# Stochastic Point-to-Point Iterative Learning Tracking Without Prior Information on System Matrices

Dong Shen, *Member, IEEE*, Jian Han, and Youqing Wang, *Senior Member, IEEE*

***Abstract*— This paper contributes to a point-to-point iterative learning control problem for stochastic systems without prior information on system matrices. The stochastic approximation technique with gradient estimation by random difference is introduced to design the update law for input. It is strictly proved that the input sequence would converge almost surely to the optimal one, which minimizes the averaged tracking performance index. An illustrative simulation shows the effectiveness of the proposed algorithm.**

***Note to Practitioners*—In many practical applications, the system would perform a given task cycle by cycle. There are also many industrial processes operating periodically. In a word, repeatability is inherent in these systems. Then, it would lead to constant improvements to the system performance, just as we are able to learn from experiences and subsequently improve our behaviors in daily lives. This is exactly the control idea in this paper. On the other hand, learning, by trials and errors, allows us to make modifications and corrections on the actions. This is why the algorithm proposed in this paper could achieve accurate tracking without system information. To be specific, the algorithm updates differently during the odd cycles and the even cycles. For the odd cycle, the control is added with a small perturbation, while for the subsequent even cycle, the control is updated by estimating the gradient with the help of trial information. The detailed algorithm steps are provided with strict analysis on the convergence and optimality properties.**

***Index Terms*— Almost sure convergence, iterative learning control (ILC), point-to-point control, stochastic approximation.**

## I. Introduction

As is well known, learning is a basic skill of human to survive in the ancient times and to improve their work and lives in the modern times. Inspired by this intuitive idea, iterative learning control (ILC) is introduced to improve the tracking performance, where the system could accomplish a given task over a fixed time interval repeatedly. As learning is drawn into the input update law, the algorithm of ILC could be simple but effective as shown in many previous studies. Some excellent surveys can be found in [1]–[3]. It is noted that reinforcement learning and/or approximate dynamic programming based control also integrates the concept of learning into control strategy [4]; however, ILC differs from these methods in that ILC pays more attention to the performance improvement along the iteration axis rather than that along the time axis.

The standard ILC usually requires the system output to track a desired objective over the whole time interval. However, in many practical applications, maybe only some points are required to be tracked accurately while the others are free. As a simple but a classic example, a basketball player shoots from a fixed position repeatedly. What the player focuses is whether the basketball hits the target, rather than whether the basketball tracks some settled trajectory. In other words, only the terminal point is considered, and this kind of ILC is termed terminal iterative learning control (TILC) [5]–[7]. As a general case, take a train or subway passing some stations into consideration. One could find that only the schedule that the train/subway arrives at each station is requested, while the running status between stations is much flexible. In this case, the point-to-point iterative learning control (P2PILC) is more suitable for the control objective, because more degrees of design freedom are allowable. Both TILC and P2PILC have been considered in the previous studies.

In [8] and [9], the point-to-point tracking problem was solved through iteratively updating the reference instead of the input profile along the iteration axis, which showed a novel way for the point-to-point control problem. The benefit was that they made a good use of the freedom of trajectory. Another promising method to deal with the point-to-point problem is directly updating the control signal based on specified tracking data, as shown in [9]. In addition, Owens *et al.* [10] presented a norm-optimal ILC solution to the continuous-time point-to-point problem and Chu *et al.* [11] showed a successive projection-based method.

For a MIMO system, the required pass points in the above literature are the whole output vectors for arbitrary given time instances, while, in practice, we may only claim part components of the output vector to satisfy constraints. This kind of point-to-point tracking problem was studied in [12] for linear systems and in [13] for nonlinear systems, respectively. Detailed formulation of such a kind of point-to-point control was given in [12], which also provided an extensive analysis on gradient descent-based ILC and Newton method-based ILC with various mixed constraints. Freeman and Dinh [14] further investigated the norm-optimal ILC for highly coupled systems. The stochastic linear system case was addressed in [15] based on the stochastic approximation technique.

Manuscript received May 17, 2016; revised July 14, 2016; accepted October 11, 2016. Date of publication December 1, 2016; date of current version January 4, 2017. This paper was recommended for publication by Associate Editor A. Pashkevich and Editor H. Ding upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 61673045, Grant 61304085 and Grant 61374099, in part by the Beijing Natural Science Foundation under Grant 4152040, and in part by the Beijing Nova Program under Grant 2011025. *(Corresponding author: Dong Shen.)*

D. Shen and Y. Wang are with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: shendong@mail.buct.edu.cn; wang.youqing@ieee.org).

J. Han was with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China. He is now with the Informatics Institute, Faculty of Science, University of Amsterdam, Amsterdam 1098XH, The Netherlands (e-mail: J.Han@uva.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2016.2617868

However, model information is required in these papers to design the learning law. In other words, the algorithms depend on the system matrices, which may reduce the applications of P2PILC. The model-dependent design condition will be relaxed in this paper, where a random difference is introduced to estimate the gradient, so that no prior information on system matrices is requested. The almost sure convergence of the proposed algorithm to an allowable set is proved and then verified by an illustrative example.

This paper introduces the Kiefer–Wolfowitz (KW) algorithm into the design of update laws for the P2PILC problem to remove the prior requirement on system matrices, and the convergence of the proposed algorithms is detailed. It is seen that both [16] and [17] also adopted the KW algorithm to solve the traditional ILC problem, that is, the whole reference is required to be tracked. Considering the essence of the point-to-point control problem, we propose the lifted form of update laws for the whole iteration differing from the time instant separated form used in [16] and [17]. In addition, the linear stochastic system is considered in this paper, which is not well addressed in previous works. It is worth pointing out that the Kalman filtering-based approach proposed in [18] and [19] is also an effective method for stochastic ILC when partial system information is available.

The rest of this paper is arranged as follows. Section II provides the problem formulations. Section III gives the ILC algorithm and the almost sure convergence result. Section IV provides an illustrative example to show the effectiveness. The conclusion is given in Section V. The detailed proof of the main theorem is put in the Appendix.

## II. Problem Formulation

In this section, the system formulation, point-to-point ILC formulation, and control objective will be given in sequence.

### A. System Formulation

Consider the following linear time-varying system:

$$\begin{aligned} x_k(t+1) &= A_t x_k(t) + B_t u_k(t) + w_k(t+1) \\ y_k(t) &= C_t x_k(t) + v_k(t) \end{aligned} \quad (1)$$

where subscript $k$ denotes the cycle number, $k = 1, 2, \ldots$, and $t$ denotes an arbitrary time in a cycle, $t \in [0, N]$. $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}^p$, and $y_k(t) \in \mathbb{R}^q$ are the system state vector, input vector, and output vector, respectively. System matrices $\mathrm{A}_t$, $\mathrm{B}_t$, and $\mathrm{C}_t$ are with appropriate dimensions. Noise signals $w_k(t)$ and $v_k(t)$ are the system noise and the measurement noise, respectively. In this paper, the noise is simply assumed to be zero-mean Gaussian white noise, i.e., with normal distribution. In addition, for different cycles and different time instances, the noise signals are uncorrelated. The initial state $x_k(0)$ is set to $x_0$.

Here, assume that the input–output coupling matrix $C_{t+1}B_t$ is of full row rank, $\forall t = 0, 1, \ldots, N-1$. One can rewrite the input and the output into a supervector form given by

$$\begin{aligned} \boldsymbol{u}_k &= [u_k^T(0), u_k^T(1), \ldots, u_k^T(N-1)]^T \in \mathbb{R}^{pN} \\ \boldsymbol{y}_k &= [y_k^T(1), y_k^T(2), \ldots, y_k^T(N)]^T \in \mathbb{R}^{qN}. \end{aligned}$$

In addition, let

$$G = \begin{bmatrix} C_1 B_0 & 0 & \cdots & 0 \\ C_2 A_1 B_0 & C_2 B_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_N \prod_{j=1}^{N-1} A_j B_0 & C_N \prod_{j=2}^{N-1} A_j B_1 & \cdots & C_N B_{N-1} \end{bmatrix}$$

where $\prod_{m=i}^{j} A_m = A_j A_{j-1} \ldots A_i$, $j \geq i$ and $\prod_{m=i}^{j} A_m = I$, $j < i$, and it is obvious that $G \in \mathbb{R}^{qN \times pN}$. Then, we have the following relationship between the input and the output:

$$\boldsymbol{y}_k = G\boldsymbol{u}_k + \boldsymbol{y}_0 + \epsilon_k$$

where $\boldsymbol{y}_0$ is the response to initial conditions, $\boldsymbol{y}_0 = [(C_1 A_0)^T \, (C_2 A_1 A_0)^T \, \ldots \, (C_N \prod_{m=0}^{N-1} A_m)^T]^T x_0$. Without loss of generality, it is assumed $\boldsymbol{y}_0 = 0$ or $x_0 = 0$. The stochastic noise term $\epsilon_k$ is expressed by

$$\epsilon_k = \begin{bmatrix} v_k(1) + C_1 w_k(1) \\ v_k(2) + C_2 w_k(2) + C_2 A_1 w_k(1) \\ \vdots \\ v_k(N) + C_N \sum_{j=0}^{N} \left( \prod_{m=j}^{N-1} A_m \right) w_k(j) \end{bmatrix}.$$

Noticing requirements on system noise $\{w_k(t)\}$ and measurement noise $\{v_k(t)\}$, it is convenient to deduce the following condition.

*A1:* The stochastic noise $\{\epsilon_k\}$ is a zero-mean Gaussian process noise with covariance $Q$, i.e., $\epsilon_k \sim N(0, Q)$.

For the standard ILC framework, the tracking objective is

$$\boldsymbol{y}_d = \left[y_d^T(1), y_d^T(2), \ldots, y_d^T(N)\right]^T \in \mathbb{R}^{qN}. \quad (2)$$

Denote standard tracking error as $\boldsymbol{e}_k = \boldsymbol{y}_d - \boldsymbol{y}_k$.

### B. Point-to-Point Problem Formulation

As we know, in many practical applications, it is not required to track the whole trajectory $\boldsymbol{y}_d$, but its subset. Here, an equivalent model of [12] is given. Suppose that only $l_j$ components of the output at time $j$ is required to track, $0 \leq l_j \leq q$, $j = 1, 2, \ldots, N$. If $l_j = 0$, it means the output at time $j$ is completely disregarded. If $l_j \neq 0$, denote the tracking components by $1 \leq n_{j,1} < n_{j,2} < \cdots < n_{j,l_j} \leq q$. Now remove all the points that do not need to be followed from the original objective $\boldsymbol{y}_d$; one would have a new reference trajectory $\boldsymbol{y}_r$ with dimension $l$, where $l = \sum_{j=1}^{N} l_j$. In other words, $\boldsymbol{y}_r$ is a condensed reference trajectory of $\boldsymbol{y}_d$ satisfying the following relationship:

$$\boldsymbol{y}_r = \Phi \boldsymbol{y}_d \quad (3)$$

where $\Phi \in \mathbb{R}^{l \times qN}$ is a matrix with its element $\Phi_{i,j} = 1$, if the $i$th dimension of $\boldsymbol{y}_r$ locates at the $j$th dimension of $\boldsymbol{y}_d$, and otherwise $\Phi_{i,j} = 0$.

In the following, it is the condensed reference trajectory $\boldsymbol{y}_r$ rather than the original one $\boldsymbol{y}_d$ that is available for ILC update laws. Moreover, by the definition of $\Phi$ and its construction process, it is obvious that $\Phi$ is of full row rank, i.e., $rank(\Phi) = l$. Considering the system formulation, for the following design and analysis, we further need a rank property of $\Phi G$ stated in the following lemma.

*Lemma 1:* The matrix $\Phi G$ is of full row rank.

*Proof:* Since $C_{t+1}B_t$ is of full row rank, the lifted matrix $G$ also is of full row rank, i.e., $rank(G) = qN$. By the definition of $\Phi$, it is evident that $rank(\Phi) = l$. By the Sylvesters rank inequality, we have $rank(\Phi) + rank(G) - qN \leq rank(\Phi G)$, that is, $rank(\Phi G) \geq l$. On the other hand, it is evident $rank(\Phi G) \leq l$. Thus, we find $rank(\Phi G) = l$. ■

### C. Control Objective

For the standard ILC without any noise in the system, the control objective is to find an input sequence, such that $\|\boldsymbol{e}_k\| \to 0$ as $k \to \infty$. The notation $\|\cdot\|$ here denotes 2-norm and this meaning will be kept through the rest of this paper. However, this control objective is not suitable for the stochastic system. Besides, since the focus of this paper falls on the required reference points, all the components of $\boldsymbol{e}_k$ may not be available in applications. Therefore, a new performance index is needed. Note that $\boldsymbol{e}_k = \boldsymbol{y}_d - \boldsymbol{y}_k$ and $\boldsymbol{y}_r = \Phi \boldsymbol{y}_d$, and it leads to

$$\Phi \boldsymbol{e}_k = \Phi(\boldsymbol{y}_d - \boldsymbol{y}_k) = \boldsymbol{y}_r - \Phi \boldsymbol{y}_k \tag{4}$$

as the real tracking information for input updating.

There is stochastic noise involved in $\boldsymbol{y}_k$, and thus, one could not expect $\|\Phi \boldsymbol{e}_k\| \to 0$ as what have been done for the deterministic system. However, one may expect that if the associated stochastic noises are eliminated, the left part would converge to zero. To be specific, denote $\eta_k \triangleq \boldsymbol{y}_r - \Phi G \boldsymbol{u}_k$, and then, one would expect $\eta_k \xrightarrow[k\to\infty]{} 0$.

Therefore, the control objective of this paper is to design the ILC update law, such that $\boldsymbol{u}_k \to J^*$, where $J^* \triangleq \{\boldsymbol{u} : \boldsymbol{y}_r - \Phi G \boldsymbol{u} = 0\}$. Here, by $\boldsymbol{u}_k \to J^*$, we mean that $\boldsymbol{u}_k$ converges to an element in $J^*$.

*Remark 1:* As shown in [12, Lemma 1], if $\Phi G$ is of full row rank, the feasible input $\boldsymbol{u}$ forms a space $J^*$ of dimension $pN - l$. In other words, $J^*$ could be formulated as $J^* = \{(\Phi G)^\dagger \boldsymbol{y}_r + \boldsymbol{u}, \boldsymbol{u} \in \text{null}(\Phi G)\}$, where $(\Phi G)^\dagger$ denotes the pseudoinverse of $\Phi G$.

*Remark 2:* It is worth pointing out that $\eta_k$ denotes the tracking error excluding the stochastic noises. Thus, $\eta_k \to 0$ means the system output may accurately track the reference $\boldsymbol{y}_r$ asymptotically if noise effects of the current cycle do not count. Since the system noises and measurement noises of the current cycle could not be prior predicted, $\eta_k \to 0$ is actually the best tracking performance.

## III. Main Results

In this section, the ILC algorithm is designed and analyzed. Since the system matrices $\{A_t\}$, $\{B_t\}$, and $\{C_t\}$ are unknown prior, one could not directly calculate the corresponding input $\boldsymbol{u}^*$, such that $\boldsymbol{y}_r = \Phi G \boldsymbol{u}^*$ or recursively generate the input sequence based on system matrices.

To generate such an input sequence without using system matrices, the KW stochastic approximation algorithm used in [16] and [17] is introduced, where the random difference is involved to estimate the gradient. To this end, we use the following vector sequence $\{\Delta_k, k = 1, 2, \ldots\}$, where $\Delta_k \in \mathbb{R}^{pN}$, $\Delta_k \triangleq [\Delta_k^1, \Delta_k^2, \ldots, \Delta_k^{pN}]$. All components $\Delta_k^j$ are mutually independent and identically distributed random variables satisfying the following conditions:

$$|\Delta_k^j| < a, \quad |\frac{1}{\Delta_k^j}| < b, \quad \mathbb{E}\frac{1}{\Delta_k^j} = 0 \quad \forall k = 1, 2, \ldots, \quad j = 1, 2, \ldots, pN.$$

It is also assumed that the sequence $\{\Delta_k^j\}$ is independent of $\{\epsilon_k\}$. Define the $pN$-dimension vector

$$\overline{\Delta}_k = \left[\frac{1}{\Delta_k^1}, \frac{1}{\Delta_k^2}, \ldots, \frac{1}{\Delta_k^{pN}}\right] \quad k = 1, 2, \ldots$$

Then, the ILC update algorithm is described as follows.

Let $\{a_k\}$, $\{c_k\}$, and $\{M_k\}$ be the sequences of real numbers satisfying the following conditions:

$$a_k > 0, \quad a_k \to 0, \quad \sum_{k=1}^{\infty} a_k = \infty \tag{5}$$

$$c_k > 0, \quad c_k \to 0, \quad \sum_{k=1}^{\infty} \left(\frac{a_k}{c_k}\right)^{1+\frac{\delta}{2}} < \infty \tag{6}$$

$$M_k > 0, \quad M_{k+1} > M_k, \quad M_k \to \infty. \tag{7}$$

The initial input $\boldsymbol{u}_0$ is simply set to be zero. Then, the algorithm updates differently during the odd cycles and the even cycles. For the odd cycle, the control is defined as follows:

$$\boldsymbol{u}_{2k+1} = \boldsymbol{u}_{2k} + c_k \Delta_k \tag{8}$$

while for the even cycle

$$\overline{\boldsymbol{u}}_{2(k+1)} = \boldsymbol{u}_{2k} - a_k \frac{\overline{\Delta}_k}{c_k}(\|\Phi \boldsymbol{e}_{2k}\|^2) \tag{9}$$

$$\boldsymbol{u}_{2(k+1)} = \overline{\boldsymbol{u}}_{2(k+1)} \mathbf{1}_{[\|\overline{\boldsymbol{u}}_{2(k+1)}\| \leq M_{\sigma_k}]} \tag{10}$$

$$\sigma_k = \sum_{i=1}^{k-1} \mathbf{1}_{[\|\overline{\boldsymbol{u}}_{2(i+1)}\| > M_{\sigma_i}]}, \quad \sigma_0 = 0 \tag{11}$$

where $\mathbf{1}_{[\text{inequality}]}$ is an indicator function meaning that it equals 1 if the inequality indicated in the bracket is fulfilled, and 0 if the inequality does not hold.

*Remark 3:* The ILC algorithm applied to system (1) with selected tracking points $\boldsymbol{y}_r$ is given by (8)–(11), where the random difference is applied to estimate the gradient, i.e., the control update direction. Thus, prior information on system matrices could be removed. In addition, an indicator function $\mathbf{1}_{[\cdot]}$ is introduced to guarantee the boundedness of the input sequence.

*Remark 4:* Here, we give some explanations on the parameters $\{a_k\}$, $\{c_k\}$, and $\{M_k\}$. $a_k$ denotes the learning step size and it serves as the conventional learning gain incorporating with the random difference. The decrease of $a_k$ is to make a suppression of the stochastic noises. $c_k$ is used to reduce the range of deviation asymptotically, which further leads to a stable learning of the gradient. $M_k$ is a technical trick to avoid the divergence of the proposed algorithm and ensure a stable improvement of the input sequence. The selection of these parameters should satisfy (5)–(7). Usually, $a_k$ and $c_k$ is given as $\varrho k^{-\tau}$, where $\varrho$ and $\tau$ are suitable positive constants and $M_k$ is selected as $2^k$ or $3^k$.

Then, the following theorem on convergence of the algorithm could be established.

*Theorem 1:* Consider system (1) and assume A1 hold, then the input sequence generated by (8)–(11) tends to $J^*$, i.e., $\boldsymbol{u}_k \to J^*$ as $k$ goes to infinity.

*Remark 5:* Though the update laws are given based on the KW algorithm, the convergence analysis is established based on the convergence results of the well-known Robbins–Monro (RM) stochastic approximation algorithm [20]–[22]. To be specific, the proof is carried out through two steps: first, we transform the proposed update laws into the RM algorithm formulation, and then, we show the almost sure convergence by verifying the convergence conditions of corresponding RM algorithms.

*Remark 6:* The differences between the odd and even cycles lie in roles that they are played in the gradient estimation process. For the odd cycle, the control is involved by a small perturbation, while for the subsequent even cycle, the control is updated by estimating the gradient with the help of trial information. The advantage of our ILC scheme is that it could estimate its gradient based on input/output information, which, therefore, is a data-based control method. In many practical systems, the accurate math model is hard to get due to the complex mechanism, time-varying environments, large scale, and other factors. Thus, the proposed data-based method is quite significant for practical applications.

*Remark 7:* Under the framework of point-to-point control, it is easy to find that there is not only one solution in $J^*$, since $\Phi G$ is of full row rank rather than full column rank. Theorem 1 ensures that the input sequence would converge to a limitation in $J^*$, but it does not guarantee that the limitations for different experiment paths would be identical. Thus, there is an interesting question: would the input sequence converge to different limitations in different experiments? The answer is no. As a matter of fact, the limitation of the input sequence is $(\Phi G)^T(\Phi G G^T \Phi^T)\boldsymbol{y}_r$. However, it is out of the scope of this note, and thus, it is omitted.

*Remark 8:* The linear stochastic system is considered in this paper. As is well known, most real-world problems involve nonlinearities, and thus, one might be interested in the extension to nonlinear systems. A possible way is making local linearization to the nonlinear system, as shown in [13]. In addition, our algorithms require no information on system matrices, and thus it is possible to deal with nonlinear systems. However, the analysis would be much complicated and we would like to address this problem in the next step.

## IV. Illustrative Simulation

Consider a LTV system with system matrices given as follows:

$$A = \begin{bmatrix} 0.5+0.1\sin(0.3t) & -0.07 & -0.26 \\ -0.03 & 0.38+0.15\sin(2\pi/t) & -0.3 \\ -0.1 & -0.13 & 0.4+0.1t^2 \end{bmatrix}$$

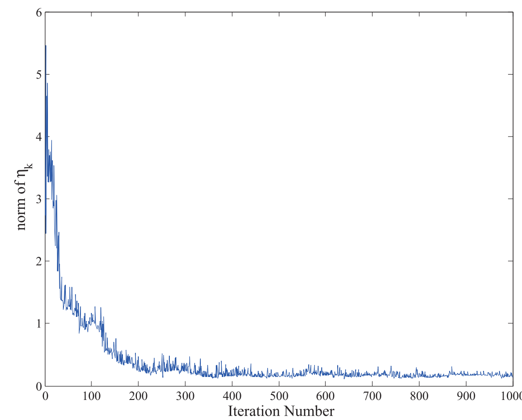


Fig. 1. Norm of the control objective: $\|\eta_k\| = \|\boldsymbol{y}_r - \Phi G \boldsymbol{u}_k\|$.

$$B = \begin{bmatrix} -1+0.1t^2 & 0.1 & 0 \\ 0 & -1.2+\cos(0.5t) & 0.4 \\ 0.2 & 0 & -1+0.08t^2 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.8-\sin(0.6t) & 0.5 & 0.3 \\ 0.4 & 1.6+0.1\sin(0.2t) & 1.2+0.15\sin(2\pi/t) \end{bmatrix}.$$

For simple illustration, let $N = 6$, and then $\boldsymbol{y}_k \in \mathbb{R}^{12}$ and $\boldsymbol{u}_k \in \mathbb{R}^{18}$. The noise $\epsilon_k$ is assumed a zero-mean Gaussian process noise with the covariance $Q = 0.05^2 I$.

Suppose the reference points are $y_d^{(2)}(1)$, $y_d^{(2)}(3)$, $y_d^{(1)}(4)$, and $y_d^{(1)}(6)$, where the superscript denotes the dimension of the output vector, which describes the general point-to-point tracking problem. That is

$$\Phi = \begin{bmatrix} 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \end{bmatrix}.$$

It is easy to verify that $\text{rank}(\Phi G) = 4$. The arbitrarily selected reference trajectory is $\boldsymbol{y}_r = [0.5\ 2\ 1.5\ 1]^T$.

The parameters for the algorithm are selected as follows. $\Delta_k^j$ is uniformly distributed on $[-1, -0.5] \cup [0.5, 1]$, $\forall k \geq 1$, $1 \leq j \leq 18$. The iteration varying sequences $\{a_k\}$ and $\{c_k\}$ choose $a_k = (1/((k+200)^{0.95}))$ and $c_k = (1/((k+1)^{0.65}))$. The expanding parameter $M_k$ is $M_k = 2^k$. The initial input $\boldsymbol{u}_0$ is simply the zero vector. Then, the algorithm runs 1000 iterations following (8)–(11).

In order to illustrate the almost sure convergence of the algorithm, the norm of the modified tracking error, i.e., $\eta_k = \boldsymbol{y}_r - \Phi G \boldsymbol{u}_k$, is shown in Fig. 1. As is shown in Section II, it suffices to demonstrate that $\eta_k \to 0$. It is shown in Fig. 1 that the error $\eta_k$ reduces rapidly. This further means that the tracking error will be caused mainly by the system noises and measurement noises of the current iteration asymptotically, while the noises cannot be eliminated by any learning algorithm. This shows the effectiveness of the algorithm from another perspective. That is, the proposed algorithm has actually achieved the best tracking performance under stochastic noises.

The whole output of the last iteration $\boldsymbol{y}_{1000}$ and the reference points $\boldsymbol{y}_r$ are shown in Fig. 2, where the solid line with cycles

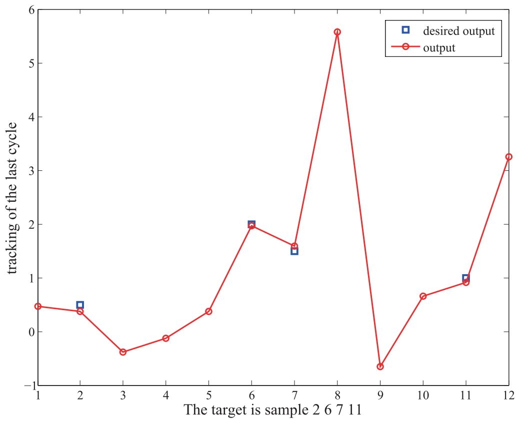


Fig. 2. Output of the last iteration $\boldsymbol{y}_{1000}$ and the reference points $\boldsymbol{y}_r$.

denotes all the components of the 12-dimension output of the last iteration and the four squares denote the reference points. The x-label 1–12 are corresponding to $y^{(1)}(1)$, $y^{(2)}(1)$, $y^{(1)}(2),\ldots$, and $y^{(2)}(6)$. Thus, the four bold squares in Fig. 2 correspond to $y_d^{(2)}(1)$, $y_d^{(2)}(3)$, $y_d^{(1)}(4)$, and $y_d^{(1)}(6)$, respectively. It can be found from the figure that the system output could track the reference points effectively under the noise environment.

However, the outputs of the unrequested points are with no consideration in the proposed algorithm. From simulations, we find that the outputs of these positions change from simulation to simulation. Therefore, the unrequested outputs leave much freedom for us to design the algorithm for seeking more optimization objectives or satisfying more constraints. It is an interesting and open topic in the stochastic point-to-point ILC problem.

## V. Conclusion

In this paper, an ILC is designed for the stochastic point-to-point control problem without prior information on system matrices. A gradient estimating method, namely, the KW stochastic approximation algorithm is introduced to design the ILC update law. The ILC algorithm is proved convergent and optimal with probability one under mild conditions. For further research, it is of interest to consider how to reduce the dimension of the ILC algorithm and improve the convergence rate. It is also of interest to consider other control objectives, such as LQG problem or energy cost constraints.

## Appendix

The following convergence theorem of stochastic approximation comes from [22].

Let $f(\cdot):\mathbb{R}^p\to\mathbb{R}^p$ and $J\triangleq\{x: f(x)=0\}$. Take a sequence of positive real numbers $M_k$ satisfying $M_{k+1}>M_k$, $M_k \xrightarrow[k\to\infty]{} \infty$. Consider the following algorithm:

$$\begin{aligned} x_{k+1} &= (x_k + a_k y_{k+1})\mathbf{1}_{[\|x_k+a_k y_{k+1}\|\le M_{\sigma_k}]} \\ &\quad + x^*\mathbf{1}_{[\|x_k+a_k y_{k+1}\|> M_{\sigma_k}]} \end{aligned} \tag{12}$$

$$y_{k+1} = f(x_k)+\varepsilon_{k+1} \tag{13}$$

$$\sigma_k = \sum_{i=1}^{k-1}\mathbf{1}_{[\|x_i+a_i y_{i+1}\|> M_{\sigma_i}]},\quad \sigma_0 = 0. \tag{14}$$

*Theorem 2:* Assume that the following H1–H4 hold.

*H1:* $a_k>0$, $a_k \xrightarrow[k\to\infty]{} 0$, and $\sum_{k=1}^{\infty} a_k=\infty$.

*H2:* There exists a continuously differentiable function $\upsilon(\cdot):\mathbb{R}^p\to R$, such that

$$\sup_{\delta\le d(x,J)\le\Delta} f^T(x)\frac{\partial \upsilon(x)}{\partial x}<0$$

for any $\Delta>\delta>0$, where $d(x,J)=\inf_y\{\|x-y\|, y\in J\}$, $\upsilon(J)$ is nowhere dense. Furthermore, there exists a constant $c_0>0$, such that $\|x^*\|<c_0$ and $\upsilon(x^*)<\inf_{\|x\|=c_0}\upsilon(x)$.

*H3:* The function $f(\cdot)$ is measurable and locally bounded.

*H4:* Along the subscripts $\{n_k\}$ of any convergent subsequence $x_{n_k}$

$$\lim_{T\to 0}\limsup_{k\to\infty}\frac{1}{T}\left\|\sum_{i=n_k}^{m(n_k,T_k)} a_i\varepsilon_{i+1}\mathbf{1}_{[\|x_i\|\le K]}\right\| = 0\quad \forall T_k\in[0,T]$$

if $K$ is sufficiently large, where $m(k,T)\triangleq\max\{m: \sum_{i=k}^m a_i\le T\}$.

Then, with any initial value $x_0$, $x_k$ defined by (12)–(14) converges to $J$ with probability one.

*Proof of Theorem 1:* Denote

$$L_k \triangleq \frac{\overline{\Delta}_k}{c_k}(\|\Phi e_{2k+1}\|^2-\|\Phi e_{2k}\|^2) \tag{15}$$

where $\Phi e_k=\boldsymbol{y}_r-\Phi G\boldsymbol{u}_k-\Phi G\epsilon_k$. Thus

$$\begin{aligned} L_k &= \frac{\overline{\Delta}_k}{c_k}(\|\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k+1}-\Phi G\epsilon_{2k+1}\|^2 \\ &\qquad -\|\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k}-\Phi G\epsilon_{2k}\|^2) \\ &= \frac{\overline{\Delta}_k}{c_k}(\|\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k+1}\|^2-\|\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k}\|^2 \\ &\qquad +\|\Phi G\epsilon_{2k+1}\|^2-\|\Phi G\epsilon_{2k}\|^2 \\ &\qquad -2(\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k+1})^T\Phi G\epsilon_{2k+1} \\ &\qquad +2(\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k})^T\Phi G\epsilon_{2k}) \\ &= \frac{\overline{\Delta}_k}{c_k}(\|\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k}-\Phi Gc_k\Delta_k\|^2 \\ &\qquad -\|\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k}\|^2+\|\Phi G\epsilon_{2k+1}\|^2-\|\Phi G\epsilon_{2k}\|^2 \\ &\qquad -2(\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k})^T\Phi G\epsilon_{2k+1} \\ &\qquad +2c_k(\Phi G\Delta_k)^T\Phi G\epsilon_{2k+1} \\ &\qquad +2(\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k})^T\Phi G\epsilon_{2k}) \\ &= \frac{\overline{\Delta}_k}{c_k}(-2c_k(\Phi G\Delta_k)^T(\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k}) \\ &\qquad +\|\Phi Gc_k\Delta_k\|^2+\|\Phi G\epsilon_{2k+1}\|^2-\|\Phi G\epsilon_{2k}\|^2 \\ &\qquad -2(\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k})^T\Phi G\epsilon_{2k+1} \\ &\qquad +2c_k(\Phi G\Delta_k)^T\Phi G\epsilon_{2k+1} \\ &\qquad +2(\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k})^T\Phi G\epsilon_{2k}) \\ &= -2(\Phi G)^T(\boldsymbol{y}_r-\Phi G\boldsymbol{u}_{2k}) \\ &\quad -\delta_k+\gamma_k+\theta_k+\alpha_{2k+1}-\alpha_{2k}-\beta_{2k+1}+\beta_{2k} \end{aligned}$$

where

$$\begin{aligned}
\delta_k &= 2\left(\overline{\Delta}_k \Delta_k^T - I\right)(\Phi G)^T(\boldsymbol{y}_r - \Phi G \boldsymbol{u}_{2k})\\
\gamma_k &= c_k \overline{\Delta}_k \|\Phi G \Delta_k\|^2\\
\theta_k &= 2\overline{\Delta}_k \Delta_k^T (\Phi G)^T \Phi G \epsilon_{2k+1}\\
\alpha_{2k+1} &= \frac{\overline{\Delta}_k}{c_k}\|\Phi G \epsilon_{2k+1}\|^2\\
\alpha_{2k} &= \frac{\overline{\Delta}_k}{c_k}\|\Phi G \epsilon_{2k}\|^2\\
\beta_{2k+1} &= 2\frac{\overline{\Delta}_k}{c_k}(\boldsymbol{y}_r - \Phi G \boldsymbol{u}_{2k})^T \Phi G \epsilon_{2k+1}\\
\beta_{2k} &= 2\frac{\overline{\Delta}_k}{c_k}(\boldsymbol{y}_r - \Phi G \boldsymbol{u}_{2k})^T \Phi G \epsilon_{2k}.
\end{aligned}$$

Set $g(x) = 2(\Phi G)^T(\boldsymbol{y}_r - \Phi G x)$ and $\xi_k = -\delta_k + \gamma_k + \theta_k + \alpha_{2k+1} - \alpha_{2k} - \beta_{2k+1} + \beta_{2k}$, and then, the algorithm (9)–(11) is

$$\begin{aligned}
\overline{\boldsymbol{u}}_{2(k+1)} &= (\boldsymbol{u}_{2k} - a_k g(\boldsymbol{u}_{2k}) - a_k \xi_k)\\
&\quad \times \mathbf{1}_{[\|\boldsymbol{u}_{2k} - a_k g(\boldsymbol{u}_{2k}) - a_k \xi_k\| \le M_{\sigma_k}]}\\
\sigma_k &= \sum_{i=1}^{k-1} \mathbf{1}_{[\|\overline{\boldsymbol{u}}_{2(i+1)}\| > M_{\sigma_i}]}, \quad \sigma_0 = 0.
\end{aligned}$$

Comparing with Theorem 2 in the Appendix, to show that $\boldsymbol{u}_{2k}$ converges to $J^*$, conditions H1–H4 should be satisfied.

H1 is fulfilled by the selection of $a_k$.

For H2, select the Lyapunov function as $v(x) = (\boldsymbol{y}_r - \Phi G x)^T \Phi G (\Phi G)^T (\boldsymbol{y}_r - \Phi G x)$, and then

$$\begin{aligned}
&\frac{\partial v(x)}{\partial x} g(x)\\
&\quad = -2(\boldsymbol{y}_r - \Phi G x)^T \Phi G (\Phi G)^T \Phi G (\Phi G)^T (\boldsymbol{y}_r - \Phi G x).
\end{aligned}$$

Noticing that $v(J^*) = 0$, it is nowhere dense. Besides, H2 is fulfilled as $x^* = 0$.

In the present case, $g(x)$ is a linear function, and thus, H3 is valid.

Thus, it only remains to verify H4.

Since $c_k \to 0$ as $k \to \infty$, one can get

$$\lim_{T\to 0} \limsup_{k\to\infty} \frac{1}{T}\left\| \sum_{i=k}^{m(k,T_k)} a_i \gamma_i \right\| = 0 \quad \forall T_k \in [0, T]. \tag{16}$$

Therefore, to check condition H4, it suffices to show

$$\begin{aligned}
&\sum_{k=1}^{\infty} a_k[-\delta_k + \theta_k + \alpha_{2k+1} - \alpha_{2k}\\
&\qquad - \beta_{2k+1} + \beta_{2k}]\mathbf{1}_{[\|\boldsymbol{u}_{2k}\| < K]} < \infty, \quad \text{a.s.}
\end{aligned} \tag{17}$$

First, check the term $a_k(\alpha_{2k+1} - \alpha_{2k})$. It is noticed

$$\begin{aligned}
&\alpha_{2k+1} - \alpha_{2k}\\
&\quad = \frac{\overline{\Delta}_k}{c_k}(\|\Phi G \epsilon_{2k+1}\|^2 - \|\Phi G \epsilon_{2k}\|^2)\\
&\quad = \frac{\overline{\Delta}_k}{c_k}\left(\mathrm{tr}\left(\Phi G \epsilon_{2k+1}\epsilon_{2k+1}^T\left(\Phi G \epsilon_{2k}\epsilon_{2k}^T (\Phi G)^T\right)\right)\right.\\
&\quad = \frac{\overline{\Delta}_k}{c_k}\left(\mathrm{tr}\left(\Phi G\left(\epsilon_{2k+1}\epsilon_{2k+1}^T - Q\right)(\Phi G)^T\right)\right.\\
&\qquad\qquad \left. - \mathrm{tr}\left(\Phi G\left(\epsilon_{2k}\epsilon_{2k}^T - Q\right)(\Phi G)^T\right)\right).
\end{aligned}$$

By assumption A1, $\{((\overline{\Delta}_k)/(c_k))\mathrm{tr}(\Phi G(\epsilon_{2k+1}\epsilon_{2k+1}^T - Q)(\Phi G)^T)\}$ and $\{((\overline{\Delta}_k)/(c_k))\mathrm{tr}(\Phi G(\epsilon_{2k}\epsilon_{2k}^T - Q)(\Phi G)^T)\}$ are the sequences of zero-mean mutually independent random vectors with bounded moments. Then, by the convergence theorem for martingale difference sequence [23]

$$\sum_{k=1}^{\infty} a_k(\alpha_{2k+1} - \alpha_{2k}) < \infty, \quad \text{a.s.} \tag{18}$$

and hence

$$\sum_{k=1}^{\infty} a_k(\alpha_{2k+1} - \alpha_{2k})\mathbf{1}_{[\|\boldsymbol{u}_{2k}\| < K]} < \infty, \quad \text{a.s.} \tag{19}$$

since $\alpha_{2k+1}$ and $\alpha_{2k}$ are independent of $\boldsymbol{u}_{2k}$.

Next, check the term $a_k(\beta_{2k+1} - \beta_{2k})$. Notice that $\boldsymbol{u}_{2k}$ is independent of $\epsilon_{2k+1}$ and $\epsilon_{2k}$, and $\Delta_k$ also is independent of $\epsilon_{2k+1}$ and $\epsilon_{2k}$; therefore, again by the convergence theorem for martingale difference sequence [23]

$$\sum_{k=1}^{\infty} a_k(\beta_{2k+1} - \beta_{2k})\mathbf{1}_{[\|\boldsymbol{u}_{2k}\| < K]} < \infty, \quad \text{a.s.} \tag{20}$$

Now, it comes to $a_k\theta_k$. Notice that $\Delta_k$ is independent of $\epsilon_{2k+1}$, and both $\Delta_k$ and $\overline{\Delta}_k$ are bounded. Besides, $\epsilon_{2k+1}$ is a martingale difference sequence. Thus

$$\sum_{k=1}^{\infty} a_k \overline{\Delta}_k \Delta_k^T (\Phi G)^T \Phi G \epsilon_{2k+1} < \infty, \quad \text{a.s.} \tag{21}$$

and hence

$$\sum_{k=1}^{\infty} a_k \theta_k \mathbf{1}_{[\|\boldsymbol{u}_{2k}\| < K]} < \infty, \quad \text{a.s.} \tag{22}$$

since $\theta_k$ is independent of $\boldsymbol{u}_{2k}$.

Finally, consider the term $a_k\delta_k$. It is noticed that

$$\overline{\Delta}_k \Delta_k^T - I = \begin{bmatrix}
0 & \frac{\Delta_k^2}{\Delta_k^1} & \frac{\Delta_k^3}{\Delta_k^1} & \cdots & \frac{\Delta_k^{pN}}{\Delta_k^1}\\
\frac{\Delta_k^1}{\Delta_k^2} & 0 & \frac{\Delta_k^3}{\Delta_k^2} & \cdots & \frac{\Delta_k^{pN}}{\Delta_k^2}\\
\frac{\Delta_k^1}{\Delta_k^3} & \frac{\Delta_k^2}{\Delta_k^3} & 0 & \cdots & \frac{\Delta_k^{pN}}{\Delta_k^3}\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
\frac{\Delta_k^1}{\Delta_k^{pN}} & \frac{\Delta_k^2}{\Delta_k^{pN}} & \frac{\Delta_k^3}{\Delta_k^{pN}} & \cdots & 0
\end{bmatrix}. \tag{23}$$

Because $\Delta_k^i$ and $\Delta_k^j$ are mutually independent, $\forall i \neq j$; thus, $\mathbb{E}(\overline{\Delta}_k \Delta_k^T - I) = 0$. Moreover, both $\Delta_k^i$ and $(1/(\Delta_k^j))$ are bounded, $\forall i, j$; thus, $\overline{\Delta}_k \Delta_k^T - I$ has a finite moment of $2 + \delta$. In addition, $\overline{\Delta}_k \Delta_k^T - I$ is independent of $\boldsymbol{u}_{2k}$. Therefore, one has

$$\sum_{k=1}^{\infty} a_k \delta_k \mathbf{1}_{[\|\boldsymbol{u}_{2k}\| < K]} < \infty, \quad \text{a.s.} \tag{24}$$

Thus, the condition H4 is verified, and the convergence theorem in the Appendix can be applied. As a result, one has

$$\boldsymbol{u}_{2k} \xrightarrow[k\to\infty]{} J^*, \quad \text{a.s.} \tag{25}$$

Since $c_k \Delta_k \xrightarrow[k\to\infty]{} 0$, from (8) and (25)

$$\boldsymbol{u}_k \xrightarrow[k\to\infty]{} J^*, \quad \text{a.s.} \tag{26}$$

This completes the proof. ■

## References


[1] H.-S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1099–1121, Nov. 2007.

[2] Y. Wang, F. Gao, and F. J. Doyle, III, "Survey on iterative learning control, repetitive control, and run-to-run control," *J. Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[3] D. Shen and Y. Wang, "Survey on stochastic iterative learning control," *J. Process Control*, vol. 24, no. 12, pp. 64–77, 2014.

[4] X. Xu, *Reinforcement Learning and Approximate Dynamic Programming*. Beijing, China: Science Press, 2010.

[5] T. Liu, D. Wang, and R. Chi, "Neural network based terminal iterative learning control for uncertain nonlinear non-affine systems," *Int. J. Adapt. Control Signal Process.*, vol. 29, no. 10, pp. 1274–1286, 2015.

[6] R. Chi, Z. Hou, S. Jin, D. Wang, and C.-J. Chien, "Enhanced data-driven optimal terminal ILC using current iteration control knowledge," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2939–2948, Nov. 2015.

[7] S. Jin, Z. Hou, and R. Chi, "Optimal terminal iterative learning control for the automatic train stop system," *Asian J. Control*, vol. 17, no. 5, pp. 1992–1999, 2015.

[8] C. T. Freeman, Z. Cai, E. Rogers, and P. L. Lewin, "Iterative learning control for multiple point-to-point tracking application," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 3, pp. 590–600, May 2011.

[9] T. D. Son, H.-S. Ahn, and K. L. Moore, "Iterative learning control in optimal tracking problems with specified data points," *Automatica*, vol. 49, no. 5, pp. 1465–1472, 2013.

[10] D. H. Owens, C. T. Freeman, and T. V. Dinh, "Norm-optimal iterative learning control with intermediate point weighting: Theory, algorithms, and experimental evaluation," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 999–1007, May 2013.

[11] B. Chu, C. T. Freeman, and D. H. Owens, "A novel design framework for point-to-point ILC using successive projection," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1156–1163, May 2015.

[12] C. T. Freeman and Y. Tan, "Iterative learning control with mixed constraints for point-to-point tracking," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 604–616, May 2013.

[13] C. T. Freeman and Y. Tan, "Point-to-point iterative learning control with mixed constraints," in *Proc. Amer. Control Conf.*, San Francisco, CA, USA, Jul. 2011, pp. 3657–3662.

[14] C. T. Freeman and T. V. Dinh, "Experimentally verified point-to-point iterative learning control for highly coupled systems," *Int. J. Adapt. Control Signal Process.*, vol. 29, no. 3, pp. 302–324, 2015.

[15] D. Shen and Y. Wang, "Iterative learning control for stochastic point-to-point tracking system," in *Proc. 12th Int. Conf. Control, Autom., Robot. Vis.*, Guangzhou, China, 2012, pp. 480–485.

[16] H. F. Chen, "Almost sure convergence of iterative learning control for stochastic systems," *Sci. China Ser. F*, vol. 46, no. 1, pp. 67–79, 2003.

[17] D. Shen and H. F. Chen, "A Kiefer–Wolfowitz algorithm based iterative learning control for Hammerstein–Wiener systems," *Asian J. Control*, vol. 14, no. 4, pp. 1070–1083, 2012.

[18] S. S. Saab, "On a discrete-time stochastic learning control algorithm," *IEEE Trans. Autom. Control*, vol. 46, no. 8, pp. 1333–1336, Aug. 2001.

[19] S. S. Saab, "Selection of the learning gain matrix of an iterative learning control algorithm in presence of measurement noise," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1761–1774, Nov. 2005.

[20] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[21] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[22] H. F. Chen, *Stochastic Approximation and Its Applications*. Dordrecht, The Netherlands: Kluwer, 2002.

[23] Y. S. Chow and H. Teicher, *Probability Theory: Independence, Interchangeability, Martingales*. New York, NY, USA: Springer, 1978.