remark 2. Note that (26) is slightly different from the traditional receding horizon control where the optimisation problem is solved at each time step. In the proposed algorithm, the optimisation problem is solved every h time steps and the obtained input (over these h time steps) then applied to the system (instead of just the first one).

4.2 Convergence Analysis

The convergence performance of the proposed data-driven receding horizon based NOILC algorithm is shown in the below theorem.

Theorem 4. The proposed data-driven receding horizon based NOILC can guarantee the tracking error norm asymptotically converges to zero, i.e.$\lim_{k\to\infty} e_k = 0$. Moreover, if the weighting matrices Q and R are scalar weightings, i.e., $Q = qI$, $R = rI$, where q and r are real positive numbers, the proposed algorithm can achieve monotonic convergence (i.e., $\|e_{k+1}\| \leq \|e_k\|$ and $\lim_{k\to\infty} e_k = 0$) iff

$$\overline{\sigma}((I + \frac{q}{r} G\tilde{G}^T)^{-1}) < 1, \tag{33}$$

for which a sufficient condition is given by

$$\frac{q}{r} > \frac{2}{\underline{\sigma}(G\tilde{G}^T)}, \tag{34}$$

where $\underline{\sigma}(\cdot)$ denotes the minimum singular value, $\overline{\sigma}(\cdot)$ denotes the maximum singular value, $\tilde{G}$ is defined as

$$\tilde{G} = \begin{bmatrix} G_h & & & & \\ & G_h & & & \\ & & \ddots & & \\ & & & G_h & \\ & & & & G_{h_r} \end{bmatrix}, \tag{35}$$

where G_h and G_{h_r} are defined in a similar way as in (6).

Proof. The proof is omitted here due to the space reason.

This theorem shows that the proposed *Algorithm 2* can achieve asymptotic convergence. It also provides the monotonic convergence condition: by increasing the ratio of Q to R, the monotonic convergence can be achieved. It is also worth noting that when $h = N$, the data-driven NOILC *Algorithm 1* proposed in Section 3 is recovered.

Remark 3. The monotonic convergence condition requires the minimum singular values of $G\tilde{G}$ which can be link to the H_∞ norm of G^{-1} and $\tilde{G}^{-1}$. The H_∞ norm of a system can be estimated from data, e.g., using the results from van Heusden et al. (2007) or Oomen et al. (2014) which is omitted here for brevity.

5. SIMULATION EXAMPLES

In this section, the proposed algorithms are verified by numerical simulations.

The system used in this paper comes from a gantry robot (Ratcliffe et al. (2006)) which is shown in Fig. 1. The gantry robot contains three independently controlled perpendicular axes: the X-axis and the Y-axis moves horizontally while the Z-axis moves vertically and placed on the top of other axes. The Z-axis is modelled as a 3rd order SISO LTI system whose transfer function is given by

$$H(s) = \frac{15.8869(s + 850.3)}{s(s^2 + 707.6s + 3.377 \times 10^5)}. \tag{36}$$

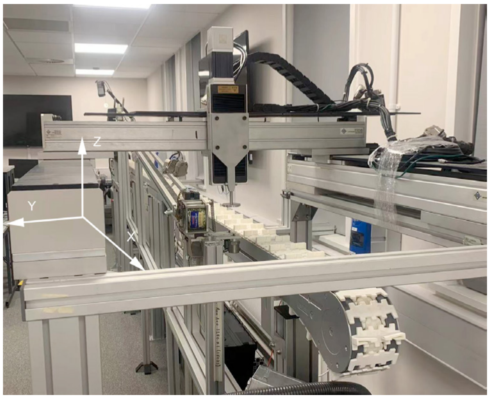

Fig. 1. Three-axis gantry robot

The system is sampled by a zero-order hold with a sampling time of $T_S = 0.01$s and the initial state is assumed to be zero. The trial length is $N = 200$. To verify the effectiveness of proposed algorithms, we simulate them over 60 iterations for different horizons to track the reference signal

$$r = 0.005 \sin(2\pi t - \frac{\pi}{2}) + 0.005 \tag{37}$$

This reference represents a typical pick and place control task of the Z-axis in a gantry robot.

The effect of different horizons is investigated which is shown in Fig. 2. The weightings are chosen to be $Q = I$ and $R = 8 \times 10^{-6} I$ and the horizon h is chosen to be $1, 2, 3, 5, 10, 20, 60, 120, 200$ respectively. Fig. 2 shows that different choices of the horizon can achieve the perfect tracking which verifies the *Theorem 4*. When the horizon and the trial length are the same (i.e., $h = 200$ in this simulation), the data-driven receding horizon based NOILC can recover the data-driven NOILC *Algorithm 1*. We also compared the model-based NOILC (which is denoted by the black dashed line) with the data-driven algorithms. The tracking performance of the model-based NOILC and *Algorithm 1* are identical which verify the *Theorem 3*. Besides, using a larger horizon tends to have a faster convergence speed.

The reference and the output of the proposed *Algorithm 2* for different iterations k are shown in Fig. 3. The weightings are chosen to be $Q = I$, $R = 8 \times 10^{-6} I$ and $h = 5$.

The effect of weighting matrices Q and R is also investigated. Any choice of Q and R can achieve the desired tracking mission and a larger ratio of Q to R can lead to a faster convergence speed. The result is omitted here due to the space reason.

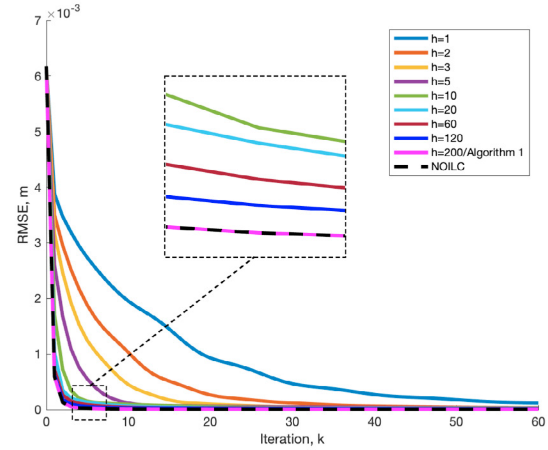

Fig. 2. Root mean square error (RMSE) for different horizon h

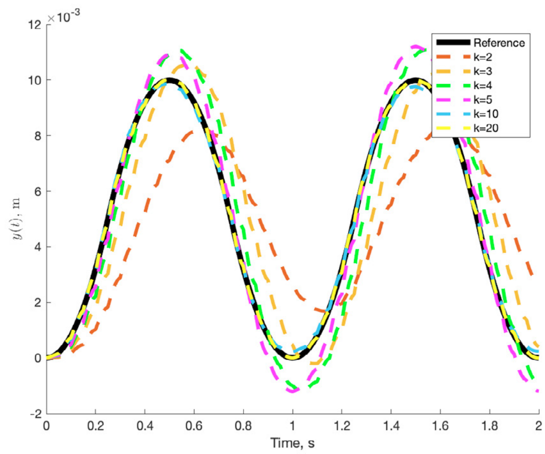

Fig. 3. Reference and output for different iterations

6. CONCLUSION

The well-known NOILC algorithm has appealing convergence properties. However it requires the system model in the control updating law which can be difficult or expensive to obtain in practice. In this paper, a data-driven NOILC framework has been proposed to remove the model requirement based on the recent developments in data-driven control. Our algorithm solves the NOILC design problem without using the system model. The tracking performance of the proposed algorithm has shown to be identical to the model-based NOILC. A novel data-driven ILC algorithm based on the receding horizon control has been further proposed to relax the requirement on the data. Numerical examples based on the gantry robot system have been presented to illustrate two proposed algorithms' performance. Future work will consider the model uncertainty including the effect of noise and the constraint handling problem, as well as their experimental verifications.

REFERENCES

Amann, N., Owens, D.H., and Rogers, E. (1996). Iterative learning control for discrete-time systems with exponential rate of convergence. *IEE Proceedings-Control Theory and Applications*, 143(2), 217–224.

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.

Blanken, L., Koekebakker, S., and Oomen, T. (2020). Data-driven feedforward tuning using non-causal rational basis functions: With application to an industrial flatbed printer. *Mechatronics*, 71, 102424.

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.

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

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.

Hätönen, J.J., Owens, D.H., and Moore, K.L. (2004). An algebraic approach to iterative learning control. *International Journal of Control*, 77(1), 45–54.

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.

Johansen, S.V., Jensen, M.R., Chu, B., Bendtsen, J.D., Mogensen, J., and Rogers, E. (2021). Broiler FCR optimization using norm optimal terminal iterative learning control. *IEEE Transactions on Control Systems Technology*, 29(2), 580–592.

Lim, I., Hoelzle, D.J., and Barton, K.L. (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.

Oomen, T., van der Maas, R., Rojas, C.R., and Hjalmarsson, H. (2014). Iterative data-driven $\mathcal{H}_\infty$ norm estimation of multivariable systems with application to robust active vibration isolation. *IEEE Transactions on Control Systems Technology*, 22(6), 2247–2260.

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

Owens, D.H., Hätönen, 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.

van Heusden, K., Karimi, A., and Bonvin, D. (2007). Data-driven estimation of the infinity norm of a dynamical system. In *2007 46th IEEE Conference on Decision and Control*, 4889–4894. IEEE.

Wang, Y. and Longman, R.W. (1996). Use of non-causal digital signal processing in learning and repetitive control. *Advances in the Astronautical Sciences*, 90(1), 649–668.

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.

Appendix A. PROOF OF THEOREM 3

To prove the *Theorem 3*, we need to show the trajectories of two algorithms at $k+1$, $k = 1, 2, \cdots$ are same, i.e.

$$\mathrm{col}(u_{k+1}, y_{k+1}) = w^D_{k+1} = w^M_{k+1}, \tag{A.1}$$

where w^D_{k+1} denotes the data-driven algorithm solution trajectory and w^M_{k+1} denotes the model-based solution trajectory.

First rewrite the control objective (21) as

$$\begin{aligned} u_{k+1} = &\arg\min_{u_{k+1}} \left\| \begin{bmatrix} u_{k+1} \\ y_{k+1} \end{bmatrix} - \begin{bmatrix} u_k \\ y_k + e_k \end{bmatrix} \right\|^2_S \\ \text{s.t.} \quad &\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} \mathbf{0}_{T_{ini},1} \\ \mathbf{0}_{T_{ini},1} \\ u_{k+1} - u_k \\ y_{k+1} - y_k \end{bmatrix}, \end{aligned} \tag{A.2}$$

where $S = \begin{bmatrix} R & \\ & Q \end{bmatrix}$ is the positive definite weighting matrix.

The zero initial condition response matrix W_0 obtain by Step (1) combined with *Theorem 2* has following property (Proposition 3, Markovsky and Rapisarda (2008))

$$\text{col span}(W_0) = \text{col span}(\begin{bmatrix} U_f \\ Y_f \end{bmatrix}) = \mathcal{G} = \text{col span}(\begin{bmatrix} I \\ G \end{bmatrix}), \tag{A.3}$$

thus there exist g_r such that

$$\begin{bmatrix} U_f \\ Y_f \end{bmatrix} g = W_0 g_r = \begin{bmatrix} u_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} I \\ G \end{bmatrix} u_{k+1}. \tag{A.4}$$

Hence, (A.2) can be written as

$$\begin{bmatrix} u_{k+1} \\ y_{k+1} \end{bmatrix} = W_0 \arg\min_{g_r} \left\| W_0 g_r - \begin{bmatrix} u_k \\ y_k + e_k \end{bmatrix} \right\|^2_S. \tag{A.5}$$

Calculate the derivative of $\left\| W_0 g_r - \begin{bmatrix} u_k \\ y_k + e_k \end{bmatrix} \right\|^2_S$ respect to g_r and set it equal to zero, it follows that

$$w^D_{k+1} = W_0 g_r = W_0 (W_0^T S W_0)^\dagger W_0^T S \begin{bmatrix} u_k \\ y_k + e_k \end{bmatrix}, \tag{A.6}$$

which is the data-driven analytic solution trajectory of (21). Note that (A.5) can be written into (using (A.4))

$$\begin{bmatrix} u_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} I \\ G \end{bmatrix} \arg\min_{u_{k+1}} \left\| \begin{bmatrix} I \\ G \end{bmatrix} u_{k+1} - \begin{bmatrix} u_k \\ y_k + e_k \end{bmatrix} \right\|^2_S, \tag{A.7}$$

which is exactly the model-based NOILC algorithm. Thus

$$\begin{aligned} \begin{bmatrix} u_{k+1} \\ y_{k+1} \end{bmatrix} &= W_0 (W_0^T S W_0)^\dagger W_0^T S \begin{bmatrix} u_k \\ y_k + e_k \end{bmatrix} = w^D_{k+1} \\ &= \begin{bmatrix} I \\ G \end{bmatrix} \left(\begin{bmatrix} I \\ G \end{bmatrix}^T S \begin{bmatrix} I \\ G \end{bmatrix} \right)^{-1} \begin{bmatrix} I \\ G \end{bmatrix}^T S \begin{bmatrix} u_k \\ y_k + e_k \end{bmatrix} = w^M_{k+1} \end{aligned}. \tag{A.8}$$

The last equation comes from (14). Therefore the tracking performance of data-driven NOILC is identical to the model-based NOILC and hence the monotonic convergence property is obvious. That completes the proof.