INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2021; **00**:1–23
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/acs

Iterative Learning Control for Repetitive Tasks with Randomly Varying Trial Lengths using Successive Projection

Zhihe Zhuang[1], Hongfeng Tao[1†], Yiyang Chen[2†], Vladimir Stojanovic[3], Wojciech Paszke[4]

[1] *Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, 1800 Lihu Road, Wuxi 214122, China.*
[2] *School of Mechanical and Electrical Engineering, Soochow University, 8 Jixue Road, Suzhou 215137, China.*
[3] *Faculty of Mechanical and Civil Engineering, Department of Automatic Control, Robotics and Fluid Technique, University of Kragujevac, 36000 Kraljevo, Serbia.*
[4] *Institute of Automation, Electronic and Electrical Engineering, University of Zielona Góra, Zielona Góra, Poland*

SUMMARY

This paper proposes an effective iterative learning control (ILC) approach based on successive projection scheme for repetitive systems with randomly varying trial lengths. A modified ILC problem is formulated to extend the classical ILC task description to incorporate a randomly varying trial length, while its design objective considers the mathematical expectation of its tracking error to evaluate the task performance. To solve this problem, this paper employs the successive projection framework to give an iterative input signal update law by defining the corresponding convex sets based on the design requirements. This update law further yields an ILC algorithm, whose convergence properties are proved to be held under mild conditions. In addition, the input signal constraint can be embedded into the design without violating the convergence properties to obtain an alternative algorithm. The performance of the proposed algorithms is verified using a numerical model to show the effectiveness at occasions with and without input constraints. Copyright © 2021 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Iterative learning control, Randomly varying trial length, Projection method, Input constraint

1. INTRODUCTION

Iterative learning control (ILC) is a control methodology specifically designed to improve the task performance of systems working repetitively. As explained in [1], its main idea is to utilize previous trial information to modify the system input signal to further achieve more precise tracking for a target repetitive task. ILC is designed for specific repetitive tasks with a finite time horizon, and its system state has to be initialized at the end of each trial. As

*Correspondence to: Hongfeng Tao and Yiyang Chen
†Email: taohongfeng@jiangnan.edu.cn and yychen90@suda.edu.cn

a simple but effective control strategy, ILC has been successfully implemented to various applications, and typical examples can be listed as industrial robots [2–4], injection molding machines [5], robotic-assisted biomedical systems [6] and chemical batch processing [7]. See [8] for a comprehensive review of ILC.

Classic ILC problems usually assume that their repetitive trial lengths are fixed constant values. However, research on how to relax this restriction has been conducted since in [9], where a new non-standard ILC algorithm that does not use the standard assumption of uniform trial length was developed for tracking periodic signals in repetitive control systems. It has been pointed out in [10] that the system task may end early in some trial due to safety considerations or under certain constraints in some practical applications. For instance, the work in [11] studied the functional electrical stimulation process for upper limb movement or gait assistance, and the results showed that the ILC implementation procedure may even be terminated before the end of the entire stimulation process due to the muscle fatigue of stroke patients. Meanwhile, the research in [12] reported that the entire task of a gantry crane has to stop when its load approaches an obstacle, since it is only designed to move within the local region of the required reference trajectory. In these applications, the ILC trial lengths are no longer identical, and modified ILC framework should be designed to adapt this change.

The existing literature has already showed the examples of ILC applications to the class of systems with non-uniform trial lengths. As stated in [11], the maximum batch length was defined as the full length and zero elements were appended to the trials without the exact full trial length. In this sense, classic ILC algorithms can then be employed to handle the repetitive tracking task based on the modified trial information. The subsequent research in [13] developed a design framework with monotonically convergence for linear systems with non-uniform trial lengths, but this framework does not involve a strict mathematical model for this kind of systems. Meanwhile, the work in [14–16] utilized random variables to denote the actual trial lengths, and hence proposed the convergence of the linear and nonlinear discrete systems in terms of mathematical expectation using an iteration-average operator. The following work in [17] further gave two novel ILC schemes based on an iteratively moving average operator. In [18], an ILC method designed by modified iteration average operator was developed for nonuniform trial lengths, while the randomness of its pass lengths is described by the recursive interval Gaussian distribution. Also, the limitation of prior probability distribution information in the control design phase was removed in [19] to propose a switching system approach with convergence properties. In addition, a deterministic model that needs no specific statistics was built to solve ILC problem with nonuniform trial lengths in [20], where a persistent full-learning assumption that there should exist a desired lengths in some bounded successive trial intervals is required. Recently, a new structure of ILC update laws was proposed in [21] based on the analysis tool of composite energy functions, which differed much from the existing approaches using contraction mapping analysis.

Existing methods for ILC problems with randomly varying trial lengths only require the gain parameters of the design to be within a certain range. Although the design freedom of gain parameters is achieved by these methods, the optimal choice cannot be obtained theoretically, which leads to lower convergence rate and non-monotonic convergence. In addition, these methods have not embedded the system constraints into their design. To

DOI: 10.1002/acs

compensate these issues, the successive projection method was first developed in [22, 23] to handle constrained ILC problems. This method utilizes abstract Hilbert space sets to formulate design requirements, and then solves a sequence of optimization problems via successive projection. In this sense, algorithms with great convergence properties are obtained, which can be used to solve problems with system constraints. On the basis of successive projection, the work in [24,25] respectively designed ILC algorithms for both discrete-time and continuous-time systems with generalized ILC problems, and provided well-defined convergence of the algorithms. In view of the great generality of successive projection in formulation and convergence properties, it can be utilized to analyze this specific class of ILC problems.

This paper aims at applying the successive projection framework to solve the ILC problems with randomly varying trial lengths. It first formulates mathematical notations as well as the problem definition in a rigorous manner, and then fully employs the successive projection scheme to propose an iterative implementational algorithm to solve this specific class of ILC problem. Convergence properties of this algorithm are also proved to guarantee the achievement of the control design objectives under mild conditions. Furthermore, this algorithm is further extended to embed an input signal constraint as an extra design objective, and the convergence properties can be proved in a similar way. The performance of the two algorithms are verified by a numerical simulation model replicating the working environment of a gantry robot, and the test results reveal their effectiveness.

The main contributions of this paper are summarized as follows:

- The successive projection framework is first used to solve ILC problems with non-uniform trial lengths, under which an optimal ILC design for problems with randomly varying trial lengths is obtained with the theoretical proof of convergence properties.
- Under the successive projection design, the probability distribution information of non-uniform trial lengths is utilized to improve ILC algorithm performance.
- Input constraints are embedded as extra design objective, under which convergence properties with non-uniform trial lengths are also proved by using successive projection.

The structure of this paper is organized as follows. The problem formulation is first addressed in Section 2. Section 3 introduces an algorithm for problems with randomly varying trial lengths under the framework of successive projection. Section 4 presents the extended scenarios of the algorithm with input constraints. Simulation verifications are shown in Section 5, and the conclusions are given in Section 6.

The main notations used in this paper are listed: $E\{\cdot\}$ and $P\{\cdot\}$ denote the the mathematical expectation and the probability of an event, respectively. $\mathbb{N}$ denotes the set of natural number and $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote the sets of n-dimensional real vectors and $n\times m$ real matrices, respectively. $l_2^m[a,b]$ denotes the space of $\mathbb{R}^m$ valued Lebesgue square-summable sequences defined on an interval $[a,b]$. The superscript T denotes the transpose and $\mathbf{0}$ denotes zero vector with appropriate dimensions. $\boldsymbol{P}_S(x)$ denotes the projection of x to the set S in some Hilbert space. $|\cdot|$ and $\langle\cdot\rangle$ respectively denote the absolute value and the inner product. $\mathbb{X}\times\mathbb{Y}$ denote the Cartesian product of two spaces $\mathbb{X}$ and $\mathbb{Y}$. Other notations will be introduced as needed in the following paper.

DOI: 10.1002/acs

2. PROBLEM FORMULATION

This section first introduces the system dynamics as well as the mathematical notations, and then formulates the definition of the exact ILC problem with randomly varying trial lengths.

2.1. System Dynamics

Consider a linear time-invariant discrete-time system with a state space form

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

where the subscript $k \in \mathbb{N}$ is the trial number index; t is the time index, $t \in [0,\ N_d]$ and N_d is the desired trial length. Note that $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}^\ell$ and $y_k(t) \in \mathbb{R}^m$ are the state, input and output of the system (1) respectively. A, B and C are system matrices with appropriate dimensions, and CB is full-rank. $y_d(t)$ is defined as the desired output trajectory. Initial state satisfies $E\{x_k(0)\} = x_0$, where x_0 is the identical expectation initial state.

For the system model (1) of the k-th trial, reformulate it to a lifted system framework, which can be rewritten into the following operator form

$$\boldsymbol{y}_k = G\boldsymbol{u}_k + \boldsymbol{d}_k, \tag{2}$$

where G and $\boldsymbol{d}_k$ represent the system model and the effect of the initial conditions respectively, i.e.

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

$$\boldsymbol{d}_k = \begin{bmatrix} (CA)^T & (CA^2)^T & \cdots & (CA^{N_d})^T \end{bmatrix}^{\mathrm{T}} x_k(0). \tag{4}$$

The input $\boldsymbol{u}_k \in l_2^\ell[0,\ N_d-1]$ and output $\boldsymbol{y}_k \in l_2^m[1,\ N_d]$ of the lifted system are denoted as follows:

$$\boldsymbol{u}_k = \left[u_k^T(0), u_k^T(1), \ldots, u_k^T(N_d-1)\right]^T, \tag{5}$$

$$\boldsymbol{y}_k = \left[y_k^T(1), y_k^T(2), \ldots, y_k^T(N_d)\right]^T. \tag{6}$$

The input Hilbert space $l_2^\ell[0,\ N_d-1]$ and output Hilbert space $l_2^m[1,\ N_d]$ are defined with inner products and associated induced norms

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_R = \boldsymbol{u}^T R \boldsymbol{v}, \quad \|\boldsymbol{u}\|_R = \sqrt{\langle \boldsymbol{u}, \boldsymbol{u} \rangle_R}, \tag{7}$$

$$\langle \boldsymbol{y}, \boldsymbol{z} \rangle_Q = \boldsymbol{y}^T Q \boldsymbol{z}, \quad \|\boldsymbol{y}\|_Q = \sqrt{\langle \boldsymbol{y}, \boldsymbol{y} \rangle_Q}, \tag{8}$$

DOI: 10.1002/acs

where $R \in \mathbb{R}^{l \cdot N_d \times l \cdot N_d}$ and $Q \in \mathbb{R}^{m \cdot N_d \times m \cdot N_d}$ are real positive definite weight matrices. The desired output $\boldsymbol{y}_d \in l_2^m[1,\ N_d]$ is denoted as

$$\boldsymbol{y}_d = \left[y_d^T(1), y_d^T(2), \ldots, y_d^T(N_d)\right]^T. \tag{9}$$

2.2. Modified ILC Problem Definition

In classical ILC, a design postulate claims that every trial has an identical trial length. However, there exist a certain range of systems, whose actual trial lengths may randomly vary from trial to trial. Denote N_k as the actual trial length of the k-th trial and N_- and N_+ as the minimum and maximum values of the actual trial lengths respectively. In practice, the maximum actual trial length is considered as the desired trial length, which means $N_d = N_+$. Then, the actual trial length varies randomly within $\{N_-, N_- + 1, \cdots, N_d\}$. So there will be $\tau = N_d - N_- + 1$ possible trial lengths in total. Let the probability of the trial length $N_-, N_- + 1, \cdots, N_d$ to be $p_1, p_2, \cdots, p_\tau$. Obviously, $p_i > 0$, for $1 \le i \le \tau$, and there exists

$$\sum_{i=1}^{\tau} p_i = 1. \tag{10}$$

When the actual trial length is less than the desired trial length in a trial, the output of the trial at $t \in [N_k + 1,\ N_d]$ is somehow missing, which means the complete tracking error is unavailable to compute the input signal for the next trial while using classical ILC. In this case, we shall append zero signal values onto the missing time instances to give a complete modified tracking error defined as

$$e_k(t) = \begin{cases} y_d(t) - y_k(t), & 1 \le t \le N_k, \\ 0, & N_k + 1 \le t \le N_d. \end{cases} \tag{11}$$

The modified tracking error of the lifted system framework is denoted as follows:

$$\boldsymbol{e}_k = \left[\overbrace{\underbrace{e_k^T(1), \cdots, e_k^T(N_k)}_{N_k}, 0, \cdots, 0}^{N_d}\right]^T. \tag{12}$$

Note that when $N_k < N_d$, $\boldsymbol{e}_k \neq \boldsymbol{y}_d - \boldsymbol{y}_k$. To eliminate the inequality, we further introduce the following random matrix

$$M_k = \begin{bmatrix} I_{N_k} \otimes I_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{(N_d - N_k)} \otimes I_m \end{bmatrix}, \quad N_- \le N_k \le N_d, \tag{13}$$

where I_l and $\mathbf{0}_l$ denote unit matrix and zero matrix with dimension of $l \times l$, and $\otimes$ denotes Kronecker product. Then the equality can be obtained as

$$\boldsymbol{e}_k = M_k \left(\boldsymbol{y}_d - \boldsymbol{y}_k\right) \in l_2^m[1,\ N_d]. \tag{14}$$

DOI: 10.1002/acs

To improve the learning efficiency, some termination strategies should be set appropriately for MIMO systems with nonuniform trial lengths in practice. In this paper, a simple termination strategy that all outputs end simultaneously when one of the outputs terminates early is adopted. Actually, this strategy matches some actual situations. For example in [12], if gantry crane stops when its load approaches an obstacle in x direction, the entire task must be terminated for all outputs, i.e., the error is not considered after time index N_k.

In order to calculate the mathematical expectation of the random matrix, Bernoulli binary random variable $\gamma_k(t)$ is introduced to represent whether the output occurs at the time t at the k-th trial. Denote $p(t)$ to be the probability of the output occurrence at time t, i.e.

$$p(t) = P\{(\gamma_k(t) = 1)\} = \begin{cases} 1, & 1 \le t \le N_- - 1, \\ \sum\limits_{i=t-N_-+1}^{\tau} p_i, & N_- \le t \le N_d. \end{cases} \tag{15}$$

The mathematical expectation of the random variables $\gamma_k(t)$ can be calculated as

$$E\{\gamma_k(t)\} = P\{\gamma_k(t) = 1\} \times 1 + P\{\gamma_k(t) = 0\} \times 0 = p(t), \tag{16}$$

which gives rise to

$$\begin{aligned} \bar{M} \triangleq E\{M_k\} &= diag\left\{\overbrace{1, 1, \cdots, 1,}^{N_- - 1} E\{\gamma_k(N_-)\}, \cdots, E\{\gamma_k(N_d)\}\right\} \otimes I_m \\ &= diag\left\{\overbrace{1, 1, \cdots, 1,}^{N_- - 1} p(N_-), \cdots, p(N_d)\right\} \otimes I_m. \end{aligned} \tag{17}$$

After defining the trial length variable, a random model is hence built to describe the system dynamics of ILC tasks with varying trial lengths. Then, the corresponding ILC design problem is defined as follows:

Definition 1
The **ILC design problem with randomly varying trial lengths** aims at designing an ILC update law

$$\boldsymbol{u}_{k+1} = f(\boldsymbol{u}_k, \boldsymbol{e}_k), \tag{18}$$

to update the input signal using previous trial's input and tracking error, which guarantees the modified tracking error converges to zero as $k \to \infty$ along the trials in the sense of mathematical expectation, i.e.

$$\lim_{k \to \infty} \|E\{\boldsymbol{e}_k\}\| = 0. \tag{19}$$

Definition 1 describes the problem to be discussed in this paper with simple and clear mathematical expression, which provides the necessary theoretical basis for the control algorithm design in the following sections.

DOI: 10.1002/acs

3. ILC DESIGN USING SUCCESSIVE PROJECTION

In this section, an iterative algorithm is designed to solve the ILC problem in Definition 1 under the successive projection framework. Also, both the implementation instructions as well as the convergence properties of the algorithm are discussed.

3.1. Successive Projection Interpretation

The design objective of the ILC problem in Definition 1 can be interpreted as: to iteratively find an optimal input $\boldsymbol{u}_\infty^*$ to make the mathematical expectation of tracking error converge to zero. This statement is equivalent to iteratively finding a point $(\mathbf{0},\ \boldsymbol{u}_\infty^*)$ in the intersection of the two following convex sets:

$$S_1 = \{(\underline{\boldsymbol{e}}, \boldsymbol{u}) \in H : \underline{\boldsymbol{e}} = E\{M\,(\boldsymbol{y}_d - \boldsymbol{y})\}, \boldsymbol{y} = G\boldsymbol{u} + \boldsymbol{d}\}\,, \tag{20}$$

$$S_2 = \{(\underline{\boldsymbol{e}}, \boldsymbol{u}) \in H : \underline{\boldsymbol{e}} = \mathbf{0}\}\,, \tag{21}$$

where S_1 and S_2 represent the requirements on system dynamic and tracking objectives respectively and $\underline{\boldsymbol{e}}$ represents the mathematical expectation of modified tracking error. M denotes the random matrix, whose definition is the same as the right side of (13). Note that H is the Hilbert space defined by

$$H = l_2^{\ell}\,[1,\ N_d] \times l_2^m[0,\ N_d - 1], \tag{22}$$

whose inner product and associated induced norm are listed as follows:

$$\langle (\underline{\boldsymbol{e}}, \boldsymbol{u})\,, (\underline{\boldsymbol{e}}, \boldsymbol{v}) \rangle_{\{Q,R\}} = \underline{\boldsymbol{e}}^T Q \underline{\boldsymbol{z}} + \boldsymbol{u}^T R \boldsymbol{v}, \tag{23}$$

$$\|(\underline{\boldsymbol{e}}, \boldsymbol{u})\|_{\{Q,R\}} = \sqrt{\langle (\underline{\boldsymbol{e}}, \boldsymbol{u})\,, (\underline{\boldsymbol{e}}, \boldsymbol{u}) \rangle_{\{Q,R\}}}, \tag{24}$$

Remark 1
In order to apply the framework of successive projection for solving problem with nonuniform trial lengths, the sets that represent both system dynamic and tracking objectives are modified by utilizing mathematical expectation. Therefore, the convergence results in this paper are all achieved in the random sense in spite of the deterministic formulation of successive projection method.

To make ILC problem solvable, the next assumption is made in this paper.

Assumption 1
The two sets S_1 and S_2 represented by (20) and (21) have intersection region in the Hilbert space H, i.e. $S_1 \cap S_2 \neq \emptyset$.

Remark 2
Assumption 1 is a sufficient and necessary condition for problem in Definition 1 being solvable, which guarantees the tracking task is achievable. However, Assumption 1 does not always holds in practice, which will be discussed in Section 4.

DOI: 10.1002/acs

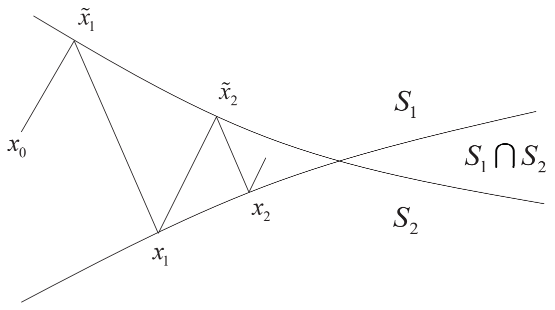

Figure 1. The implementation instructions of the successive projection method.

Since the two sets intersects with each other, there must exist a point $x^* = (\mathbf{0},\ \boldsymbol{u}_\infty^*) \in S_1 \cap S_2$ and the ILC problem has a solution.

To solve the above problem, the successive projection method developed in [26] can be adopted. The update scheme of this method is illustrated in Fig. 1, which repetitively projects the point to the position on the two sets with minimum distance. To apply this method, Lemma 1 is needed for later convergence performance analysis.

Lemma 1

Let S_1 and S_2 be two closed convex sets in a Hilbert space X. Define projection operators as

$$\boldsymbol{P}_{S_1}(x) = \arg\min_{\hat{x} \in S_1} \|\hat{x} - x\|_X^2\,, \tag{25}$$

$$\boldsymbol{P}_{S_2}(x) = \arg\min_{\hat{x} \in S_2} \|\hat{x} - x\|_X^2\,, \tag{26}$$

Let $\tilde{x}_k \in S_1$ and $x_k \in S_2$, then for $x_0 \in X$, project it successively using

$$\tilde{x}_{k+1} = \boldsymbol{P}_{S_1}(x_k), \quad x_{k+1} = \boldsymbol{P}_{S_2}(\tilde{x}_{k+1}), \quad k \geq 0, \tag{27}$$

then the monotonic convergence condition is achieved

$$\|x_{k+1} - \tilde{x}_{k+1}\|_X^2 \leq \|\tilde{x}_{k+1} - x_k\|_X^2 \leq \|x_k - \tilde{x}_k\|_X^2\,. \tag{28}$$

If $S_1 \cap S_2 \neq \emptyset$, for any $x \in S_1 \cap S_2$ and $k \geq 0$, the following inequality is satisfied

$$\|x - x_{k+1}\|_X^2 \leq \|x - \tilde{x}_{k+1}\|_X^2 \leq \|x - x_k\|_X^2 \tag{29}$$

and there always exists an integer N such that for

$$\inf_{\tilde{x}_k \in S_1} \|x - \tilde{x}_k\| < \varepsilon \quad \inf_{x_k \in S_2} \|x - x_k\| < \varepsilon,\ k \geq N \tag{30}$$

for any scalar $\varepsilon > 0$.

Proof

See [26] for the detailed proof. □

DOI: 10.1002/acs

Using Lemma 1, an algorithm is developed to solve the ILC problem in Definition 1 as follows:

Algorithm 1

Given system dynamics (1), any initial input $\boldsymbol{u}_0$ and the corresponding tracking error, then an input sequence $\{\boldsymbol{u}_k\}_{k\geq 0}$ for the ILC design problem in Definition 1 can be generated by the ILC update law

$$\boldsymbol{u}_{k+1} = \arg\min_{\hat{\boldsymbol{u}}} \left\{ \left\| \bar{M}(\boldsymbol{y}_d - G\hat{\boldsymbol{u}} - \boldsymbol{d}_d) \right\|_Q^2 + \|\hat{\boldsymbol{u}} - \boldsymbol{u}_k\|_R^2 \right\}, \tag{31}$$

where $\boldsymbol{d}_d$ is the expectation of the initial conditions and is denoted as

$$\boldsymbol{d}_d = \left[\begin{array}{cccc} (CA)^T & \left(CA^2\right)^T & \cdots & \left(CA^{N_d}\right)^T \end{array} \right]^{\mathrm{T}} x_d(0). \tag{32}$$

Proposition 1

The input sequence generated by Algorithm 1 iteratively solves the ILC design problem in Definition 1.

Proof

Since $X = H$ and the two convex sets are defined by (20) and (21), the successive projection method is applied to solve the equivalent problem according to Lemma 1. In this sense, it follows that $\tilde{x} = (\underline{\tilde{\boldsymbol{e}}}, \tilde{\boldsymbol{u}}) \in S_1$ and $x = (\boldsymbol{0}, \boldsymbol{u}) \in S_2$. Then, the projection operator $\boldsymbol{P}_{S_1}$ is computed as

$$\begin{aligned} \boldsymbol{P}_{S_1}(x) &= \arg\min_{\hat{x}\in S_1} \|\hat{x} - x\|_H^2 \\ &= \arg\min_{(\underline{\hat{\boldsymbol{e}}}, \hat{\boldsymbol{u}})\in H} \|(\underline{\hat{\boldsymbol{e}}}, \hat{\boldsymbol{u}}) - (\boldsymbol{0}, \boldsymbol{u})\|_{\{Q,R\}}^2 \\ &= \arg\min_{(\underline{\hat{\boldsymbol{e}}}, \hat{\boldsymbol{u}})\in H} \left\{ \|\underline{\hat{\boldsymbol{e}}} - \boldsymbol{0}\|_Q^2 + \|\hat{\boldsymbol{u}} - \boldsymbol{u}\|_R^2 \right\} \\ &= \arg\min_{\hat{\boldsymbol{u}}} \left\{ \left\| \bar{M}(\boldsymbol{y}_d - G\hat{\boldsymbol{u}} - \boldsymbol{d}_d) \right\|_Q^2 + \|\hat{\boldsymbol{u}} - \boldsymbol{u}\|_R^2 \right\}, \end{aligned} \tag{33}$$

which is an optimization problem, and its solution is $\hat{\boldsymbol{u}} = \tilde{\boldsymbol{u}}^*$, where $\tilde{\boldsymbol{u}}^*$ is the right hand side part of (33), then we have

$$\boldsymbol{P}_{S_1}(x) = (\bar{M}(\boldsymbol{y}_d - G\tilde{\boldsymbol{u}}^* - \boldsymbol{d}_d), \tilde{\boldsymbol{u}}^{\boldsymbol{*}}). \tag{34}$$

Similarly, the projection operator $\boldsymbol{P}_{S_2}$ gives rise to

$$\begin{aligned} \boldsymbol{P}_{S_2}(\tilde{x}) &= \arg\min_{\hat{x}\in S_2} \|\hat{x} - \tilde{x}\|_H^2 \\ &= \arg\min_{(\boldsymbol{0}, \hat{\boldsymbol{u}})\in H} \|(\boldsymbol{0}, \hat{\boldsymbol{u}}) - (\underline{\tilde{\boldsymbol{e}}}, \tilde{\boldsymbol{u}})\|_{\{Q,R\}}^2 \\ &= \arg\min_{(\boldsymbol{0}, \hat{\boldsymbol{u}})\in H} \left\{ \|\boldsymbol{0} - \underline{\tilde{\boldsymbol{e}}}\|_Q^2 + \|\hat{\boldsymbol{u}} - \tilde{\boldsymbol{u}}\|_R^2 \right\}. \end{aligned} \tag{35}$$

DOI: 10.1002/acs

Note that the solution of the optimization problem (35) can be simply taken as $\hat{\boldsymbol{u}} = \tilde{\boldsymbol{u}}$, so we have

$$\boldsymbol{P}_{S_2}(x) = (\boldsymbol{0}, \tilde{\boldsymbol{u}}). \tag{36}$$

From Lemma 1, given an initial point $x_0 = (\boldsymbol{0}, \boldsymbol{u}_0) \in S_2$, the successive projection yield the ILC input update law (31) along the trials. □

When solving ILC problems with randomly varying trial lengths in practice, Algorithm 1 can be easily implemented without any difficulties.

3.2. Implementation Instructions of the Update Law

To obtain the straightforward form of the ILC update law (31), the quadratic programming (QP) problem on its right hand side should be solved. Therefore, according to the inner product of Hilbert space (23) and related induced norm (24), a performance index function is defined as follows:

$$\boldsymbol{J}_{k+1}\left(\boldsymbol{u}_{k+1}\right) = \left\|E\left\{\boldsymbol{e}_{k+1}\right\}\right\|_Q^2 + \left\|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\right\|_R^2. \tag{37}$$

Then, the solution to obtain the ILC update law is shown below.

Theorem 1
The update law (31) has a feedforward solution

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + L\boldsymbol{e}_k, \tag{38}$$

where $L = \left(G^TKG + R\right)^{-1} G^T \bar{M}^T Q$ is the learning operator and $K = E\left\{M_k^T Q M_k\right\}$. In addition, the inequality

$$\left\|I - Q\bar{M}GL - \left(Q\bar{M}GL\right)^T + (GL)^T KGL + L^T RL\right\| \leq \|Q\| \tag{39}$$

is satisfied for any R and Q.

Proof
Since the mathematical expectations of random matrices at these trials are the same, substitute (2) and (14) into the performance index function (37) to derive the solution

$$\left(G^TKG + R\right) E\left\{\boldsymbol{u}_{k+1}\right\} = \left(G^TKG + R\right) E\left\{\boldsymbol{u}_k\right\} + G^T E\left\{M_k^T\right\} QE\left\{\boldsymbol{e}_k\right\} + G^T K E\left\{\boldsymbol{d}_k - \boldsymbol{d}_{k+1}\right\}. \tag{40}$$

Since $E\{x_k(0)\} = x_0$, then

$$E\left\{\boldsymbol{d}_k - \boldsymbol{d}_{k+1}\right\} = \boldsymbol{d}_d - \boldsymbol{d}_d = \boldsymbol{0}. \tag{41}$$

Since the matrix $(G^TKG + R)$ is invertible, substitute (41) into (40) to obtain the feedforward solution (38).

Furthermore, note that the above solution yields

$$\boldsymbol{J}_{k+1}\left(\boldsymbol{u}_{k+1}\right) \leq \boldsymbol{J}_{k+1}\left(\boldsymbol{u}_k\right), \tag{42}$$

which is equivalent to

$$\|E\{\boldsymbol{e}_{k+1}\}\|_Q^2 + \|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\|_R^2 \le \|E\{\boldsymbol{e}_k\}\|_Q^2 . \tag{43}$$

Then, substitute (38) into the left side of (43) to give

$$E\left\{\boldsymbol{e}_k^T\left[(I - M_kGL)^T Q (I - M_kGL) + L^T RL\right]\boldsymbol{e}_k\right\} \le E\left\{\boldsymbol{e}_k^T Q\boldsymbol{e}_k\right\}. \tag{44}$$

Taking the 2-norm $\|\cdot\|$ on both sides of (44) to give

$$\left\|E\left\{(I - M_kGL)^T Q (I - M_kGL) + L^T RL\right\}\right\| \le \|Q\|, \tag{45}$$

which gives rise to (39). □

Remark 3
There is no strict rule on the selection of the weight matrices Q and R, but some comments are provided in [27]. As usual, both increasing the value of Q and reducing R accelerate the algorithms convergence, while the robustness may become poor simultaneously. Also, (39) should be satisfied in this paper.

While deriving the feedforward solution (38), a property is exploited such that the mathematical expectations of random matrices at different trials are the same. To obtain the value of K, we follow the same way as shown in (17). Therefore, when $Q = qI$, M_k is a diagonal matrix, then we have

$$K = E\left\{M_k^T QM_k\right\} = qI \cdot diag\left\{\overbrace{1, \cdots, 1,}^{N_- -1} E\left\{{\gamma_k}^2(N_-)\right\}, \cdots, E\left\{{\gamma_k}^2(N_d)\right\}\right\} \otimes I_m, \tag{46}$$

where q is a non-negative scalar. Noting that when elements on the diagonal in M_k is 1, for $t \in [N_- + 1,\ N_d]$, the corresponding elements in M_k^T is 1, necessarily. Then, calculate the value of $E\left\{{\gamma_k}^2(t)\right\}$ in (46) as like (16), such that

$$E\left\{{\gamma_k}^2(t)\right\} = P\left\{{\gamma_k}^2(t) = 1\right\} \times 1 + P\left\{{\gamma_k}^2(t) = 0\right\} \times 0 = p(t), \tag{47}$$

which gives rise to

$$K{=}qI \cdot diag\left\{\overbrace{1, 1, \cdots, 1,}^{N_- -1} p(N_-), \cdots, p(N_d)\right\} \otimes I_m = q\bar{M}. \tag{48}$$

In addition, it is noted that for any trials in systems with nonuniform trial lengths, the probabilities of the output occurrence at different time instants are not all the same, and these probabilities will monotonically reduce as time goes forward, i.e.,

$$P\{\gamma_k(t) = 1\} > P\{\gamma_k(t+1) = 1\},\ t \in [N_-,\ N_d - 1]. \tag{49}$$

DOI: 10.1002/acs

Therefore, it can be considered that we add weights along the time axis of each trial according to the essential property of the varying trial lengths when $\bar{M}$ as well as K are introduced in the update law (38), and thus a better performance can be achieved. It is also noted from (49) that if there exists an output at the later time instant, there must exists an output at the previous time instant, but the opposite is not necessarily true. Then, for any $a, b \in [1,\ N_d]$, we have

$$P\left\{\gamma_k(a)\gamma_k(b)=1\right\}=P\left\{\gamma_k(a)=1\right\}=p\left(a\right),\ a>b. \tag{50}$$

Based on the property (50), to show the better performance of the proposed algorithm in the sense of random, the variance of the tracking error, which is denoted as $D\left\{\boldsymbol{e}_{k+1}\right\}$, can be employed. It follows that

$$\begin{aligned} D\left\{\boldsymbol{e}_{k+1}\right\}=&E\left\{\left(\boldsymbol{e}_{k+1}-E\left\{\boldsymbol{e}_{k+1}\right\}\right)\left(\boldsymbol{e}_{k+1}-E\left\{\boldsymbol{e}_{k+1}\right\}\right)^T\right\} \\ =&E\left\{\beta\right\}Z-\bar{M}Z\bar{M}^T, \end{aligned} \tag{51}$$

where

$$Z=(\boldsymbol{y}_d-G\boldsymbol{u}_{\boldsymbol{k+1}}-\boldsymbol{d}_d)(\boldsymbol{y}_d-G\boldsymbol{u}_{\boldsymbol{k+1}}-\boldsymbol{d}_d)^T,$$

$$E\left\{\beta\right\}=\begin{bmatrix} p(1) & p(2) & p(3) & \cdots & p(N_d) \\ p(2) & p(2) & p(3) & \cdots & p(N_d) \\ p(3) & p(3) & p(3) & \cdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ p(N_d) & & \cdots & & p(N_d) \end{bmatrix}\otimes I_m.$$

With (51), further effectiveness of the proposed algorithm can be checked, which is shown in Section 5.

3.3. Convergence Properties Analysis

According to Assumption 1, the two sets S_1 and S_2 intersect at a point $x^*=(\boldsymbol{0},\ \boldsymbol{u}_\infty^*)$ in the Hilbert space H. The convergence properties of Algorithm 1 are explained as below.

Theorem 2

If $S_1\cap S_2\neq\emptyset$, Algorithm 1 achieves monotonic convergence of tracking error in the sense of mathematical expectation,

$$\left\|E\left\{\boldsymbol{e}_{k+1}\right\}\right\|\leq\left\|E\left\{\boldsymbol{e}_k\right\}\right\|, \tag{52}$$

and

$$\lim_{k\to\infty}\boldsymbol{u}_k=\boldsymbol{u}_\infty^*,\quad \lim_{k\to\infty}\left\|E\left\{\boldsymbol{e}_k\right\}\right\|=0. \tag{53}$$

Proof

As both S_1 and S_2 are finite-dimensional closed convex sets in the Hilbert space H, from (30)

DOI: 10.1002/acs

in Lemma 1, we have

$$\lim_{k\to\infty}\left\{\|E\{0-\boldsymbol{e}_k\}\|_Q^2+\|\boldsymbol{u}_\infty^*-\boldsymbol{u}_k\|_R^2\right\}=0,$$
$$\lim_{k\to\infty}\left\{\|E\{0-\tilde{\boldsymbol{e}}_k\}\|_Q^2+\|\boldsymbol{u}_\infty^*-\tilde{\boldsymbol{u}}_k\|_R^2\right\}=0, \tag{54}$$

which gives rise to

$$\lim_{k\to\infty}\boldsymbol{u}_k=\lim_{k\to\infty}\tilde{\boldsymbol{u}}_k=\boldsymbol{u}_\infty^*,\ \lim_{k\to\infty}\|E\{\boldsymbol{e}_k\}\|=\lim_{k\to\infty}\|E\{\tilde{\boldsymbol{e}}_k\}\|=0. \tag{55}$$

Therefore, $\{\tilde{x}_k=(E\{\tilde{\boldsymbol{e}}_k\},\tilde{\boldsymbol{u}}_k)\}_{k\geq 0}$ and $\{x_k=(E\{\boldsymbol{e}_k\},\boldsymbol{u}_k)\}_{k\geq 0}$ both converge to $x^*=(\mathbf{0},\boldsymbol{u}_\infty^*)$. Furthermore, there exists

$$\boldsymbol{J}_{k+1}(\boldsymbol{u}_k)=\|E\{\tilde{\boldsymbol{e}}_k\}\|_Q^2\geq\boldsymbol{J}_{k+1}(\boldsymbol{u}_{k+1})=\|E\{\boldsymbol{e}_{k+1}\}\|_Q^2+\|\boldsymbol{u}_{k+1}-\boldsymbol{u}_k\|_R^2\geq\|E\{\boldsymbol{e}_{k+1}\}\|_Q^2, \tag{56}$$

which proves monotonic convergence properties. □

Theorem 2 theoretically guarantees the convergence properties of Algorithm 1 in terms of tracking error, which promotes its potential practical application.

4. EXTENDED SCENARIOS WITH INPUT CONSTRAINTS

When the input signal is constrained, Assumption 1 may not generally true in practice. However, the successive projection method can be still utilized to solve the ILC problem in Definition 1. Therefore, an extended ILC algorithm is specially designed to handle input constraints, whose convergence properties are also proved.

4.1. Input Constraint Forms

To ensure the safety in an actual production process or achieve extra performance requirements, certain constraints on the input signals are required. According to engineering characteristics, the input constraint set $\boldsymbol{\Omega}$ is a convex set on plenty of occasions. Some exemplary forms of the input constraints are listed as below.

- *Input saturation constraint*

$$\boldsymbol{\Omega}=\{\boldsymbol{u}\in l_2[0,N_d-1]:|u(t)|\leq Z(t),0\leq t\leq N_d-1\}, \tag{57}$$

- *Input energy constraint*

$$\boldsymbol{\Omega}=\left\{\boldsymbol{u}\in l_2[0,N_d-1]:\sum_{t=0}^{N_d-1}u^T(t)u(t)\leq\sum_{t=0}^{N_d-1}Z(t)\right\}, \tag{58}$$

DOI: 10.1002/acs

• *Input rate constraint*

$$\boldsymbol{\Omega} = \{\boldsymbol{u} \in l_2[0, N_d-1] : |\Delta u(t)| \leq Z(t), 0 \leq t \leq N_d - 1\}, \tag{59}$$

where $Z(t)$ is a non-negative scalar for $0 \leq t \leq N_d - 1$ and $\Delta u(t) = u(t) - u(t-1)$ with $\Delta u(0) = u(0)$.

Input constraints include but are not limited to the above three forms, in which saturation constraint form is one of the most common input constraints in practice, which ensures the system operates within a safe region. Therefore, the input saturation constraint is considered in latter numerical simulation section.

4.2. ILC Algorithm Design

When the input signal is constrained, the quadratic programming problem with randomly varying trial lengths and input constraints is indeed a constrained QP problem as follows:

$$\boldsymbol{u}_{k+1} = \arg\min_{\hat{\boldsymbol{u}} \in \boldsymbol{\Omega}} \|E\{M(\boldsymbol{y}_d - G\hat{\boldsymbol{u}} - \boldsymbol{d})\}\|_Q^2 + \|\hat{\boldsymbol{u}} - \boldsymbol{u}_k\|_R^2, \tag{60}$$

which is hard to solve directly. In this case, the successive projection method can be still used to design the following algorithm under input constraints, which is relatively easier to implement in practice.

Algorithm 2
Given system dynamics (1), any initial input $u_0 \in \boldsymbol{\Omega}$ and the corresponding tracking error, an input sequence $\{\boldsymbol{u}_k\}_{k \geq 0}$ for the ILC design problem in Definition 1 can be generated by the ILC update law

$$\tilde{\boldsymbol{u}}_{k+1} = \boldsymbol{u}_k + L\boldsymbol{e}_k \tag{61}$$

followed by the input projection

$$\boldsymbol{u}_{k+1} = \arg\min_{\hat{\boldsymbol{u}} \in \boldsymbol{\Omega}} \|\hat{\boldsymbol{u}} - \tilde{\boldsymbol{u}}_{k+1}\|_R^2, \tag{62}$$

where L is the same as that in (38).

Proposition 2
The input sequence generated by Algorithm 2 iteratively solves the ILC design problem in Definition 1 with input constraints.

Proof
According to Lemma 1, the functionality of Algorithm 2 can be still explained by successive projection. The sets S_1 and S_2 are now defined by

$$S_1 = \{(\underline{\boldsymbol{e}}, \boldsymbol{u}) \in H : \underline{\boldsymbol{e}} = E\{M(\boldsymbol{y}_d - \boldsymbol{y})\}, \boldsymbol{y} = G\boldsymbol{u} + \boldsymbol{d}\}, \tag{63}$$

$$S_2 = \{(\underline{\boldsymbol{e}}, \boldsymbol{u}) \in H : \underline{\boldsymbol{e}} = \boldsymbol{0}, \boldsymbol{u} \in \boldsymbol{\Omega}\}. \tag{64}$$

Similar to the proof of Algorithm 1, the solution of $\boldsymbol{P}_{S_1}$ is

$$\boldsymbol{P}_{S_1}(x) = (E\{M(\boldsymbol{y}_d - G\tilde{\boldsymbol{u}}^* - \boldsymbol{d})\}, \tilde{\boldsymbol{u}}^*), \tag{65}$$

where $\tilde{\boldsymbol{u}}^* = \boldsymbol{u} + LM(\boldsymbol{y}_d - G\boldsymbol{u} - \boldsymbol{d})$ is obtained from (38). The solution of $\boldsymbol{P}_{S_2}$ is

$$\boldsymbol{P}_{S_2}(x) = (\boldsymbol{0}, \boldsymbol{u}^*), \tag{66}$$

where $\boldsymbol{u}^* = \arg\min_{\hat{\boldsymbol{u}}\in\Omega} \|\hat{\boldsymbol{u}} - \tilde{\boldsymbol{u}}\|_R^2$ is the input projection on $\boldsymbol{\Omega}$.

Then, according to Lemma 1, given an initial point $x_0 = (\boldsymbol{0}, \boldsymbol{u}_0) \in S_2$, the input sequence $\{\boldsymbol{u}_k\}_{k\geq 0}$ updated by (61) and (62) using successive projection can solve the ILC problem in Definition 1 with input constraints. □

The successive projection method can incorporate the input constraints into the algorithm design perfectly, which is a benefit in solving tracking problems with constraints.

Remark 4

It is difficult to solve the constrained QP problem, which may cause online computational problems. Algorithm 2 converts the constrained QP problem into an unconstrained QP problem and an input projection step through the method of successive projection, which are both non-trivial to solve. When the input constraint set $\boldsymbol{\Omega}$ is in form of saturation constraint (57), the solution of (62) can be simply computed as follows:

$$u_{k+1}(t) = \begin{cases} Z(t), & \tilde{u}_{k+1}(t) > Z(t), \\ \tilde{u}_{k+1}(t), & |\tilde{u}_{k+1}(t)| \leq Z(t), \\ -Z(t), & \tilde{u}_{k+1}(t) < -Z(t). \end{cases} \tag{67}$$

4.3. Convergence Properties Analysis

The convergence properties of Algorithm 2 are shown in the theorem below.

Theorem 3

If $S_1 \cap S_2 \neq \emptyset$, Algorithm 2 achieves convergence of the input signal and the tracking error in mathematical expectation, i.e.

$$\lim_{k\to\infty} \boldsymbol{u}_k = \boldsymbol{u}_\infty^*, \quad \lim_{k\to\infty} \|E\{\boldsymbol{e}_k\}\| = 0. \tag{68}$$

Furthermore, the monotonically convergence can be achieved with respect to the performance index defined by

$$\boldsymbol{J}_k^e = \|E\{(I - M_k GL)\boldsymbol{e}_k\}\|_Q^2 + \|L\boldsymbol{e}_k\|_R^2. \tag{69}$$

Proof

The proof of (68) follows from the proof of Theorem 2 and is hence omitted here. Note that $\|x_k - \tilde{x}_k\|_H^2$ in (28) is the minimum distance from x_k to the set S_1, i.e.

$$\|x_k - \tilde{x}_k\|_H^2 = \min_{\hat{\boldsymbol{u}}} \left\{ \|E\{M(\boldsymbol{y}_d - G\hat{\boldsymbol{u}} - \boldsymbol{d})\}\|_Q^2 + \|\hat{\boldsymbol{u}} - \boldsymbol{u}_k\|_R^2 \right\}. \tag{70}$$

DOI: 10.1002/acs

Substitute $\boldsymbol{u}^* = \boldsymbol{u}_k + \boldsymbol{L}\boldsymbol{e}_k$ into the above equation to give

$$\boldsymbol{J}_k^e = \|x_k - \tilde{x}_k\|_H^2 = \|E\left\{(I - M_k GL)\,\boldsymbol{e}_k\right\}\|_Q^2 + \|\boldsymbol{L}\boldsymbol{e}_k\|_R^2\,, \tag{71}$$

and

$$\boldsymbol{J}_{k+1}^e = \|x_{k+1} - \tilde{x}_{k+1}\|_H^2 = \|E\left\{(I - M_{k+1} GL)\,\boldsymbol{e}_{k+1}\right\}\|_Q^2 + \|\boldsymbol{L}\boldsymbol{e}_{k+1}\|_R^2\,. \tag{72}$$

According to (28), there exists $\boldsymbol{J}_k^e \geq \boldsymbol{J}_{k+1}^e$, and the monotonic convergence of the performance index $\boldsymbol{J}_k^e$ is achieved. □

Remark 5

Theorem 3 obtains a specific form of monotonic convergence of the weighted error norm as shown in (69). Moreover, without input constraints, Theorem 3 collapses Theorem 2 that achieves the monotonic convergence of tracking error in the sense of mathematical expectation as in (52).

When analyzing the convergence of Algorithm 1, the situation when $S_1 \cap S_2 = \emptyset$ also needs to be considered. This is because that the input constraints may contradict with the tracking design objective and there does not exist a single possible plan under the constraints to perform desired tracking, which means the Assumption 1 may not hold. Another lemma is introduced for convergence analysis when Assumption 1 does not hold.

Lemma 2

Let S_1 and S_2 also be two closed convex sets in a Hilbert space X. For $x_0 \in X$, let $\tilde{x}_k \in S_1$ and $x_k \in S_2$, project it successively using (27). If $S_1 \cap S_2 = \emptyset$, then the distance between the two sets S_1 and S_2 converges to the minimum distance $d\left(S_1, S_2\right)$ between the two sets defined by

$$d\left(S_1, S_2\right) = \min_{\tilde{x} \in S_1, x \in S_2} \|\tilde{x} - x\|_X^2\,. \tag{73}$$

Proof

Also see [26] for the detailed proof. □

Under Lemma 2, the corresponding results are shown in the following theorem.

Theorem 4

If $S_1 \cap S_2 = \emptyset$, Algorithm 2 makes the tracking error converge to a bound in the sense of mathematical expectation and the monotonic convergence with respect to the performance index (69) is also achieved.

Proof

Using Lemma 2, define $r_1 = (\underline{\tilde{\boldsymbol{e}}}, \tilde{\boldsymbol{u}}) \in S_1, r_2 = (\boldsymbol{0}, \boldsymbol{u}_s^*) \in S_2$ as the two end points of the line segment when the two sets take the minimum distance, and this is also the solution to the following optimization problem

$$(r_1, r_2) = \arg \min_{\tilde{x} \in S_1, x \in S_2} \|\tilde{x} - x\|_H^2\,, \tag{74}$$

DOI: 10.1002/acs

which is equivalent to

$$(\boldsymbol{u}, \boldsymbol{u}_s^*) = \arg \min_{\boldsymbol{u}\in\Omega, \tilde{\boldsymbol{u}}} \left\{ \| E \left\{ M(\boldsymbol{y}_d - G\tilde{\boldsymbol{u}} - \boldsymbol{d}) \right\} \|_Q^2 + \| \tilde{\boldsymbol{u}} - \boldsymbol{u} \|_R^2 \right\}. \tag{75}$$

Therefore, the optimal solution with input constraints is

$$\boldsymbol{u}_s^* = \arg \min_{\boldsymbol{u}\in\Omega} \left\{ \min_{\tilde{\boldsymbol{u}}} \| E \left\{ M \left(\boldsymbol{y}_d - G\tilde{\boldsymbol{u}} - \boldsymbol{d} \right) \right\} \|_Q^2 + \| \tilde{\boldsymbol{u}} - \boldsymbol{u} \|_R^2 \right\}. \tag{76}$$

The optimal solution of the inner minimization problem in (76) is

$$\tilde{\boldsymbol{u}} = \boldsymbol{u} + LM \left(\boldsymbol{y}_d - G\boldsymbol{u} - \boldsymbol{d} \right). \tag{77}$$

Substitute (77) into (76) to give

$$\boldsymbol{u}_s^* = \arg \min_{\boldsymbol{u}\in\Omega} \left\{ \| E \left\{ (I - M_k GL) \, \boldsymbol{e}_k \right\} \|_Q^2 + \| L\boldsymbol{e}_k \|_R^2 \right\} = \arg \min_{\boldsymbol{u}\in\Omega} \boldsymbol{J}_k^e \tag{78}$$

The weights $(I - M_k GL)$ and L before $\boldsymbol{e}_k$ in (69) are both invertible, so the performance index (69) is strictly convex. Due to the constraint set $\boldsymbol{\Omega}$ is convex, (78) has unique solution. So we have

$$\lim_{k\to\infty} \| E \left\{ \boldsymbol{e}_k \right\} \| = \| E \left\{ M_k \left(\boldsymbol{y}_d - G\boldsymbol{u}_s^* - \boldsymbol{d}_d \right) \right\} \| = \left\| \bar{M} \left(\boldsymbol{y}_d - G\boldsymbol{u}_s^* - \boldsymbol{d}_d \right) \right\| \triangleq a, \tag{79}$$

where a is a constant. Therefore, the tracking error converges to a bound in the sense of mathematical expectation. The proof of the monotonic convergence properties of (69) is similar to that of Theorem 3. □

By using the idea of successive projection, whether Assumption 1 holds or not, the algorithm tries its best to reduce the distance between the two sets, which makes the tracking error in the sense of mathematical expectation converge asymptotically.

5. NUMERICAL SIMULATION VERIFICATIONS

The proposed algorithms are tested on a numerical simulation model to perform a control task with physical meanings. The results illustrate the effectiveness and feasibility of these algorithms, and comparisons are made with respect to classical ILC algorithm to show their advantages.

5.1. Control Task Specifications

A numerical simulation model is employed in this section to check the algorithm performance. This model replicates the working environment of the vertical moving axis of a three-axis gantry robot. In order to avoid collision between actuator and frame, some output terminal conditions should also be taken into account. For example, when the output exceeds the given range or region, this task must end earlier before reaching the expected actual trial length, which may result in the random variance of the trial lengths. The transfer function of this

Prepared using acsauth.cls
DOI: 10.1002/acs

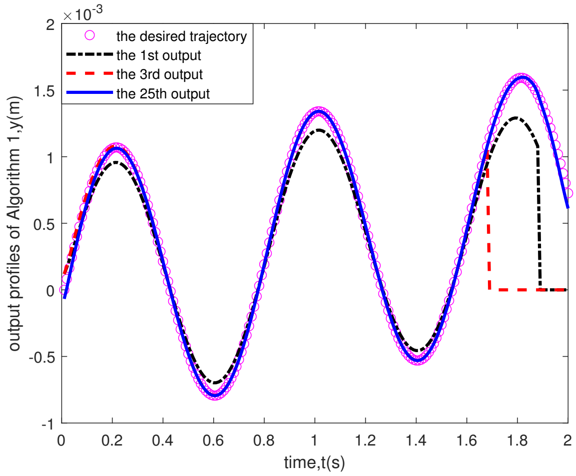

Figure 2. The reference trajectory and output trajectories of Algorithm 1.

model is

$$G_z(s) = \frac{15.8869\,(s + 850.3)}{s\left(s^2 + 707.6s + 3.377 \times 10^5\right)}, \tag{80}$$

which together with a closed-loop gain 300 yield the state-space matrices in (1) as

$$\begin{aligned} A &= \begin{bmatrix} 0.0214 & 0.0451 & 0.0124 \\ -0.0515 & -0.0497 & -0.1771 \\ 0.0916 & 0.1202 & 0.9081 \end{bmatrix}, \ B = \begin{bmatrix} -5.03 \times 10^{-5} \\ 7.16 \times 10^{-4} \\ 3.71 \times 10^{-4} \end{bmatrix}, \\ C &= \begin{bmatrix} 0 & 0.0621 & 0.8245 \end{bmatrix}, \end{aligned} \tag{81}$$

after discretization using a zero-order holder at sample time $T_s = 0.01s$. Assume the actual trial length of the repetitive process is $T = 2s$, which means the total number of sample points at each trial process is 200, i.e. $N_d = 200$. When performing a tracking task, the gantry robot may move up and down. Therefore, for the vertical axis, the desired reference trajectory is defined as

$$y_d(t) = 0.001\left[\sin\left(\frac{\pi t}{100}\right) + \sin\left(\frac{\pi t}{10}\right) + \sin\left(\frac{5\pi t}{2}\right)\right]. \tag{82}$$

Moreover, set initial state as a random variables with probability $P\left[x_k(0) = x^1\right] = P\left[x_k(0) = x^2\right] = P\left[x_k(0) = x^3\right] = 1/3$, where $x^1 = [0, 0, 0.0001]^T$, $x^1 = [0, 0, 0]^T$ and $x^1 = [0, 0, -0.0001]^T$ and it satisfies $E\{x_k(0)\} = x_d(0) = [0, 0, 0]^T$. Let the trial length vary from 160 to 200 with discrete uniform distribution, which means $p_i = 1/41$. Without loss of generality, set $u_0 = 0$.

5.2. Simulation Results

Choose $Q = 10I$ and $R = 0.0001I$, and perform Algorithm 1 with a total number of 25 trials with the last trial length being the desired length for better observation. The output trajectories are plotted in Fig. 2 for the first few trials and the final trial, and the reference trajectory is also plotted in the same figure for evaluation. It is obvious that the system trial length

DOI: 10.1002/acs

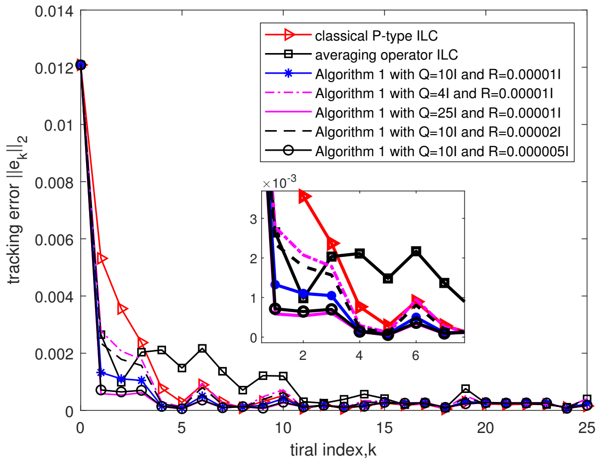

Figure 3. 2-norm of tracking errors along the trials without input constraints.

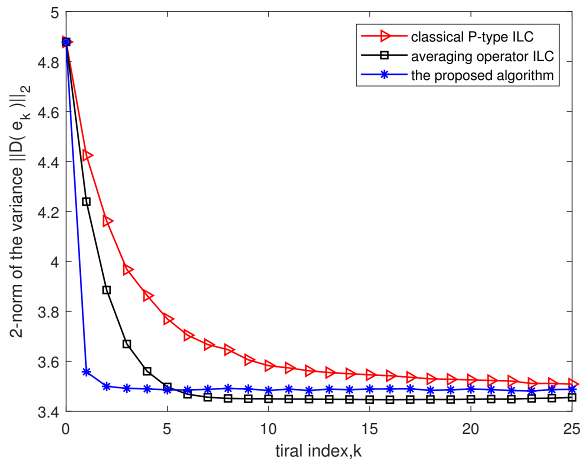

Figure 4. 2-norm of variances with respect to the tracking errors.

varies randomly but the output can track the reference trajectory after 25 trials. Moreover, the tracking errors of the system along the trials are shown in Fig. 3. Recall that the 2-norm of tracking error is proved to converge monotonically in the sense of mathematical expectation in Theorem 2, while the profiles in Fig. 3 can not show this stochastic property. For comparisons, the classical P-type ILC algorithm in [19] and averaging operator ILC in [15] for the same problem are simulated and plotted in Fig. 3. The learning gains of both comparative methods are designed as causal gain mentioned in [19]. These comparisons reveal the advantages of Algorithm 1 in convergence speed. In addition, the comparisons of different choice with respect to weight matrices are also represented in Fig. 3, which is magnified clearly inside. Increase the value of weight matrix Q and retain R, faster reduction of errors is obtained. In contrast, reduce the value of weight matrix R and retain Q, the convergence of Algorithm 1 is accelerated. From the simple comparison, we can verify what is mentioned in Remark 3. Besides, 2-norm of variances with respect to the tracking error sequences for the proposed algorithm and

DOI: 10.1002/acs

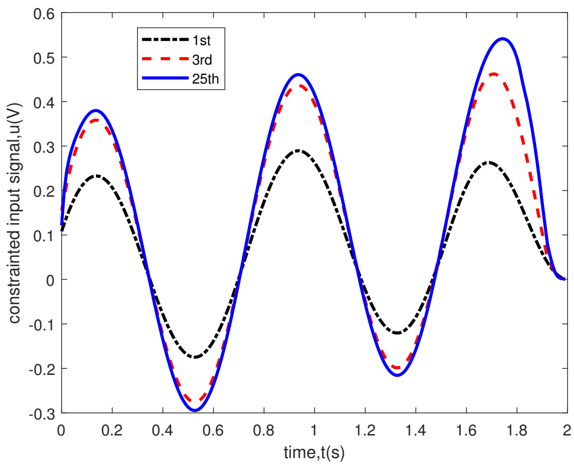

Figure 5. Input signals of Algorithm 2 when Assumption 1 holds.

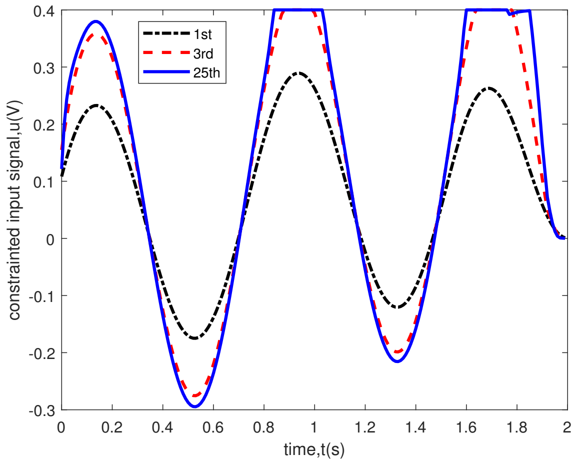

Figure 6. Input signals of Algorithm 2 when Assumption 1 does not hold.

comparative methods are presented in Fig. 4. Compared with classical P-type ILC, methods with operation of averaging, including both the proposed algorithm and the averaging operator ILC, can achieve lower variances. It means that smaller fluctuation is obtained when with nonuniform trial lengths. Accordingly, while the convergent variance of the averaging operator ILC is smaller than that of the proposed algorithm, faster convergence speed can be achieved in optimal ILC design.

Furthermore, consider the situation with input constraints whose form is taken as the saturated form shown in (67), and choose $Q = 20I$ and $R = 0.00001I$. Set $|u(t)| \leq 0.8, t = 1, 2, \cdots, N_d$ and $|u(t)| \leq 0.4, t = 1, 2, \cdots, N_d$ representing the situation when Assumption 1 holds and does not hold respectively. Algorithm 2 is hence conduced for 25 trials for these two cases, whose tracking performance is similar to the above unconstrained case. The corresponding input signals for the first few trials and the final trial are shown in Fig. 5 and Fig. 6. These results confirm the constraint handling ability of this algorithm.

DOI: 10.1002/acs

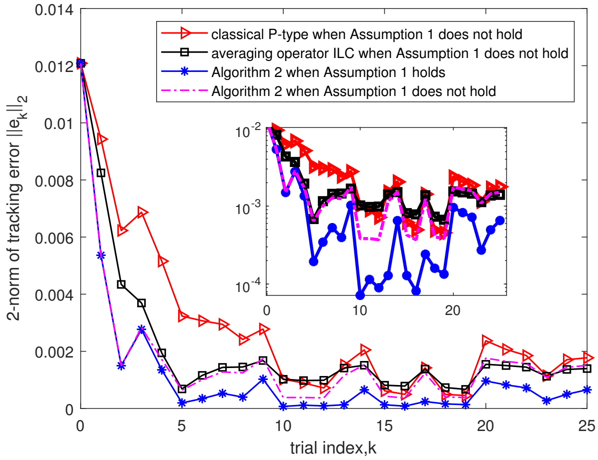

Figure 7. 2-norm of tracking errors under input constraints.

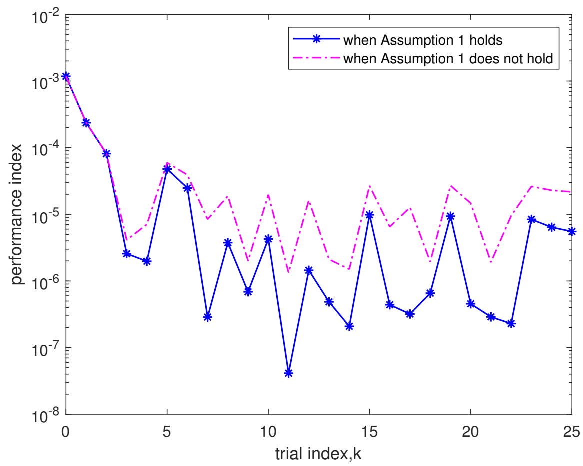

Figure 8. Performance index of Algorithm 2 under input constraints.

It is shown in Fig. 7 that when the input constraint applies, the tracking error can still converge when Assumption 1 holds or not and the convergence speed is still faster than other two comparative methods. It is interesting to see that the convergence bound a of Algorithm 2 seems to be smaller than that of P-type ILC when Assumption 1 does not hold and this phenomenon is worth studying. The actual value of the performance index (69) at these two cases is shown in Fig. 8 under logarithmic coordinates, which demonstrates the convergence of this value. However, the convergence is not monotonic, which seems to differ from what have been proved before. This is because of the fact that actual value of the performance index (69) differs from its exception value.

DOI: 10.1002/acs

6. CONCLUSION AND FUTURE WORK

The successive projection method is utilized to address the repetitive tracking problems with randomly varying trial lengths. With the mathematical formulation of an specific ILC problem, two algorithms are designed via successive projection method to solve this problem for the situations that input is constrained or not. The implementation instructions as well as the convergence properties tracking error in the sense of mathematical expectation are shown in detail. A numerical simulation case study is performed to show the effectiveness and feasibility of the proposed algorithms on a gantry robot based working environment. Also, comparison is made with classical ILC algorithm to reveal its advantages.

For future work, the proposed algorithms will be implemented on an experimental test platform to check its practical performance. Also, when the systems with randomly varying trial lengths only track specific points or the systems may have different time scales, the modification and extension of these algorithms needs to be taken into consideration. At last, the robust performance against model uncertainty of these algorithms needs a rigorous analysis in further study.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Nos.61773181, 61203092,62103293), 111 Project (B12018), the Fundamental Research Funds for the Central Universities (No.JUSRP51733B), the Natural Science Foundation of Jiangsu Province (BK20210709), the Serbian Ministry of Education, Science and Technological Development (No.451-03-9/2021-14/200108), the National Science Centre in Poland (No.2020/39/B/ST7/01487) and Suzhou Municipal Science and Technology Bureau (SYG202138).

REFERENCES

1. Arimoto S, Kawamura S, Miyazaki F. Bettering operation of robots by learning. *Journal of Robotic Systems*. 1984;1(2):123-140.
2. Liu ZJ, Liu JK, He W. An adaptive iterative learning algorithm for boundary control of a flexible manipulator. *International Journal of Adaptive Control and Signal Processing*. 2017;31(6):903-916.
3. Zhao YM, Lin Y, Xi F, Guo S. Calibration-based iterative learning control for path tracking of industrial robots. *IEEE Transactions on Industrial Electronics*. 2015;62(5):2921-2929.
4. Jin X. Fault-tolerant iterative learning control for mobile robots non-repetitive trajectory tracking with output constraints. *Automatica*. 2018;94:63-71.
5. Gao F, Yang Y, Shao C. Robust iterative learning control with applications to injection molding process. *Chemical Engineering Science*. 2001;56(24):7025-7034.
6. Freeman CT. Control System Design for Electrical Stimulation in Upper Limb Rehabilitation. *Springer International Publishing*. 2016.
7. Tao HF, Paszke W, Rogers E, Yang HZ, Galkowski K. Iterative learning fault-tolerant control for differential time-delay batch processes in finite frequency domains. *Journal of Process Control*. 2017;56:112-128.

8. Bristow DA, Tharayil M, Alleyne AG. A survey of iterative learning control. *IEEE Control Systems Magazine*. 2006;26(3):96-114.
9. Moore KL. A non-standard iterative learning control approach to tracking periodic signals in discrete-time nonlinear systems. *International Journal of Control*. 2000;73(10):955-967.
10. Shen D, Li XF. A survey on iterative learning control with randomly varying trial lengths: model, synthesis, and convergence analysis.*Annual Reviews in Control*. 2019;48:89-102.
11. Seel T, Schauer T, Raisch J. Iterative learning control for variable pass length systems. *IFAC Proceedings Volumes*. 2011;44(1):4880-4885.
12. Guth M, Seel T, Raisch J. Iterative learning control with variable pass length applied to trajectory tracking on a crane with output constraints. *52nd IEEE Conference on Decision and Control*, Florence,Italy,2013:6676-6681.
13. Seel T, Schauer T, Raisch J. Monotonic convergence of iterative learning control systems with variable pass length. *International Journal of Control*. 2017;90(3):393-406.
14. Li XF, Xu JX, Huang DQ. An iterative learning control approach for linear systems with randomly varying trial lengths. *IEEE Transactions on Automatic Control*. 2014;59(7):1954-1960.
15. Li XF, Xu JX. Lifted system framework for learning control with different trial lengths. *International Journal of Automation and Computing*. 2015;12(3):273-280.
16. Li XF, Xu JX, Huang DQ. Iterative learning control for nonlinear dynamic systems with randomly varying trial lengths. *International Journal of Adaptive Control and Signal Processing*. 2015;29(11):1341-1353.
17. Li XF, Shen D. Two novel iterative learning control schemes for systems with randomly varying trial lengths. *Systems & Control Letters*. 2017;107:9-16.
18. Shi JT, Xu JX, Sun J, Yang YH. Iterative learning control for time-varying systems subject to variable pass lengths: application to robot manipulators. *IEEE Transactions on Industrial Electronics*. 2020;67(10):8629-8637.
19. Shen D, Zhang W, Wang YQ. On almost sure and mean square convergence of P-type ILC under randomly varying iteration lengths. *Automatica*. 2016;63:359-365.
20. Meng DY, Zhang JY. Deterministic convergence for learning control systems over iteration-dependent tracking intervals. *IEEE Transactions on Neural Networks and Learning Systems*. 2018;29(8):3885-3892.
21. Jin X. Iterative learning control for MIMO nonlinear systems with iteration-varying trial lengths using modified composite energy function analysis. *IEEE Transactions on Cybernetics*. 2020. doi:10.1109/TCYB.2020.2966625.
22. Chu B, Owens DH. Accelerated norm-optimal iterative learning control algorithms using successive projection. *International Journal of Control*. 2009;82(8):1469-1484.
23. Chu B, Owens DH. Iterative learning control for constrained linear systems. *International Journal of Control*. 2010;83(7):1394-1413.
24. Chen YY, Chu B, Freeman CT. Generalized iterative learning control using successive projection: algorithm, convergence, and experimental verification. *IEEE Transactions on Control Systems Technology*. 2020;28(6):2079-2091.
25. Chen YY, Chu B, Freeman CT, Liu YH. Generalized iterative learning control with mixed system constraints: a gantry robot based verification. *Control Engineering Practice*. 2020;95:104260.
26. Owens DH, Jones RP. Iterative solution of constrained differential/algebraic systems. *International Journal of Control*. 1978;27(6):957-974.
27. Owens DH, Freeman CT, Dinh TV. Norm-optimal iterative learning control with intermediate point weighting: theory, algorithms, and experimental evaluation. *IEEE Transactions on Control Systems Technology*. 2013;21(3):999-1007.

DOI: 10.1002/acs