A Novel Markov Chain Based ILC Analysis for Linear Stochastic Systems Under General Data Dropouts Environments

Dong Shen, *Member, IEEE*, and Jian-Xin Xu, *Fellow, IEEE*

***Abstract*—This technical note contributes to the convergence analysis for iterative learning control (ILC) for linear stochastic systems under general data dropout environments, i.e., data dropouts occur randomly at both the measurement and actuator sides. Data updating in the memory array is arranged in such a way that data at every time instance is updated independently, which allows successive data dropouts both in time and iteration axes. The update mechanisms for both the computed input and real input are proposed and then the update process of both inputs is shown to be a Markov chain. By virtue of Markov modeling, a new analysis method is developed to prove the convergence in both mean square and almost sure senses. An illustrative example verifies the theoretical results.**

***Index Terms*—Almost sure convergence, data dropout, iterative learning control, Markov chain, mean square convergence.**

I. Introduction

In practical industrial processes, many systems complete a given task in a finite time interval and repeat the process continuously. Then, one is interested in how the repetition could help improve the control performance. This motivates the introduction of an intelligent control strategy called iterative learning control (ILC). In ILC, the tracking information from previous iterations is taken full advantage of to gradually improve the performance from iteration to iteration. After three decades of developments, ILC has shown a distinct advantage in high-precision tracking and fewer requirements on system information [1]–[3].

Meanwhile, based on the fast development of communication technology, currently, more and more control systems are implemented in the networked mode to enhance flexibility and robustness. For example, the ILC has been applied to two-link robotic fish in [4], where the control signal is transmitted to the robot fish through wireless network. Similar examples include the formation control of satellites and unmanned aerial vehicles (UAV). In this implementation, the plant and the controller are separated and communicated with each other through wired/wireless networks. Therefore, a problem arising naturally is the data dropout phenomenon, which would damage the tracking performance. This problem motivates us to consider networked ILC under data dropout conditions and some pioneer papers have been reported [5]–[16].

Most of the reported results considered a special case that the data dropout problem only occurs at the measurement side [5]–[12]. In other words, the network at the actuator side was assumed to work well so that the generated input signal can be always transmitted to the system timely. In this case, a binary valued variable was introduced to denote the occurrence of data dropout. However, when moving to the general case that data dropout occurred at both measurement and actuator sides [13]–[16], the one-side transmission results cannot be extended directly due to the fact that the input signal generated by the controller and the one used for the system are not always identical, which is different from the case in [5]–[12]. In order to ensure a feasible compensation mechanism, additional requirements were imposed in the existing papers. To be specific, in [13], [14], the dropped data was compensated with the data one-step back within the same iteration. Consequently, a limitation arises in that the data at adjacent time instants cannot be dropped at the same iteration. In [15], [16], the dropped data was compensated with the data at the same time instant one-iteration back. However, this required that there was no simultaneous data dropout at the same time instant across any two adjacent iterations. Moreover, such requirements on data dropout might be difficult to meet in practical applications. Thus, it is of great significance to address the general and random successive data dropout problem.

This technical note aims to complete the exploration of this topic. To be specific, the general data dropout occurring at both measurement and actuator sides are considered for linear stochastic systems. The data dropout is modeled by a Bernoulli random variable without further conditions. In other words, successive data dropouts with arbitrary length are allowed in this technical note, which is seldom considered in previous papers. Because the existence of successive data dropouts, the updating mechanism proposed in this technical note only uses the available data while guaranteeing strong convergence properties. Moreover, for simplicity, the P-type update law is adopted in this technical note to illustrate the learning convergence, though learning controller does not restrict to P-type. Furthermore, the main difficulty in proving the convergence for the general data dropout problem lies in the random asynchronism between the input updating at the controller and the system. In this technical note, we first analyze the sample path behavior of the update process and then reveal that the process actually is a Markov chain. This paves a novel way for convergence analysis. The ILC for Markovian switching systems was reported in [17], where the controller and mean square stability conditions were obtained for a leader-follower network based on a group of LMIs. The techniques in this technical note differ from [17] that we directly establish the mean square and almost sure convergence for stochastic systems with the classic P-type law and mild design conditions.

This technical note distinguishes from previous papers in the following novelties: 1) this technical note allows randomly successive data dropouts at both measurement and actuator sides, which makes the

Manuscript received October 6, 2016; revised December 4, 2016; accepted December 7, 2016. Date of publication December 9, 2016; date of current version October 25, 2017. This work was supported by National Natural Science Foundation of China (61673045, 61304085), Beijing Natural Science Foundation (4152040) and the China Scholarship Council (201506885019). Recommended by Associate Editor P. Bolzern.

D. Shen is with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, P. R. China (e-mail: shendong@mail.buct.edu.cn).

J.-X. Xu is with the Department of Electrical and Computer Engineering, National University of Singapore 117576, Singapore (e-mail: elexujx@nus.edu.sg).

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/TAC.2016.2638044

0018-9286 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

input signal updates at controller and plant asynchronous and has not been addressed in ILC field; 2) we formulate the control problem under data dropouts into a Markov chain, and a novel analysis framework is then provided; 3) the rigorous convergence proof both in mean square sense and almost sure sense under stochastic noises is given, which is hard to achieve by traditional analysis approaches; and 4) the results show the effectiveness and robustness of traditional P-type update law against random factors. In addition, a decreasing gain sequence is incorporated into the ILC algorithm to actively cope with the systemic and measurement noises.

The technical note is arranged as follows. Section II gives the problem formulation. Detailed behavior analysis, convergence proof, and performance discussions are provided in Section III. Section IV presents an illustrative example to verify the theoretical results. Concluding remarks are provided in Section V.

II. Problem Formulation

Consider the following linear time varying stochastic system

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

where $t = 0, 1, \cdots, N$ denotes the time index, N is the iteration length, and $k = 1, 2, \cdots,$ denotes the iteration index. $u_k(t) \in \mathbb{R}^p$, $y_k(t) \in \mathbb{R}^q$, and $x_k(t) \in \mathbb{R}^n$ are the system input, output, and state, respectively. The system matrices A_t, B_t, and C_t are with appropriate dimensions. $w_k(t)$ and $v_k(t)$ are system noises and measurement noises, respectively.

The desired reference is $y_d(t)$, which satisfies the following formulation,

$$\begin{aligned} x_d(t+1) &= A_t x_d(t) + B_t u_d(t),\\ y_d(t) &= C_t x_d(t). \end{aligned} \tag{2}$$

The following assumptions are required for further analysis.

A 1: The input-output coupling matrix $C_{t+1}B_t \in \mathbb{R}^{q\times p}$ is of full-column rank, $t = 0, \cdots, N-1$ and therefore $q \geq p$.

A 2: The initial state $x_k(0)$ is precisely reset, i.e., $x_k(0) = x_d(0)$.

Remark 1: The initial state reset is a critical issue in ILC field. The Assumption A2 is the well-known identical initialization condition (i.i.c.), which has been used in many ILC papers. Efforts have been dedicated to the relaxation of i.i.c., but they require further information on the system or additional control mechanism. However, the extension of i.i.c. is out of our scope, thus, we simply assume A2.

Denote the increasing σ-algebra $\mathcal{F}_k \triangleq \sigma\{x_j(t), w_j(t), v_j(t), 1 \leq j \leq k, t = 0, 1, \cdots, N\}$ generated by the information from the first iteration to the kth iteration. We give the following assumption on stochastic noises.

A 3: The stochastic noises $w_k(t)$ and $v_k(t)$ are independent for different time instants and they are independent with each other. For each t, $\mathbb{E}\{w_k(t)|\mathcal{F}_{k-1}\} = 0$, $\mathbb{E}\{v_k(t)|\mathcal{F}_{k-1}\} = 0$, $\sup_k \mathbb{E}\{w_k^2(t)|\mathcal{F}_{k-1}\} < \infty$, $\sup_k \mathbb{E}\{v_k^2(t)|\mathcal{F}_{k-1}\} < \infty$. Here $\mathbb{E}(\cdot)$ denotes the mathematical expectation operator.

The control objective of this technical note is to design a learning algorithm such that the generated input sequence could track the desired trajectory asymptotically along iteration axis under data dropouts and stochastic noises conditions. Because of the existence of stochastic noises, the actual output could not precisely track the desired trajectory. Thus, our objective is to minimize the following performance index

$$V_t = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \|y_d(t) - y_k(t)\|^2. \tag{3}$$

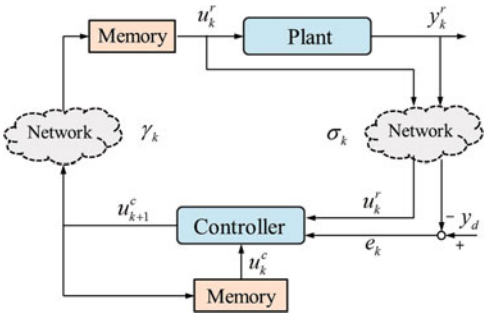

Fig. 1. Block diagram of the proposed ILC framework.

By Lemma 1 of [11], in order to minimize the above performance index, it is sufficient to show that the input sequence $u_k(t) \to u_d(t)$, $t = 0, 1, \cdots, N-1$. This is the objective of subsequent analysis. In addition, it is noticed from Lemma 1 of [11] that the minimum of the above performance index (3) is a linear combination of the upper bounds of covariance of both system and measurement noises $w_k(t)$, $v_k(t)$. In other words, the ultimate tracking performance is determined by the stochastic noises.

In this technical note, the general networked framework is considered, that is, both networks at the measurement and actuator sides would suffer random data dropouts. The data dropouts are modeled by two random variables, $\sigma_k(t)$ and $\gamma_k(t)$, subject to Bernoulli distribution. Specifically, both $\sigma_k(t)$ and $\gamma_k(t)$ equal 1 if the corresponding data are transmitted successfully, and 0 otherwise. Moreover, $\mathbb{P}(\sigma_k(t) = 1) = \overline{\sigma}$ and $\mathbb{P}(\gamma_k(t) = 1) = \overline{\gamma}$, $0 < \overline{\sigma}, \overline{\gamma} < 1$, where $\mathbb{P}(\cdot)$ denotes the probability of the indicted event. Both $\sigma_k(t)$ and $\gamma_k(t)$ are independent for different time instant t and iteration k.

The block diagram of the framework is illustrated in Fig. 1. In this framework, the update of the input follows the intermittent type. In other words, if the data are successfully transmitted at the measurement side, then the algorithm would update its input signal; while if the data are lost during transmission at the measurement side, then the algorithm would stop updating and retain the stored input signal of the previous iteration. At the actuator side, if the input signal is successfully transmitted, then the plant will use this new input signal; if the input signal is lost, then the plant will operate with the stored input signal of the previous iteration. The data dropouts at measurement and actuator sides occur independently.

In this framework, we denote the control signal generated by the learning controller, called the *computed control*, as $u_k^c(t)$, and the real control signal fed to the plant, called the *real control*, as $u_k^r(t)$. The workflow of Fig. 1 is as follows: when the system finishes one batch, all the data are transmitted back to the controller, then the controller computes the control signal for the next batch, and the computed control would then be transmitted to the plant so that the system could run for the next batch. Then the updating law in the controller is formulated as

$$\begin{aligned} u_{k+1}^c(t) = {} & \sigma_{k+1}(t)u_k^r(t) + (1-\sigma_{k+1}(t))u_k^c(t)\\ & + \sigma_{k+1}(t)a_k L_t e_k(t+1), \end{aligned} \tag{4}$$

while the actually used control signal for the $(k+1)$th iteration is

$$u_{k+1}^r(t) = \gamma_{k+1}(t)u_{k+1}^c(t) + (1-\gamma_{k+1}(t))u_k^r(t), \tag{5}$$

where $e_k(t) \triangleq y_d(t) - y_k(t)$, L_t is the learning gain matrix and a_k is a decreasing sequence to ensure zero-error tracking. The sequence $\{a_k\}$ should satisfy $a_k > 0$, $\sum_{k=1}^{\infty} a_k = \infty$, and $\sum_{k=1}^{\infty} a_k^2 < \infty$.

Remark 2: The decreasing sequence $\{a_k\}$ used in (4) is a technical means to handle the stochastic noises. If the stochastic noises are eliminated from the system, the term a_k can be removed from (4). It is well known that an appropriate decreasing gain for the correction term in updating processes is a necessary requirement to ensure convergence in the recursive computation for optimization, identification, and tracking of stochastic systems [18], [19]. This fact is also illustrated in ILC literature [2], [20], [21].

Remark 3: The random variables $\sigma_{k+1}(t)$ and $\gamma_{k+1}(t)$ are defined independently along both iteration and time axes. Thus it is apparent that successive data dropouts in time axis are allowed in this formulation. Moreover, from (4) it is noticed that if $\sigma_{k+1}(t) = 1$, i.e., the data is successfully transmitted, then the *computed control* is updated; otherwise if $\sigma_{k+1}(t) = 0$, then the *computed control* copies its corresponding value of the previous iteration. However, in the latter case, the corresponding *computed control* of the previous iteration may likewise copy the value of its previous iteration. Consequently, successive data dropouts in iteration axis are also allowed. In addition, No extra storage beyond one batch size is required by the memory array because only the latest data needs to be stored, as shown in the updating laws (4) and (5) through two random variables $\sigma_{k+1}(t)$ and $\gamma_{k+1}(t)$. In other words, at each time instance, if the input is updated, then the updated value replaces the stored data; otherwise, the stored input keeps unchanged.

III. Main Results

In this section, we first show that the update process of the computed control and the real control is a Markov chain, which paves a critical and novel way to obtain the convergence. Then the convergence in both mean square and almost sure senses are established. In addition, the convergence speed is explicitly discussed at the end of this section.

A. Markov Chain of the Input Sequence

In this subsection, we will establish the Markov chain of the input sequences. To make our idea clear, we first consider the case for an arbitrary time instant and then generalize it to the whole iteration.

Now, for arbitrary fixed time instant t, $0 \le t \le N-1$, let us consider the sample path behaviors of update algorithms (4) and (5), where a sample path means an arbitrary sequence with respect to iteration k. It is noticed that the computed control and the real control are the same whenever $\gamma_k = 1$. In the following, the sample path behavior is called "*synchronization*" if the computed control and real control are equal to each other; otherwise, it is called "*asynchronization*". Moreover, it is called a "*renewal*" if both computed control and real control are in the state of *synchronization* but different from their last *synchronization*. The following lemma shows that the sample path behavior actually is a Markov chain in terms of *synchronization* and *asynchronization*.

Lemma 1: Consider the updating laws (4) and (5). The updating of the values of $u_k^r(t)$ and $u_k^c(t)$ forms a Markov chain.

Proof: We start from the kth iteration where $u_k^r(t) = u_k^c(t)$. That is, the computed control and real control are in the state of *synchronization* at the kth iteration. Then, for the $(k+1)$th iteration, four possible outcomes exist.

1) *Case 1:* $\sigma_{k+1}(t) = 0$ and $\gamma_{k+1}(t) = 1$.
   The probability of this case is $\mathbb{P}(\sigma_{k+1}(t) = 0)\mathbb{P}(\gamma_{k+1}(t) = 1) = (1-\overline{\sigma})\overline{\gamma}$. In this case, from (4) and (5) one has $u_{k+1}^c(t) = u_k^c(t) = u_k^r(t)$, $u_{k+1}^r = u_{k+1}^c(t) = u_k^r(t)$. Thus the computed control and the real control retain the same as the kth iteration.
2) *Case 2:* $\sigma_{k+1}(t) = 0$ and $\gamma_{k+1}(t) = 0$.
   The probability of this case is $\mathbb{P}(\sigma_{k+1}(t) = 0)\mathbb{P}(\gamma_{k+1}(t) = 0) = (1-\overline{\sigma})(1-\overline{\gamma})$. In this case, it is obvious that $u_{k+1}^c(t) = u_k^c(t) = u_k^r(t)$, $u_{k+1}^r(t) = u_k^r(t)$. That is, no change in computed control nor in real control occurs.
3) *Case 3:* $\sigma_{k+1}(t) = 1$ and $\gamma_{k+1}(t) = 1$.
   The probability of this case is $\mathbb{P}(\sigma_{k+1}(t) = 1)\mathbb{P}(\gamma_{k+1}(t) = 1) = \overline{\sigma}\overline{\gamma}$. In this case, we find that $u_{k+1}^c(t) = u_k^r(t) + a_k L_t e_k(t+1)$, $u_{k+1}^r(t) = u_{k+1}^c(t) = u_k^r(t) + a_k L_t e_k(t+1)$. In other words, the computed control and the real control are updated simultaneously and are still equal to each other. In short, a *renewal* occurs.
4) *Case 4:* $\sigma_{k+1}(t) = 1$ and $\gamma_{k+1}(t) = 0$.
   The probability of this case is $\mathbb{P}(\sigma_{k+1}(t) = 1)\mathbb{P}(\gamma_{k+1}(t) = 0) = \overline{\sigma}(1-\overline{\gamma})$. In this case, only the computed control is updated, $u_{k+1}^c(t) = u_{k+1}^c(t) = u_k^r(t) + a_k L_t e_k(t+1)$, $u_{k+1}^r(t) = u_k^r(t)$. As a result, the state becomes *asynchronization*.

From the above discussions, we find that (a) the computed control $u_{k+1}^c(t)$ and the real control $u_{k+1}^r(t)$ stay in the state of *synchronization* except in the last case; and (b) a *renewal* occurs when no data dropouts happen at the measurement and the actuator sides with a probability $\overline{\sigma}\overline{\gamma}$.

Therefore, we further discuss the last case, that is, we assume that the computed control and the real control become the last case at the $(k+1)$th iteration. Then, four possible outcomes exist for the $(k+2)$th iteration with probabilities of the four outcomes being the same to cases 1–4 above.

1) *Case 1':* $\sigma_{k+2}(t) = 0$ and $\gamma_{k+2}(t) = 1$.
   In this case, the real control is updated, $u_{k+2}^c(t) = u_{k+1}^c(t) = u_k^r(t) + a_k L_t e_k(t+1)$, $u_{k+2}^r(t) = u_{k+2}^c(t) = u_k^r(t) + a_k L_t e_k(t+1)$. That is, the computed control and the real control achieve *synchronization*, and a *renewal* occurs.
2) *Case 2':* $\sigma_{k+2}(t) = 0$ and $\gamma_{k+2}(t) = 0$.
   In this case, no change happens to both the computed control and the real control, $u_{k+2}^c(t) = u_{k+1}^c(t) = u_k^r(t) + a_k L_t e_k(t+1)$, $u_{k+2}^r(t) = u_{k+1}^r(t) = u_k^r(t)$. Then the computed control and the real control are still in the state of *asynchronization*.
3) *Case 3':* $\sigma_{k+2}(t) = 1$ and $\gamma_{k+2}(t) = 1$.
   In this case, both the computed control and the real control are updated, $u_{k+2}^c(t) = u_{k+1}^r(t) + a_{k+1} L_t e_{k+1}(t+1)$, $u_{k+2}^r(t) = u_{k+2}^c(t)$. As a result, the computed control and the real control become *synchronization* again, and a *renewal* occurs.
4) *Case 4':* $\sigma_{k+2}(t) = 1$ and $\gamma_{k+2}(t) = 0$.
   In this case, only the computed control is updated, $u_{k+2}^c(t) = u_{k+1}^r(t) + a_{k+1} L_t e_{k+1}(t+1)$, $u_{k+2}^r(t) = u_{k+1}^r(t)$. However, the real control remains the same to the $(k+1)$th iteration and therefore, the computed control retains the same state to the $(k+1)$th iteration. Thus, the computed control and real control are in the state of *asynchronization*.

The analysis indicates that (a) from the *asynchronization* state, the computed control $u_{k+2}^c(t)$ and the real control $u_{k+2}^r(t)$ will either remain unchanged state or become *synchronization* again; and (b) a *renewal* occurs whenever the state changes into *synchronization*.

As a result, we can conclude that the computed control and the real control have two states, i.e., *synchronization* and *asynchronization*, respectively. Moreover, from the state of *synchronization*, the probability of the inputs retaining *synchronization* is $1-\overline{\sigma}(1-\overline{\gamma})$, while the probability of the inputs switching to *asynchronization* is $\overline{\sigma}(1-\overline{\gamma})$. From the state of *asynchronization*, the probabilities of retaining *asynchronization* and switching to *synchronization* are $1-\overline{\gamma}$ and $\overline{\gamma}$, respectively. Therefore, the two states would switch between each other following a Markov chain, as shown in Fig. 2. The proof of this lemma is completed. ■

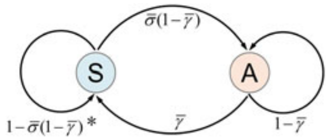
$\overline{\sigma}(1-\overline{\gamma})$
S
A
$1-\overline{\sigma}(1-\overline{\gamma})^*$
$\overline{\gamma}$
$1-\overline{\gamma}$

Fig. 2. Illustration of the Markov chain of *synchronization* and *asynchronization*. S: synchronization; A: asynchronization; $*$: there is a *renewal* with probability $\overline{\sigma\gamma}$.

Remark 4: If $\overline{\gamma} = 1$, which means there is no data dropout at the actuator side, then the matrix P is singular as the second column being zero. This implies that the computed control and real input would always be in the state of *synchronization*. This special case has been discussed in many previous papers. On the other hand, if $\overline{\sigma} = 1$, which means that there is no data dropout at the measurement side, then P is also singular as its first row coincides with the last row, and the Markov chain degrades to be a simple Bernoulli sequence. The convergence of such special case could be easy to established.

Next we verify that the sample path behaviors for the whole iteration also form a Markov chain. Lemma 1 indicates that, for arbitrary time instant t, the states of *synchronization* and *asynchronization* between $u_k^c(t)$ and $u_k^r(t)$ form a Markov chain. From the analysis steps of Lemma 1, it can be seen that the switching of such two states are only determined by the data dropout variables at both sides, i.e., $\sigma_k(t)$ and $\gamma_k(t)$. In other words, the Markov property of the sample path behavior of $u_k^c(t)$ and $u_k^r(t)$ is irrelevant with their specific values.

Moreover, notice that the random variables $\sigma_k(t)$ and $\gamma_k(t)$ modeling the data dropouts at both sides are independent for different time instants. Owing to such independence among different time instants, the combination of the N Markov chains, generated by the computed control and the real control for each time instant t, $0 \le t \le N-1$, is also a Markov chain.

Specifically speaking, let us introduce a new notation $\tau_k(t)$ to describe the states of *synchronization* and *asynchronization*. That is, we let $\tau_k(t) = 1$ if the computed control and the real control achieves *synchronization*, otherwise, $\tau_k(t) = 0$ if they achieves *asynchronization*. As we have explained above, the states of *synchronization* or *asynchronization* are determined only by the data dropout variables $\sigma_k(t)$ and $\gamma_k(t)$, therefore, $\tau_k(i)$ is independent of $\tau_k(j)$ for any $i \ne j$. Meanwhile, the evolution of $\tau_k(t)$ is a Markov chain. To show the overall behaviors of all time instants, we further introduce a vector $\varphi_k \triangleq [\tau_k(0), \cdots, \tau_k(N-1)]^T$, which is a stack of $\tau_k(t)$. Since each variable $\tau_k(t)$ is binary, the vector φ_k has 2^N possible outcomes, i.e., $[0, 0, \cdots, 0]^T$, $[1, 0, \cdots, 0]^T$, $\cdots$, $[1, 1, \cdots, 1]^T$. We denote the set of all possible values of φ_k as $\mathcal{S}$. The Markov property of φ_k can be proved directly by the definition of Markov chain. Note that, $\{\tau_k(t)\}$ is a Markov chain, $\forall t$, that is, $\mathbb{P}\{\tau_k(t) = i_k | \tau_{k-1} = i_{k-1}, \cdots, \tau_1(t) = i_1\} = \mathbb{P}\{\tau_k(t) = i_k | \tau_{k-1} = i_{k-1}\}$, $\forall t$, $i_k \in \{0, 1\}$. Therefore, $\mathbb{P}\{\varphi_k = \theta_k | \varphi_{k-1} = \theta_{k-1}, \cdots, \varphi_1 = \theta_1\} = \mathbb{P}\{\varphi_k = \theta_k | \varphi_{k-1} = \theta_{k-1}\}$, where $\theta_i \in \mathcal{S}$. As a consequence, the switching of the inputs for the general case is also a Markov chain of 2^N states.

B. Convergence Analysis

In this subsection, the convergence of the input sequences in both mean square and almost sure senses is established. To this end, the original algorithms (4) and (5) are first transformed as a switching system whose random matrix switches as a Markov chain.

We first consider the algorithms for arbitrary fixed time instant t. From Fig. 2, it is observed that the transition matrix of the Markov chain is

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} 1-\overline{\sigma}(1-\overline{\gamma}) & \overline{\sigma}(1-\overline{\gamma}) \\ \overline{\gamma} & 1-\overline{\gamma} \end{bmatrix}, \tag{6}$$

where $p_{11} \triangleq \mathbb{P}(\tau_{k+1}(t) = 1 | \tau_k(t) = 1)$, $p_{12} \triangleq \mathbb{P}(\tau_{k+1}(t) = 0 | \tau_k(t) = 1)$, $p_{21} \triangleq \mathbb{P}(\tau_{k+1}(t) = 1 | \tau_k(t) = 0)$, $p_{22} \triangleq \mathbb{P}(\tau_{k+1}(t) = 0 | \tau_k(t) = 0)$ with p_{ij} being the element of P at the ith row and jth column, $\tau_k(t)$ denoting the state at iteration k, $\forall t$, while $\tau_k(t) = 1$ and $\tau_k(t) = 0$ denoting the states of *synchronization* and *asynchronization*, respectively. Note that $0 < \overline{\sigma}, \overline{\gamma} < 1$; thus, P is irreducible, aperiodic, and recurrent, which further means P is ergodic. In addition, $p_{ij} > 0$, $i, j = 1, 2$.

Note that *renewal* can occur both at the state of *synchronization* and *asynchronization*. Moreover, whenever a *renewal* occurs, both the computed control and the real control are improved. In other words, the real control is updated if a *renewal* occurs, otherwise, it remains unchanged. Therefore, it is concluded that updating of the real control also follows a Markov jump way. We could further introduce a random variable $\lambda_k(t)$ to denote whether a renewal happens or not, i.e., $\lambda_k(t) = 1$ if a renewal happens, and 0 otherwise. Recalling Fig. 2, we find that the occurrence probability of *renewal* depends on its state of the last iteration. That is, the evolution of $\lambda_k(t)$ is also a irreducible, aperiodic, recurrent, and ergodic Markov chain.

The update of the real control has the following two formulations, that is, when $\lambda_k(t) = 0$, $u_{k+1}^r(t) = u_k^r(t) = u_k^r(t) + a_k \mathbf{0}_{p\times q} e_k(t+1)$, and when $\lambda_k = 1$, $u_{k+1}^r(t) = u_k^r(t) + a_k L_t e_k(t+1)$, where $\mathbf{0}_{i\times j}$ denotes zero matrix with appropriate dimensions. We can unify these two cases into the following one

$$u_{k+1}^r(t) = u_k^r(t) + a_k \lambda_k(t) L_t e_k(t+1), \tag{7}$$

where $\lambda_k(t)$ values 0 or 1 subject to a two-state Markov chain.

In order to show the convergence, we now lift the above recursion along the time axis. Specifically, denote $Y_k = [y_k(1), \cdots, y_k(N)]^T \in \mathbb{R}^{qN}$, $U_k^r = [u_k^r(0), \cdots, u_k^r(N-1)]^T \in \mathbb{R}^{pN}$, $Y_k(0) = [C_1 A_0 x_k(0), \cdots, C_N A_{N-1,0} x_k(0)]^T \in \mathbb{R}^{qN}$ and $\mathcal{H}$ is

$$\begin{bmatrix} C_1 B_0 & 0 & 0 & \cdots & 0 \\ C_2 A_1 B_0 & C_2 B_1 & 0 & \cdots & 0 \\ C_3 A_{2,1} B_0 & C_3 A_2 B_1 & C_3 B_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_N A_{N-1,1} B_0 & C_N A_{N-1,2} B_1 & C_N A_{N-1,3} B_2 & \cdots & C_N B_{N-1} \end{bmatrix}$$

with $A_{i,j} \triangleq A_i A_{i-1} \cdots A_j$, $i \ge j$. The stochastic noise term ϵ_k is expressed as

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

The the lifted system is $Y_k = \mathcal{H} U_k + \epsilon_k + Y_k(0)$, while the desired reference $y_d(t)$ and associated desired input $u_d(t)$ can be lifted in similar formulations, $Y_d = \mathcal{H} U_d + Y_d(0)$. From A2–A3, one has that $Y_k(0) = Y_d(0)$ and $\mathbb{E}\{\epsilon_k | \mathcal{F}_{k-1}\} = 0$, $\mathbb{E}\{\|\epsilon_k\|^2 | \mathcal{F}_{k-1}\} < \infty$. In addition, the lifted tracking error is $E_k \triangleq Y_d - Y_k = \mathcal{H}(U_d - U_k^r) - \epsilon_k$.

Then the lifted form of the recursion (7) is

$$\begin{aligned} U_{k+1}^r &= U_k^r + a_k \Lambda_k \mathcal{L} E_k \\ &= U_k^r + a_k \Lambda_k \mathcal{L} \mathcal{H} \Delta U_k^r - a_k \Lambda_k \epsilon_k, \end{aligned} \tag{8}$$

where $\Lambda_k = \text{diag}\{\lambda_k(0), \cdots, \lambda_k(N-1)\} \otimes I_{p\times p}, \mathcal{L} = \text{diag}\{L_0, \cdots, L_{N-1}\} \in \mathbb{R}^{pN\times qN}$, and $\Delta U_k^r = U_d - U_k^r$. Subtracting both sides of the last equation (8) from U_d leads to

$$\Delta U_{k+1}^r = \Delta U_k^r - a_k \Lambda_k \mathcal{L}\mathcal{H}\Delta U_k^r + a_k \Lambda_k \epsilon_k. \tag{9}$$

It is worthy pointing out that Λ_k is a random matrix with 2^N possible outcomes because all its diagonal entries are binary and the switching of Λ_k among its outcomes follows a irreducible, aperiodic, recurrent, and ergodic Markov chain. In particular, there are two special case of Λ_k, namely, $I_{pN\times pN}$ and $\mathbf{0}_{pN\times pN}$ denoting that the inputs at all the N time instants are renewed and unchanged, respectively.

Next it is sufficient to show the zero-error convergence of the recursion (9) under mild design conditions. That is, we can move to design the learning matrix $\mathcal{L}$ and propose the convergence results. Note that $\mathcal{H}$ is a lower triangular block matrix with its diagonal blocks being $C_{t+1}B_t$. Thus we could design the learning gain matrix L_t such that $L_t C_{t+1} B_t$ are with positive eigenvalues. Before presenting the main theorem, we first give two technical lemmas for convergence analysis.

Lemma 2: Let $\{\xi_k\}$ be a sequence of positive real numbers and such that

$$\xi_{k+1} \leq (1 - d_1 a_k)\xi_k + d_2 a_k^2 (d_3 + \xi_k), \tag{10}$$

where $d_i > 0$, $i = 1, \cdots, 3$, are constants and a_k satisfies $a_k > 0$, $\sum_{k=1}^{\infty} a_k = \infty$, and $\sum_{k=1}^{\infty} a_k^2 < \infty$, then $\lim_{k\to\infty} \xi_k = 0$.

Proof: From (10), we have

$$\xi_{k+1} \leq (1 - d_1 a_k + d_2 a_k^2)\xi_k + d_2 d_3 a_k^2. \tag{11}$$

Since $\lim_{k\to\infty} a_k = 0$, we can choose a sufficient large integer $k_0 > 0$ such that $1 - d_1 a_k + d_2 a_k^2 < 1$ for all $k \geq k_0$, and then we have

$$\xi_{k+1} \leq \xi_k + d_4 a_k^2, \tag{12}$$

where $d_4 \triangleq d_2 d_3$. As a result, it follows from (12) and $\sum_{k=1}^{\infty} a_k^2 < \infty$ that $\sup_k \xi_k < \infty$, and ξ_k converges. Based on this boundedness, we have from (11) that $\xi_{k+1} \leq (1 - d_1 a_k)\xi_k + d_5 a_k^2$ with $d_5 > 0$ being suitable constant. Again, noticing $\sum_{k=1}^{\infty} a_k = \infty$ and $\sum_{k=1}^{\infty} a_k^2 < \infty$, we conclude that $\lim_{k\to\infty} \xi_k = 0$. ■

Lemma 3 ([22]): Let $X(n)$, $Z(n)$ be nonnegative stochastic processes (with finite expectation) adapted to increasing σ-algebra $\{\mathcal{F}_n\}$ and such that

$$\mathbb{E}\{X(n+1)|\mathcal{F}_n\} \leq X(n) + Z(n), \tag{13}$$

$$\sum_{n=1}^{\infty} \mathbb{E}[Z(n)] < \infty. \tag{14}$$

Then $X(n)$ converges almost surely, as $n \to \infty$.

Theorem 1: Consider the linear time varying system (1) and assume A1–A3 hold. Moreover, the data at both measurement and actuator sides are dropped or successfully transmitted for the whole iteration together. The learning update laws (4) and (5) guarantee that the generated input sequence converges to the desired input both in mean square sense and almost sure sense if the learning gain matrix L_t satisfies that all eigenvalues of $L_t C_{t+1} B_t$ are positive, $t = 0, 1, \cdots, N-1$. As a result, the desired reference $y_d(t)$ is asymptotically tracked according to the index (3).

Proof: From Lemma 1 and the transformations, we have (9). Now we show that ΔU_k^r converge to zero both in mean square sense and almost sure sense.

Note that all eigenvalues of the matrix $\mathcal{L}\mathcal{H}$ are positive, thus there exists a positive definite matrix $\mathcal{Q}$ such that $(\mathcal{L}\mathcal{H})^T\mathcal{Q} + \mathcal{Q}\mathcal{L}\mathcal{H} = I$. Moreover, according to the form of Λ_k, we have $(\mathcal{L}\mathcal{H})^T \Lambda_k^T \mathcal{Q} + \mathcal{Q}\Lambda_k \mathcal{L}\mathcal{H} \geq 0$. Then, we define a weighted norm for ΔU_k^r as $\|\Delta U_k^r\|_{\mathcal{Q}}^2 \triangleq (\Delta U_k^r)^T \mathcal{Q} \Delta U_k^r$, which can be regarded as a Lyapunov function. Now, we take the weighted norm to both sides of (9),

$$\begin{aligned}\|\Delta U_{k+1}^r\|_{\mathcal{Q}}^2 &= \|\Delta U_k^r\|_{\mathcal{Q}}^2 + a_k^2 \|\Lambda_k \mathcal{L}\mathcal{H}\Delta U_k^r\|_{\mathcal{Q}}^2 + a_k^2 \|\Lambda_k \epsilon_k\|_{\mathcal{Q}}^2 \\ &\quad - a_k (\Delta U_k^r)^T ((\mathcal{L}\mathcal{H})^T \Lambda_k^T \mathcal{Q} + \mathcal{Q}\Lambda_k \mathcal{L}\mathcal{H})\Delta U_k^r \\ &\quad + 2a_k (\Delta U_k^r)^T \mathcal{Q}\Lambda_k \epsilon_k \\ &\quad - 2a_k (\Delta U_k^r)^T (\mathcal{L}\mathcal{H})^T \Lambda_k^T \mathcal{Q}\Lambda_k \epsilon_k. \end{aligned} \tag{15}$$

Define a new increasing σ-algebra $\mathcal{F}_k' \triangleq \sigma\{x_j(t), w_j(t), v_j(t), \sigma_j(t), \gamma_j(t), 1 \leq j \leq k-1, t = 0, \cdots, N\}$. In view of (4)–(5), it is evident that $U_k^r \in \mathcal{F}_k'$, that is, U_k^r is adapted to $\mathcal{F}_k'$. From A3, we have $\mathbb{E}\{\epsilon_k|\mathcal{F}_k'\} = 0$. As a result,

$$\mathbb{E}\{2a_k (\Delta U_k^r)^T \mathcal{Q}\Lambda_k \epsilon_k|\mathcal{F}_k'\} = 0, \tag{16}$$

$$\mathbb{E}\{2a_k (\Delta U_k^r)^T (\mathcal{L}\mathcal{H})^T \Lambda_k^T \mathcal{Q}\Lambda_k \epsilon_k|\mathcal{F}_k'\} = 0. \tag{17}$$

Therefore, it is straightforward to have that

$$\begin{aligned}&\mathbb{E}\{\|\Delta U_{k+1}^r\|_{\mathcal{Q}}^2|\mathcal{F}_k'\} \\ &\quad = \|\Delta U_k^r\|_{\mathcal{Q}}^2 + c_0 a_k^2 (\|\Delta U_k^r\|_{\mathcal{Q}}^2 + \mathbb{E}\{\|\epsilon_k\|_{\mathcal{Q}}^2|\mathcal{F}_k'\}) \\ &\qquad - a_k (\Delta U_k^r)^T \mathbb{E}\{(\mathcal{L}\mathcal{H})^T \Lambda_k^T \mathcal{Q} + \mathcal{Q}\Lambda_k \mathcal{L}\mathcal{H}|\mathcal{F}_k'\}\Delta U_k^r, \end{aligned} \tag{18}$$

where $c_0 = \max\{\|LH\|^2, 1\}$ as $\|\Lambda_k\| \leq 1$. Note that Λ_k is a diagonal matrix, and the evolution of Λ_k is a irreducible and ergodic Markov chain. Thus there is a positive probability for Λ_k to be the identity matrix $I_{pN\times pN}$. In addition, all the eigenvalues of the remaining $2^N - 1$ possible diagonal-matrix of Λ_k are either 1 or 0. Consequently, there exists some constant $c_1 > 0$ such that

$$\mathbb{E}\{(\mathcal{L}\mathcal{H})^T \Lambda_k^T \mathcal{Q} + \mathcal{Q}\Lambda_k \mathcal{L}\mathcal{H}|\mathcal{F}_k'\} \geq c_1 I. \tag{19}$$

By A3 we have $\mathbb{E}\{\|\epsilon_k\|_{\mathcal{Q}}^2|\mathcal{F}_k'\} < c_2$ where $c_2 > 0$ is a suitable constant.

Now we move to show the mean square convergence. Denote $\xi_k \triangleq \mathbb{E}\|\Delta U_k^r\|_{\mathcal{Q}}^2$. Then taking mathematical expectation of both sides of (18) and using (19) lead to

$$\xi_{k+1} \leq (1 - c_1 c_3 a_k)\xi_k + c_0 a_k^2 (c_2 + \xi_k).$$

where c_3 is positive constant such that $I \geq c_3 \mathcal{Q}$. Then by Lemma 2, we have that $\mathbb{E}\|\Delta U_k^r\|_{\mathcal{Q}}^2 \to 0$, implying that $\mathbb{E}\|\Delta U_k^r\|^2 \to 0$. That is, the zero-error convergence of ΔU_k^r in mean square sense is proved.

Next, we proceed to show the almost sure convergence of ΔU_k^r. Denote $\eta_k \triangleq \|\Delta U_k^r\|_{\mathcal{Q}}^2$. Substituting (19) into (18), we have that

$$\begin{aligned}\mathbb{E}\{\eta_{k+1}|\mathcal{F}_k'\} &\leq (1 - c_1 c_3 a_k)\eta_k + c_0 a_k^2 (\mathbb{E}\{\|\epsilon_k\|^2|\mathcal{F}_k'\} + \eta_k) \\ &\leq \eta_k + c_0 a_k^2 (\mathbb{E}\{\|\epsilon_k\|^2|\mathcal{F}_k'\} + \eta_k). \end{aligned} \tag{20}$$

Note that two terms on the right-hand side of last inequality, i.e., η_k and $c_0 a_k^2 (\mathbb{E}\{\|\epsilon_k\|^2|\mathcal{F}_k'\} + \eta_k)$, corresponds to $X(n)$ and $Z(n)$ in Lemma 3, respectively. Moreover, it is evident that $\sum_{k=1}^{\infty} \mathbb{E}\left[c_0 a_k^2 (\mathbb{E}\{\|\epsilon_k\|^2|\mathcal{F}_k'\} + \eta_k)\right] = \sum_{k=1}^{\infty} c_0 a_k^2 (\mathbb{E}\|\epsilon_k\|^2 + \xi_k) < \infty$, where the convergence of ξ_k is used. In other words, (13) and (14) in Lemma 3 are fulfilled. Therefore, it follows that $\eta_k = \|\Delta U_k^r\|_{\mathcal{Q}}^2$ converges almost surely as $k \to \infty$. On the other hand, we have show that ΔU_k^r converges to zero in mean square, thus ΔU_k^r converges to zero almost surely from probability theory. The proof is completed. ■

Remark 5: In this technical note, the data dropouts are modeled by random variables subject to Bernoulli distribution, which is a widely used model in this research area. The major reason for such assumption is to allow the successive data dropouts with arbitrary length. In addition, such assumption also helps us to make an explicit convergence proof for the general data dropout problem. Note that the critical technique for establishing the convergence is the Markov property of the sample path behavior, thus the proposed method given in this technical note can be applied to the Markovian data dropout case [23], [24]. On the other hand, the inherent update mechanism is the occurrence of *renewal*, which implies that the essential convergence of the proposed algorithms only requires that the data are not completely lost for each time instant.

Remark 6: If no noise is involved in the system, that is, both $w_k(t)$ and $v_k(t)$ are eliminated, then the decreasing gain a_k could be removed from the algorithms. In this case, the recursion of input error (9) reduces to

$$\Delta U_{k+1}^r = \Delta U_k^r - \Lambda_k \mathcal{L}\mathcal{H}\Delta U_k^r \tag{21}$$

and an exponential convergence speed can be then obtained. Moreover, the design of learning gain matrix L_t would be different with or without a_k. For the case with a_k, the condition on L_t is that all eigenvalues of $L_t C_{t+1} B_t$ are positive real numbers. Roughly speaking, this condition can be relaxed to the one that all eigenvalues of $L_t C_{t+1} B_t$ are with positive real parts. On the other hand, for the case without a_k, the matrix L_t should satisfy that the spectral radius of $I - L_t C_{t+1} B_t$ is less than one to ensure convergence. The latter design condition can be derived following the traditional contraction mapping method. Therefore, the introduction of decreasing sequence $\{a_k\}$ also relax the design range of L_t.

C. Discussions on Convergence Speed

In this subsection, we give a brief description on the convergence speed of the proposed algorithms. As a matter of fact, the convergence speed depends on two individual factors, namely, the *renewal* frequency and the decreasing gain sequence $\{a_k\}$. The former factor reflects the influence of data dropout on ILC, while the latter factor is a design factor originating from the stochastic approximation algorithm. In the traditional ILC problem where no data dropout occurs, the convergence speed is only determined by the designed decreasing gain a_k.

Now let us check how is the influence of data dropout on the convergence speed. To this end, we give an explicit description on the *renewal* frequency. Recalling Fig. 2 and its transition matrix (6), the associated stationary distribution π of the Markov chain can be calculated from $\pi P = \pi$ and is given as

$$\pi \triangleq \left[\frac{\overline{\gamma}}{\overline{\gamma} + \overline{\sigma} - \overline{\sigma}\cdot\overline{\gamma}}, \quad \frac{\overline{\sigma} - \overline{\sigma}\cdot\overline{\gamma}}{\overline{\gamma} + \overline{\sigma} - \overline{\sigma}\cdot\overline{\gamma}}\right]. \tag{22}$$

Note that *renewal* can occur both at the state of *synchronization* and *asynchronization*. From Fig. 2, it is observed that the probability of occurrence of *renewal* at the state of *synchronization* is $\overline{\sigma\gamma}$, while the probability at the state of *asynchronization* is $\overline{\gamma}$. Therefore, the probability of *renewal* along the iteration axis can be calculated

$$\begin{aligned}\mathbb{P}(renewal) &= \frac{\overline{\gamma}}{\overline{\gamma} + \overline{\sigma} - \overline{\sigma}\cdot\overline{\gamma}}\cdot\overline{\sigma}\cdot\overline{\gamma} + \frac{\overline{\sigma} - \overline{\sigma}\cdot\overline{\gamma}}{\overline{\gamma} + \overline{\sigma} - \overline{\sigma}\cdot\overline{\gamma}}\cdot\overline{\gamma}\\ &= \frac{\overline{\sigma}\cdot\overline{\gamma}}{\overline{\gamma} + \overline{\sigma} - \overline{\sigma}\cdot\overline{\gamma}}.\end{aligned} \tag{23}$$

In this technical note, for clear expression, we simply assume that the probability distributions for different time instants are the same. Thus the above probability of *renewal* is the average of the whole iteration. This probability describes the *renewal* frequency of the learning algorithms under data dropout environment. It is noticed that the probability is determined by the sum and product of the successful transmission probability at both sides, i.e., $\overline{\sigma} + \overline{\gamma}$ and $\overline{\sigma\gamma}$. As a consequence, the *renewal* frequency is neither determined by the worst side nor the simple sum of both sides. Two facts are observed as follows.

1) Define the function $f(\overline{\sigma}, \overline{\gamma}) = \mathbb{P}(renewal)$. Evidently, $f(\overline{\sigma}, \overline{\gamma}) = f(\overline{\gamma}, \overline{\sigma})$. Moreover, through simple calculations, one has

$$\frac{\partial f(\overline{\sigma}, \overline{\gamma})}{\partial\overline{\sigma}} = \frac{\overline{\gamma}^2}{(\overline{\gamma} + \overline{\sigma} - \overline{\sigma}\cdot\overline{\gamma})^2} > 0.$$

This condition means that a large successful transmission rate corresponds to increased number of renewals, and thus, faster convergence speed.

2) It is well known that $\overline{\sigma\gamma} \le \left(\frac{\overline{\sigma}+\overline{\gamma}}{2}\right)^2$ where the equality holds if and only if $\overline{\sigma} = \overline{\gamma}$. This implies that, when the sum $\overline{\sigma} + \overline{\gamma}$ is fixed, the closer $\overline{\sigma}$ approaches to $\overline{\gamma}$, the larger the product $\overline{\sigma\gamma}$ is and so is the probability $\mathbb{P}(renewal)$. In other words, the convergence speed increases.

IV. Illustrative Simulations

In this section, we apply the proposed algorithms to a permanent magnet linear motor (PMLM), which is described by the following discretized model [25]

$$\begin{cases} x(t+1) = x(t) + v(t)\Delta + \varepsilon_1(t+1) \\ v(t+1) = v(t) - \Delta\frac{k_1 k_2 \psi_f^2}{Rm}v(t) + \Delta\frac{k_2\psi_f}{Rm}u(t) + \varepsilon_2(t+1) \\ y(t) = v(t) + \epsilon(t) \end{cases}$$

where x and v denote the motor position and rotor velocity, $\Delta = 10$ ms the sampling time interval, $R = 8.6\,\Omega$ the resistance of stator, $m = 1.635$ kg the rotor mass, and $\psi_f = 0.35$ Wb the flux linkage, $k_1 = \pi/\tau$ and $k_2 = 1.5\pi/\tau$, where $\tau = 0.031$ m is the pole pitch. The stochastic noises $\varepsilon_1(t)$, $\varepsilon_2(t)$, and $\epsilon(t)$ obey zero-mean distribution $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.03$.

In this simulation, we set the whole iteration length as 1 s, i.e., $N = 100$. The desired reference is $y_d(t) = 1/3[\sin(t/20) + 1 - \cos(3t/20)]$, $0 \le t \le 100$. The initial state satisfies A2. The control input for the first iteration is simply set to be 0. The learning gain $L_t = 50$ and the decreasing sequence is set to be $a_k = 1/k$. The algorithms are run for 150 iterations.

The general algorithms (4) and (5) are used. The random variables $\gamma_k(t)$ and $\sigma_k(t)$ for data dropouts are defined separately for different time instants rather than a unified variable for the entire iteration. We introduce data dropout rate (DDR) as the probability $\mathbb{P}(\sigma_k(t) = 0)$ or $\mathbb{P}(\gamma_k(t) = 0)$. In other words, DDR denotes the percentage of lost packages over the total packages. For simplicity, the DDRs for both the measurement and actuator sides are equal in the following. In this example, four cases are simulated, that is, $\mathrm{DDR} = 10\%$, 20%, 30%, and 40%, respectively. The averaged tracking error profiles along iteration axis for all cases are shown in Fig. 3, where the averaged tracking error is defined as $\overline{e}_k \triangleq \frac{\sum_{t=1}^{N}|e_k(t)|}{N}$. As can be seen, the convergence speed slows as the DDR increases.

As has been shown in Theorem 1, the condition on learning gain L_t is that all eigenvalues of $L_t C_{t+1} B_t$ are positive. Moreover, a faster convergence speed can be achieved when L_t is designed such that the eigenvalues are with larger magnitude; however, such case would lead to bad transient performance such as overshot before convergence. Thus, there is a trade-off between the convergence speed and transient performance.

If there is no noise involved in the system, i.e., the noises $w_k(t)$ and $v_k(t)$ are removed from the system (1), then we can also delete the decreasing sequence a_k from the update laws (4) as mentioned in Remark 2. In this case, an exponential convergence speed is achieved, as shown in Fig. 4, where the learning gain L_t is selected as $L_t = 8$ due to the previous value does not satisfy the condition given in Remark 6.

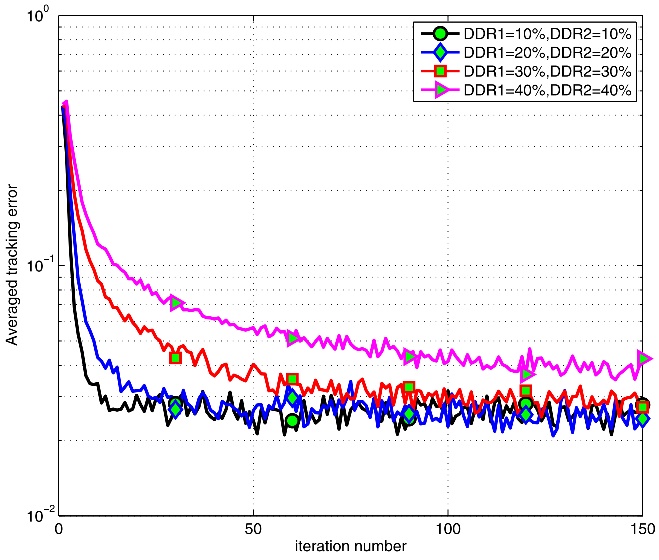

Fig. 3. Tracking error profiles along iteration axis with different DDRs: Noise case (DDR1: DDR at the measurement side; DDR2: DDR at the actuator side).

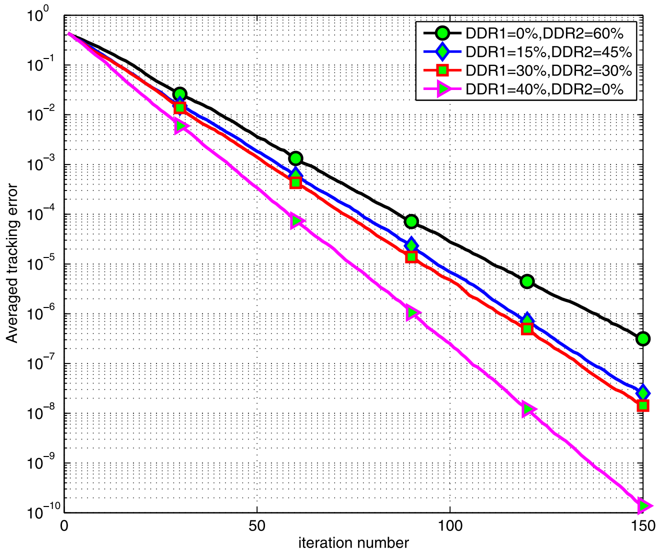

Fig. 4. Tracking error profiles along iteration axis with different DDRs: Noise free case (DDR1: DDR at the measurement side; DDR2: DDR at the actuator side).

In the figure, all profiles are approximate lines in the logarithmic coordinate, which imply the exponential convergence property.

In addition, Fig. 4 also verifies the relationship between convergence speed and DDR rate. For the above three lines, the sum of DDRs at both sides are identical. It can be found that the fastest convergence speed belongs to the case DDR1 = DDR2. Moreover, although the worst DDR in the forth line is 40%, it still behaves faster than the third line where the worst DDR is 30%. Such observations coincide with Subsection III.C.

To demonstrate the effect of the decreasing sequence a_k for stochastic systems, we display the input profiles for (4) with and without a_k in Fig. 5. As can be seen, the introduction of a_k enables a stable convergence of input, while the input keeps fluctuating if such sequence is removed from the update law. This verifies the necessity of decreasing term in algorithms for stochastic systems. However, it should be pointed out that the decreasing sequence may make the learning controller unsuitable if large changes occur to the desired reference after several iterations. As a consequence, the design of learning laws depends on the practical application requirements.

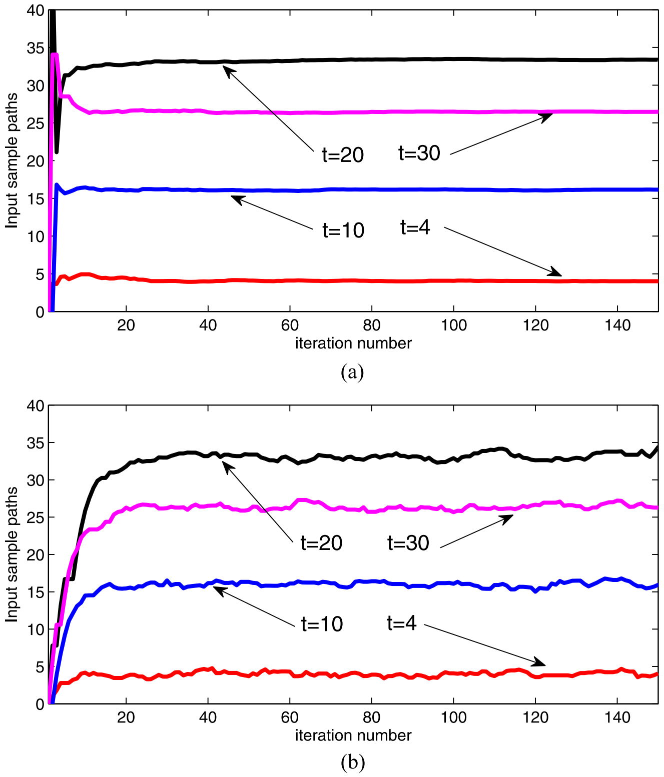

Fig. 5. Input sample paths along iteration axis with different DDRs. (a) With a_k case, (b) Without a_k case.

V. Conclusion

ILC under general data dropout environments is explored in this technical note. The data dropouts are allowed to occur randomly at both the measurement and actuator sides. As a result, the control update process consists of two parts, i.e., the computed input and the real input. A novel convergence analysis framework is proposed in this technical note. To be specific, the update process is first proved to be a Markov chain by directly analyzing its sample path behavior. Then the convergence in both mean square and almost sure senses is established strictly. In addition, the technical note also demonstrates the effectiveness and robustness of conventional P-type update law against random factors.

References

[1] H.-S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: Survey and categorization from 1998 to 2004," *IEEE Trans. Syst., Man, Cybern. C*, vol. 37, no. 6, pp. 1099–1121, 2007.

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

[3] D. Huang, J.-X. Xu, V. Venkataramanan, and T. C. T. Huynh, "High performance tracking of piezoelectric positioning stage using current-cycle iterative learning control with gain scheduling," *IEEE Trans. Ind. Electron.*, vol. 61, no. 2, pp. 1085–1098, 2014.

[4] X. Li, Q. Ren, and J.-X. Xu, "Precise speed tracking control of a robotic fish via iterative learning control," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2221–2228, 2016.

[5] H. S. Ahn, Y. Q. Chen, and K. L. Moore, "Intermittent iterative learning control," in *Proc. 2006 IEEE Int. Symp. Intelligent Control*, pp. 832–837.

[6] H. S. Ahn, K. L. Moore, and Y. Q. Chen, "Discrete-time intermittent iterative learning controller with independent data dropouts," in *Proc. 2008 IFAC World Congr.*, pp. 12442–12447.

[7] H. S. Ahn, K. L. Moore, and Y. Q. Chen, "Stability of discrete-time iterative learning control with random data dropouts and delayed controlled signals in networked control systems," in *Proc. IEEE Int. Conf. Control Automation, Robotics, and Vision 2008*, pp. 757–762.

[8] X. Bu, Z.-S. Hou, and F. Yu, "Stability of first and high order iterative learning control with data dropouts," *Int. J. Control, Automation, Syst.*, vol. 9, no. 5, pp. 843–849, 2011.

[9] X. Bu, Z.-S. Hou, F. Yu, and F. Wang, "H-∞ iterative learning controller design for a class of discrete-time systems with data dropouts," *Int. J. Syst. Sci.*, vol. 45, no. 9, pp. 1902–1912, 2014.

[10] C. Liu, J.-X. Xu, and J. Wu, "Iterative learning control for remote control systems with communication delay and data dropout," *Math. Problems in Eng.*, Article ID 705474, pp. 1–14, 2012.

[11] D. Shen and Y. Wang, "Iterative learning control for networked stochastic systems with random packet losses," *Int. J. Control*, vol. 88, no. 5, pp. 959–968, 2015.

[12] D. Shen and Y. Wang, "ILC for networked nonlinear systems with unknown control direction through random Lossy channel," *Syst. Control Lett.*, vol. 77, pp. 30–39, 2015.

[13] X. Bu, F. Yu, Z.-S. Hou, and F. Wang, "Iterative learning control for a class of nonlinear systems with random packet losses," *Nonlinear Anal., Real World Appl.*, vol. 14, no. 1, pp. 567–580, 2013.

[14] Y.-J. Pan, H. J. Marquez, T. Chen, and L. Sheng, "Effects of network communications on a class of learning controlled non-linear systems," *Int. J. Syst. Sci.*, vol. 40, no. 7, pp. 757–767, 2009.

[15] L.-X. Huang and Y. Fang, "Convergence analysis of wireless remote iterative learning control systems with dropout compensation," *Math. Problems in Eng.*, Article ID 609284, pp. 1–9, 2013.

[16] J. Liu and X. Ruan, "Networked iterative learning control approach for nonlinear systems with random communication delay," *Int. J. Syst. Sci.*, vol. 47, no. 16, pp. 3960–3969, 2016.

[17] J. Emelianova, P. Pakshin, K. Galkowski, and E. Rogers, "Stability of nonlinear discrete repetitive processes with Markovian switching," *Syst. Control Lett.*, vol. 75, pp. 108–116, 2015.

[18] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. New York: Springer-Verlag, 1990.

[19] P. E. Caines, *Linear Stochastic Systems*. New York: Wiley, 1988.

[20] S. S. Saab, "A discrete-time stochastic learning control algorithm," *IEEE Trans. Autom. Control*, vol. 46, no. 6, pp. 877–887, 2001.

[21] 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, 2005.

[22] J. N. Tsitsiklis, D. P. Berteskas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, 1986.

[23] K. You and L. Xie, "Minimum data rate for mean square stabilizability of linear systems with Markovian packet losses," *IEEE Trans. Autom. Control*, vol. 56, no. 4, pp. 772–785, 2011.

[24] Y. Shi and B. Yu, "Output feedback stabilization of networked control systems with random delays modeled by Markov chains," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1668–1674, 2009.

[25] W. Zhou, M. Yu, and D. Huang, "A high-order internal model based iterative learning control scheme for discrete linear time-varying systems," *Int. J. Automation and Computing*, vol. 12, no. 3, pp. 330–336, 2015.