Asian Journal of Control, Vol. 20, No. 3, pp. 1102–1114, May 2018
Published online 16 February 2017 in Wiley Online Library (wileyonlinelibrary.com) DOI: 10.1002/asjc.1480

INTERMITTENT AND SUCCESSIVE ILC FOR STOCHASTIC NONLINEAR SYSTEMS WITH RANDOM DATA DROPOUTS

Dong Shen, Chao Zhang, and Yun Xu

ABSTRACT

The iterative learning control (ILC) problem is addressed in this paper for stochastic nonlinear systems with random data dropouts. The data dropout is modeled by the conventional Bernoulli random variable to describe the successful transmission or loss. Both intermittent and successive ILC are considered, where the former stops updating if no information is received, while the latter keeps updating based on the latest available data. It is strictly proved the almost sure convergence of both algorithms. The simulations on a mechanical model are provided to show the comparisons and effectiveness of the proposed algorithms.

Key Words: Iterative learning control, data dropouts, intermittent scheme, successive scheme, stochastic nonlinear system.

I. INTRODUCTION

Iterative learning control (ILC) is a kind of intelligent control strategy, applied to those systems that complete a given task over a finite time interval repeatedly. For these kinds of systems, such as industrial processes and robots, one can update the input signal in terms of the inputs and outputs from previous iterations as well as the desired reference. Thus the tracking performance is successively improved along the iteration axis, differing from traditional control strategies that improve control performance along time axis. It was first proposed by Arimoto [1], where the ILC was designed for better operation performance of robots. Now ILC has been developed for three decades and a lot of excellent achievements have been reported [2–4]. Meanwhile, ILC is applicable in many practical types of equipment such as permanent magnet step motors [5], robotic-assisted rehabilitation [6], and industrial robots [7].

In practical applications, the plant and the controller usually communicate with each other through wired/wireless networks [8]. In this setting, the data may be dropped during transmission due to complex transmission conditions such as network congestion, broken linkages, and transmission errors. As is well known, the data dropouts can damage the tracking performance seriously. Therefore, it is a critical topic to be well handled. To describe the randomness of data dropouts, the Bernoulli random variable is widely used [9,10]. In this model, the data dropout is expressed as an independent process. For each data packet, only two cases are taken into consideration, namely, successful transmission and loss. Some compensating mechanisms for dropped data are provided to get achievable control performances in [11]. As a matter of fact, the data dropout is a lasting and quite hot issue in the field of automatic control.

Concerning the topic of ILC dealing with random data dropouts, several papers can also be found. However, the problem is not well solved. Ahn *et al.* made early attempts in [12–14] based on the Kalman filtering analysis techniques proposed in [28]. Thus the mean-square stability of ILC algorithms was derived for time-invariant linear systems. The major differences among [12–14] were the locations that data dropouts took place. To be specific, only output loss was discussed in [12,13] while the case that data dropouts happened to the input as well as output was dealt with in [14]. However, it is noticed the system matrices should be prior known for the design of ILC under this framework. In addition, it is quite hard to extend the proposed approach to nonlinear systems because of the essential character of the Kalman filtering technique.

Moreover, Bu *et al.* also contributed to this issue in [15–17] from the statistics point of view. That is, the convergence analysis was given based on the mathematical expectation of tracking errors. The linear system case was considered in [15] while nonlinear case was addressed in [16]. In addition, [17] provided an H_∞ ILC analysis for discrete-time system with random data dropouts,

Manuscript received March 8, 2016; revised July 20, 2016; accepted December 7, 2016.
The authors are with College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China.
Dong Shen is the corresponding author (e-mail: shendong@mail.buct.edu.cn).
This work is supported by National Natural Science Foundation of China (61673045, 61304085), Beijing Natural Science Foundation (4152040).

where the H_∞ performance problem was defined and discussed in the iteration domain. By taking mathematical expectations, the random iterative equation is transformed into deterministic forms, where the inherent randomness of data dropout is then concealed. Thus it is proved that only the expectation of tracking error converges. However, mathematical expectation is not sufficient to describe the performance of the final tracking error, which is a random variable.

Furthermore, [18,19] also discussed the ILC problem under random data dropouts. The almost sure convergence was strictly proved for both the known control direction case [18] and the unknown control direction case [19]. However, it was required that the iteration length of successive data dropouts should be finite in [18,19], and this finite length requirement of successive dropouts is somewhat tight as is not totally stochastic. It is evident that the widely used Bernoulli random variable model of data dropouts can not be covered by [18,19]. As a matter of fact, the relaxation of finite length requirement is quite difficult and nontrivial. The reason is that there are two random factors involved together, *i.e.*, data dropout and successive dropout iteration length, and it is hard to consider them individually.

In addition, it is found that in all the above papers [12–19], only the intermittent updating strategy is adopted. That is, the ILC algorithm only updates its input when the data is not dropped. Otherwise, the algorithm just stops updating and waits for the next successfully transmitted data packet. In other words, the iterations with no available data are totally wasted since nothing is done during these iterations, which further slows down the convergence speed.

Based on the above results, the motivation of this paper is to make a comprehensive answer to the ILC problem for the nonlinear system under random data dropouts. First of all, we take the widely used Bernoulli random variable model of data dropouts into account, where the successive iterations of data dropouts could be arbitrarily large and can not be covered [18,19]. Moreover, we deal with the random data dropout and successive dropout iteration length simultaneously following the stochastic analysis way. It is important to point out that direct convergence derivations according to the random variables are difficult, and this is why previous papers took mathematical expectations, as in [15–17], or covariance, as in [12–14], to eliminate the randomness. Furthermore, we establish the almost sure convergence for the Bernoulli model case. It is apparent that there is no other convergence property that could lead to the almost sure convergence. In addition, we discuss two update strategies to handle the data dropouts problem, one of which is the traditional intermittent updating strategy, and the other one is successive updating strategy. The latter one means that the algorithm keeps updating no matter whether the data is dropped or not. Last but not least, to make the algorithm more suitable for practical applications, we aim to use simple data-driven algorithm. That is, only the available input and output information is used to generate the input sequence and the system information is neither required or estimated.

It should be emphasized that this paper aims to complete the theory of ILC under data dropout conditions, rather than provide another novel ILC algorithm. To the best knowledge of the authors, this is the first time that the almost sure convergence of ILC for nonlinear systems under random data dropouts is shown, which are described by the Bernoulli random variable model. The results reported in this paper could not be derived using the techniques proposed in previous papers. Moreover, the conventional P-type algorithm with a prior design of the learning gain is adopted to express our contribution. On one hand, it is because that the conventional P-type ILC algorithm possesses good robustness according to random factors. On the other hand, the algorithm could be further modified to cope with other issues in ILC field. In addition, this paper discusses the nonlinear discrete-time system with stochastic measurement noises. Meanwhile, the output suffers random data dropouts. As a result, the classic contraction mapping method and composite energy function method fail to deal with the problem. In this paper, we propose an alternative convergence analysis based on stochastic approximation technique. Last but not least, in order to deal with the successive dropouts iteration length, detailed estimations are given in the analysis, which makes the analysis technique nontrivial although the proof framework seems similar to our previous work.

In summary, the main contributions, comparing with previous related papers, are listed as follows.

- The ILC for nonlinear systems with random data dropouts is addressed in this paper. The measurement noises are also involved in the output.
- Two update algorithms, namely, intermittent ILC algorithm (I-ILC) and successive ILC algorithm (S-ILC), are proposed to deal with the data dropouts problem. By I-ILC we mean the algorithm only updates the input signal when the corresponding data is successfully transmitted, while by S-ILC we mean the algorithm keeps updating with the latest available data no matter whether data dropouts happen.
- Both I-ILC and S-ILC are data-driven algorithms. That is, only the input and output as well as the desired tracking reference are used to construct the

update law. The system information and prior probability distribution on the randomness are assumed unknown.
- The almost sure convergence of the proposed algorithms for Bernoulli model of data dropouts is strictly proved. Specifically, the input sequences generated by both algorithms are proved to converge to the desired input in the almost sure sense.

The rest of the paper is arranged as follows. Problem formulation is given in Section II including system formulation, data dropouts model, control objective, learning laws and preliminary lemmas. The almost sure convergence results are given in Section III. Section IV provides an illustrative simulation to show the effectiveness of proposed algorithms. Section V concludes the paper. All proofs are given in the appendices.

Notation. $\mathbb{R}$ denotes the real number field, and $\mathbb{R}^n$ is the n-dimensional real space. $\mathbb{N}$ is the set of all positive integers. $I_{n\times n}$ denotes n-dimensional identity matrix. $\mathbb{P}$ denotes the probability of an event while $\mathbb{E}$ is the mathematical expectation. The superscript T denotes transpose of a matrix or vector. For two sequences $\{a_n\}$ and $\{b_n\}$, we called $a_n = O(b_n)$ if $b_n \geq 0$ and there exists $L > 0$ such that $|a_n| \leq Lb_n$, $\forall n$, and $a_n = o(b_n)$ if $b_n \geq 0$ and $(a_n/b_n) \to 0$ as $n \to \infty$. The abbreviations "i.o." and "a.s." denote "infinitely often" and "almost surely", respectively.

II. PROBLEM FORMULATION

2.1 System formulation

Consider the following time-varying nonlinear system with stochastic measurement noise

$$\begin{aligned} x_k(t+1) &= f(t, x_k(t)) + \boldsymbol{b}(t, x_k(t))u_k(t) \\ y_k(t) &= \boldsymbol{c}(t)x_k(t) + w_k(t) \end{aligned} \tag{1}$$

where $k = 1, 2, \cdots$ denotes different iteration number, while $t = 0, 1, \cdots, N$ labels different time instances in an iteration, and N is the length of each iteration. $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}$, and $y_k(t) \in \mathbb{R}$ denote the state, the input, and the output of the system, respectively. $f(t, x_k(t))$, $\boldsymbol{b}(t, x_k(t))$, and $\boldsymbol{c}(t)$ denote unknown system information. The random variable $w_k(t)$ is the measurement noise. Many practical systems can be modeled by the affine nonlinear model, such as mass-spring system [20], single-link manipulator system [21], and two-link planar robot arm [7]. However, as will be shown below, the proposed algorithms require little information on system model, which shows that ILC is a favorable data driven approach to deal with nonlinear systems.

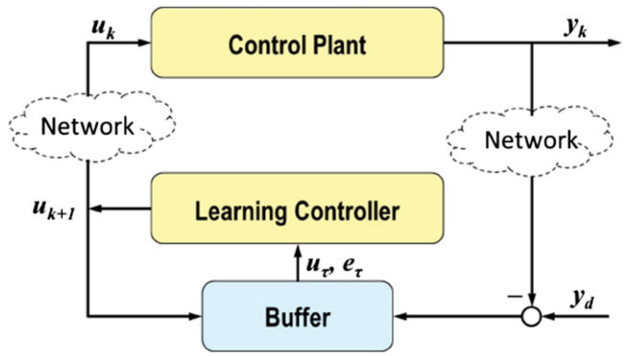

Fig. 1. Block diagram of networked control system. [Color figure can be viewed at wileyonlinelibrary.com]

The setup of the control system is illustrated in Fig. 1, where the plant and learning controller locate separately and communicate via networks. Due to network congestion, linkage interrupt and transmission error, the data may be dropped out through the networks. However, for concise expression without loss of any generality, the data dropouts are only considered for the side of output, *i.e.*, the random data dropouts only happen on the network from the measurement output to the buffer, while the network from learning controller to control plant is assumed to work well. This formulation is adopted to make our following expressions clear and the focal point highlighted. When considering the general data dropouts at both sides, the asynchronous update between the control signal generated by the learning controller and the one fed to the plant should be taken into account and more detailed analysis are required. However, this is out of the scope of this paper.

Let the desired reference be $y_d(t)$, $t = 0, 1, \cdots, N$, with initial state $x_d(0)$, where $y_d(0) = \boldsymbol{c}(0)x_d(0)$.

The following mild assumptions are given for system (1).

A 1. For any $t = 0, 1, \cdots, N$, the functions $f(t, x)$ and $\boldsymbol{b}(t, x)$ are continuous with respect to the second argument x.

Remark 1. A1 could be relaxed to the case that the functions $f(t, x)$ and $\boldsymbol{b}(t, x)$ are allowed to have discontinuities with respect to x away from $x_d(t)$, where $x_d(t)$ is defined later in Remark 4. Since $x = x_d(t)$ is unknown prior, thus A1 is simply assumed.

A 2. The input/output coupling value $\boldsymbol{c}(t+1)\boldsymbol{b}(t, x)$ is unknown, but it is nonzero and does not change its sign during learning processes. Without loss of any generality, it is assumed known that $\boldsymbol{c}(t+1)\boldsymbol{b}(t, x) > 0$ for expression convenience in the rest of this paper.

Remark 2. The input/output coupling value denotes the control direction. Control direction is a necessary information for the design of controller. This is why we assume that $\boldsymbol{c}(t+1)\boldsymbol{b}(t,x)$ does not change its sign. Otherwise, the controller would be very complex since we have to design a scheme to find the right control direction adaptively. Similar techniques in [19] can be used to handle this issue. Since it is out of the scope of this paper, we simply give A2.

Remark 3. In A2, the assumption that $\boldsymbol{c}(t+1)\boldsymbol{b}(t,x)$ is nonzero implies that the relative degree of the system (1) is 1. However, this case can be extended to the high relative degree case with slightly revisions to the learning algorithms. To be specific, assume the system is of high relative degree τ, that is, for any t, $\boldsymbol{c}\frac{\partial f^{\tau-1}(f+\boldsymbol{b}u)}{\partial u}$ is nonzero and $\boldsymbol{c}\frac{\partial f^{i}(f+\boldsymbol{b}u)}{\partial u}=0$, $0\le i\le \tau-2$, where $f^i(x)=f^{i-1}\circ f(x)$ and $\circ$ denotes the composite operator of functions [23]. For this case, when updating the input at time instant t, the tracking error at time $t+\tau$ is used for the learning algorithms given in the next section instead of the one at time $t+1$.

Remark 4. If the system (1) is noise free, then based on A2 we could recursively define the optimal input $u_d(t)$ as follows, $t=0,1,\cdots,N-1$

$$\begin{aligned} u_d(t) &= [\boldsymbol{c}(t+1)\boldsymbol{b}(t,x_d(t))]^{-1} \\ &\quad \times (y_d(t+1)-\boldsymbol{c}(t+1)f(t,x_d(t))) \\ x_d(t+1) &= f(t,x_d(t))+\boldsymbol{b}(t,x_d(t))u_d(t), \end{aligned}$$

with the initial state $x_d(0)$. It is obvious that the following relationship holds for the desired reference $y_d(t)$,

$$\begin{aligned} x_d(t+1) &= f(t,x_d(t))+\boldsymbol{b}(t,x_d(t))u_d(t) \\ y_d(t) &= \boldsymbol{c}(t)x_d(t) \end{aligned} \tag{2}$$

It is worth pointing out that (2) is the well-known realizable condition for ILC [16,18,24]. Here, with the help of assumption on input/output coupling value, *i.e.*, A2, we can establish this realizable condition directly. However, due to the fact that nonlinear functions $f(\cdot,\cdot)$, $\boldsymbol{b}(\cdot,\cdot)$ and output coefficient vector $\boldsymbol{c}(\cdot)$ are unknown, the recursive defined optimal input $u_d(t)$ cannot be actually used, thus we have to design ILC update algorithms such that the generated input sequence converges to the optimal input.

A 3. The initial values can be precisely reset asymptotically in the sense that $x_k(0)\to x_d(0)$ as $k\to\infty$.

Remark 5. In many papers the initial state usually is required to be $x_d(0)$ [3,4]. In A3, it is required that the accurate initial state could be reset asymptotically. This is a technical condition, which aims to leave a space to design suitable initial value learning algorithms to realize this asymptotical re-initialization condition such as the one given in [25,26]. It is obvious that the classic identical initial condition (i.i.c.) is a special case of A3. For further discussions on initial condition, we refer to [27].

A 4. For each t the measurement noise $\{w_k(t), k=0,1,\cdots\}$ is a sequence of independent and identically distributed (i.i.d.) random variables with $\mathbb{E}w_k(t)=0$, $\sup_k \mathbb{E}\|w_k(t)\|^2<\infty$, and $\lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n} w_k(t)w_k^T(t)=R_w^t$ a.s., where R_w^t is an unknown matrix.

Remark 6. In A4, the condition on measurement noises is made according to the iteration axis, rather than the time axis. Thus this requirement is not rigorous as the process would be performed repeatedly and independently.

2.2 Data dropouts model

Similar to [12–17], here we adopt Bernoulli random variable to model the random data dropouts. To be specific, a random variable $\gamma_k(t)$ is introduced to indicate whether the measurement packet $y_k(t)$ is successfully transmitted or not. To be specific, $\gamma_k(t)=1$ if $y_k(t)$ is successfully transmitted and $\gamma_k(t)=0$ otherwise. Without loss of any generality,

$$\mathbb{P}(\gamma_k(t)=1)=\rho,\quad \mathbb{P}(\gamma_k(t)=0)=1-\rho \tag{3}$$

where $0<\rho<1$. That is, the probability that the measurement $y_k(t)$ is successfully transmitted is ρ, $\forall k,t$.

2.3 Control objective

Based on the above assumptions, the control objective of this paper is to design an ILC algorithm to generate the input sequence such that the the following averaged tracking index is minimized, $\forall t=0,1,\cdots,N$, under random data dropouts

$$V_t=\limsup_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\|y_d(t)-y_k(t)\|^2 \tag{4}$$

where $y_d(t)$ is the desired reference. If we define the control output as $z_k(t)=\boldsymbol{c}(t)x_k(t)$, then it is easy to see that $z_k(t)\to y_d(t)$ as $k\to\infty$ whenever the tracking index (4) is minimized and vice versa. That is, the index (4) implies that the precise tracking performance could be achieved if measurement noises are eliminated.

2.4 Updating algorithms

In this subsection, two ILC update laws are proposed to achieve the control objective under stochastic measurement noises, one of which is called intermittent ILC algorithm and the other one is called successive ILC algorithm.

Denote the tracking error $e_k(t) = y_d(t) - y_k(t)$.

- Intermittent ILC Algorithm (I-ILC):

$$u_{k+1}(t) = u_k(t) + a_k \gamma_k(t+1) e_k(t+1) \tag{5}$$

  where a_k is the learning step-size.
- Successive ILC Algorithm (S-ILC):

$$u_{k+1}(t) = u_k(t) + a_k e_k^*(t+1) \tag{6}$$

  where a_k has the same meaning to the I-ILC case, while $e_k^*(t)$ is the latest available tracking error, defined as

$$e_k^*(t) = \begin{cases} e_k(t), & \text{if } \gamma_k(t) = 1 \\ e_{k-1}^*(t), & \text{if } \gamma_k(t) = 0 \end{cases} \tag{7}$$

The learning step-size $\{a_k\}$ is a decreasing sequence and it should satisfy

$$a_k > 0, a_k \to 0, \sum_{k=1}^{\infty} a_k = \infty, \sum_{k=1}^{\infty} a_k^2 < \infty \tag{8}$$

Remark 7. It is clear that $a_k = a/k$ meets all these requirements, where $a > 0$ is a constant. This learning step-size a_k is introduced to suppress the effect of stochastic noises as iteration number goes to infinity and to guarantee a zero-error convergence of the input sequence. Notice that a_k decreases to zero, thus the learning procedure would become negligible after enough learning iterations. The reason for the design is as follows. The tracking error consists of two parts, the actual output error and measurement noise. At the beginning of the learning, it is believed that the actual output error would be dominant; while after enough learning iterations, the actual output error would be very small and the measurement noise may be dominant in the tracking error. Therefore, if no suppressing mechanism exists, the input sequence can be always changing due to random measurement noises. In order to avoid this unstable condition, the learning gain is designed to be decreasing. In addition, the learning gain derived by the Kalman filtering technique is also decreasing (see [28]), which coincides with our idea.

Remark 8. The reason why the first algorithm (5) is called intermittent ILC algorithm is that the algorithm only updates its signal when the output is successfully received. In other words, the input signal would stop updating if the corresponding output is lost. As a result, the algorithm (5) would update in some iterations and keep the latest one in other iterations. In addition, it is noticed that the updating frequency is equal to the successful transmission rate due to the inherent mechanism of (5). Therefore, roughly speaking, the larger the data dropout rate is, the slower the algorithm converges. This motivates us to find whether a faster convergence speed could be achieved under large data dropout rate.

Remark 9. Different from (5), the other algorithm (6) always keeps updating no matter whether the corresponding output is lost or not. If the output of the last iteration is received, then the algorithm would update its input by using this output; while if the output is lost, then the algorithm would update its input by using the latest available output information of certain previous iteration. As a matter of fact, the algorithm (6) is

$$\begin{aligned} u_{k+1}(t) = {} & u_k(t) + a_k \gamma_k(t+1) e_k(t+1) \\ & + a_k(1 - \gamma_k(t+1)) e_{k-1}^*(t+1) \end{aligned} \tag{9}$$

Therefore, the essential difference between I-ILC and S-ILC is that the former would stop updating if the corresponding output is lost while the latter will keep updating with available information.

Remark 10. In this paper, to make our idea for the convergence analysis clearer, we adopt the single input single output (SISO) formulation to reduce the expression complexity. However, the results can be easily extended to multiple input multiple output (MIMO) case with slight modification to the algorithms following similar steps given below. The major modification to the proposed algorithms is to multiply the tracking error from the left by a learning gain matrix L_t such that all eigenvalues of $L_t C(t+1) B(t,x)$ are with positive real parts, where $C(t+1)B(t,x)$ denotes the multi-dimensional input/output coupling matrix, *i.e.*, the counterpart of $\mathbf{c}(t+1)\mathbf{b}(t,x)$.

2.5 Preliminary lemmas

For simplicity of writing, let us set $f_k(t) = f(t, x_k(t))$, $f_d(t) = f(t, x_d(t))$, $\boldsymbol{b}_k(t) = \boldsymbol{b}(t, x_k(t))$, $\boldsymbol{b}_d(t) = \boldsymbol{b}(t, x_d(t))$, $\delta u_k(t) = u_d(t) - u_k(t)$, $\delta f_k(t) = f_d(t) - f_k(t)$, $\delta \boldsymbol{b}_k(t) = \boldsymbol{b}_d(t) - \boldsymbol{b}_k(t)$, and $\boldsymbol{c}^+ \boldsymbol{b}_k(t) = \boldsymbol{c}(t+1)\boldsymbol{b}_k(t)$.

For further analysis, the following lemmas are needed while the proofs are put in Appendix A.

Lemma 1. Assume A1–A3 hold for system (1). If $\lim_{k\to\infty} \delta u_k(s) = 0$, $s = 0, 1, \cdots, t$, then at time instance

$t+1$, $\|\delta x_k(t+1)\| \xrightarrow[k\to\infty]{} 0$, $\|\delta f_k(t+1)\| \xrightarrow[k\to\infty]{} 0$, $\|\delta \boldsymbol{b}_k(t+1)\| \xrightarrow[k\to\infty]{} 0$.

Lemma 2. Assume A1–A4 hold for system (1) and tracking reference $y_d(t)$, then the index (4) will be minimised for any arbitrary time $t+1$ if the control sequence $\{u_k(t)\}$ is admissible and satisfies $u_k(i) \xrightarrow[k\to\infty]{} u_d(i)$, $i = 0, 1, \cdots, t$. In this case, $\{u_k(t)\}$ is called the optimal control sequence.

III. CONVERGENCE OF THE PROPOSED UPDATE LAWS

3.1 Intermittent ILC algorithm case

In this subsection, the convergence analysis of the intermittent ILC algorithm (5) is given. Compared with (6), the proof of the I-ILC case would be more intuitive since it keeps its input invariant when the corresponding output is lost.

Recalling (3), we have that $\mathbb{E}\gamma_k(t) = \rho$ and $\mathbb{E}\gamma_k^2(t) = \rho$. Denote $\delta x_k(t) = x_d(t) - x_k(t)$ and $\delta u_k(t) = u_d(t) - u_k(t)$. Subtracting both side of (5) one has

$$\delta u_{k+1}(t) = \delta u_k(t) - a_k\gamma_k(t+1)e_k(t+1)$$

Notice that $e_k(t) = y_d(t) - y_k(t)$, then

$$\begin{aligned}\delta u_{k+1}(t) &= \delta u_k(t) - a_k\gamma_k(t+1)(y_d(t+1) - y_k(t+1))\\ &= \delta u_k(t) - a_k\gamma_k(t+1)\boldsymbol{c}(t+1)\delta x_k(t+1)\\ &\quad + a_k\gamma_k(t+1)w_k(t+1)\\ &= \delta u_k(t) - a_k\gamma_k(t+1)\boldsymbol{c}^+\boldsymbol{b}_k(t)\delta u_k(t)\\ &\quad - a_k\gamma_k(t+1)[\boldsymbol{c}^+\delta f_k(t) + \boldsymbol{c}^+\delta\boldsymbol{b}_k(t)u_d(t)]\\ &\quad + a_k\gamma_k(t+1)w_k(t+1)\end{aligned}$$

Then we have the following convergence theorem.

Theorem 1. Consider the stochastic system (1), index (4), and update law (5), assume assumptions A1–A4 hold, then the input $u_k(t)$ generated by (5) with learning gain sequence $\{a_k\}$ satisfying (8) converges to $u_d(t)$ almost surely as $k \to \infty$, $\forall t$.

The proof is in Appendix C.

Remark 11. As has been pointed out in Remark 8, the algorithm (5) only updates itself when the corresponding output package is well received. Thus, if the data dropout rate is large, then the learning step-size a_k during the updating iterations will decrease to zero fast, which will further lead to a slow convergence speed. To overcome this disadvantage, one could change the learning step-size only when the output is well received. In other words, the following algorithm is an alternative of (5),

$$\begin{aligned}u_{k+1}(t) &= u_k(t) + a_{\mu_k(t)}\gamma_k(t+1)e_k(t+1)\\ \mu_k(t) &= \sum_{i=1}^{k}\gamma_i(t+1)\end{aligned}$$

3.2 Successive ILC algorithm case

Now we come to the S-ILC case. Comparing with (5), the updating of (6) is deterministic in the sense that the algorithm updates itself every iteration. However, the technical proof of the convergence is more complex than that of the I-ILC case because the error information in (6) is no longer straightforward. As one could see, if the output of the last iteration is lost during transmission, then the error used in (6) is unknown for analysis because of successive data dropouts. That is, the error information could come from any previous iteration with different probabilities.

To form this situation, the stochastic stopping time sequence $\{\tau_k^t, k = 1, 2, \cdots, 0 \le t \le N\}$ is introduced to denote the random iteration-delay of the update due to random data dropouts. The algorithm (6) is reformulated as follows:

$$u_{k+1}(t) = u_k(t) + a_k e_{k-\tau_k^{t+1}}(t+1) \tag{10}$$

where the stopping time $\tau_k^{t+1} \le k$. In other words, for the updating of input at t of $(k+1)$-th iteration, no information of $e_m(t+1)$ with $m > k - \tau_k^{t+1}$ is received and only $e_{k-\tau_k^{t+1}}(t+1)$ is available. In addition, according to the S-ILC settings, for the m-th iteration with $k - \tau_k^{t+1} < m \le k$, the input $u_m(t)$ is successively updated with the same error $e_{k-\tau_k^{t+1}}(t+1)$.

For the convergence analysis, the major difficulty lies in the technical analysis of the influences caused by random iteration delays or stochastic stopping times τ_k^t. Therefore, the analysis is completed by two steps. The first step is to show the convergence of (10) without any iteration-delay, *i.e.*, $\tau_k^t = 0, \forall k, t$. The second step devotes to the effect of stopping times τ_k^t.

When there is no iteration-delay, *i.e.*, $\tau_k^t = 0$, the algorithm (10) turns into

$$u_{k+1}(t) = u_k(t) + a_k e_k(t+1) \tag{11}$$

This actually is the conventional ILC for systems without any data dropout. The convergence analysis of this algorithm could be derived directly following the similar steps of Theorem 1 by letting $\gamma_k(t) \equiv 1, \forall t, k$. Thus, we can give the following theorem without proof.

Theorem 2. Consider the stochastic system (1) without any data dropout, index (4), and update law (11), assume assumptions A1–A4 hold, then the input $u_k(t)$ generated by (11) with learning gain sequence $\{a_k\}$ satisfying (8) converges to $u_d(t)$ almost surely as $k \to \infty, \forall t$.

Now we are able to give the following convergence theorem for the S-ILC case.

Theorem 3. Consider the stochastic system (1), index (4), and update law (6), assume assumptions A1–A4 hold, then the input $u_k(t)$ generated by (6) with learning gain sequence $\{a_k\}$ satisfying (8) converges to $u_d(t)$ almost surely as $k \to \infty, \forall t$.

The proof is given in Appendix D.

Remark 12. The key step in the proof of the above theorem is to show that the effect of random data dropouts is asymptotically negligible. In other words, as the iteration number goes to infinity, the random iteration-delay is not with the same magnitude of iteration number. That is, the random iteration-delay is negligible comparing with the large enough iteration number. Consequently, the behaviors of S-ILC are close to those of the conventional learning algorithm (11) as iteration number increases.

IV. ILLUSTRATIVE SIMULATIONS

In order to show the effectiveness of the proposed ILC algorithm and verify the convergence analysis, a DC-motor driving a single rigid link through a gear is taken as an example [29]. The single-link mechanism is shown in Fig. 2, while the dynamics is expressed as the following second-order differential equation.

$$(J_m + \frac{J_l}{n^2})\ddot{\theta}_m + (B_m + \frac{B_l}{n^2})\dot{\theta}_m + \frac{Mgl}{n}\sin(\frac{\theta_m}{n}) = u \quad (12)$$

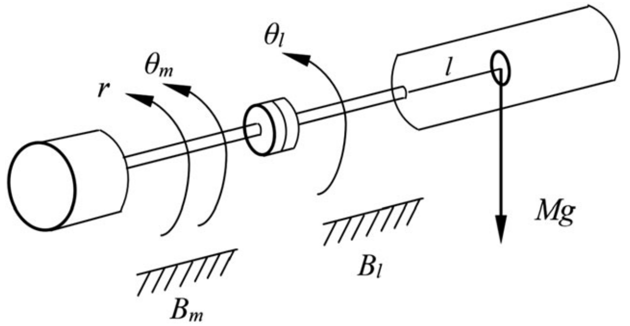

Fig. 2. Single-link mechanism.

Table I. Notations meaning of (12).

Notation	Meaning
J_m	motor inertia
B_m	motor damping coefficient
θ_m	motor angle
J_l	link inertia
B_l	link damping coefficient
θ_l	link angle, $\theta_l = \theta_m/n$
n	gear ratio
u	motor torque
M	lumped mass
g	gravitational acceleration
l	the center of mass from the axis of motion

where the notations are described in Table I.

By Euler's approximation, we have the discrete-time state-space expression with sate and output being $x = (x_1, x_2)^T = (\theta_m, \dot{\theta}_m)^T$ and $y = \dot{\theta}_l$, respectively and the system function and matrices are

$$f(x,t) = \begin{bmatrix} x_1(t) + \Delta x_2(t) \\ x_2(t) + \frac{\Delta}{J_m + J_l/n^2}\Big[-(B_m + \frac{B_l}{n^2})x_2(t) \\ -\frac{Mgl}{n}\sin\left(\frac{x_1(t)}{n}\right)\Big] \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{\Delta}{J_m + J_l/n^2} \end{bmatrix}, \quad C = \left[0, \frac{1}{n}\right]$$

where Δ is the discrete time interval. In this simulation, let $\Delta = 50ms$ and let the operation period be $3s$, thus iteration length is $N = 60$. Other parameters are given as follows: $J_m = 0.3$, $J_l = 0.44$, $B_m = 0.3$, $B_l = 0.25$, $M = 0.5$, $g = 9.8$, $n = 1.6$, and $l = 0.15$.

The desired trajectory is $y_d(t) = \frac{1}{3}\sin(t/20) + 1 - \cos(3t/20)$, $0 \le t \le 60$. The initial input, *i.e.*, input for the first iteration, is simply assumed to be $u_0(t) = 0$. The initial state is first fixed at $x_k(0) = [0, 0]^T$. The output is involved with a stochastic noise $w_k(t) \sim N(0, 0.1^2)$. The learning gain is set as $a_k = 5/k$. The algorithms have been run for 500 iterations.

We first set the probability as $\rho = 0.75$. In other words, for any given time instance, the data of about 25% iterations might be lost during transmission. To make expression simple, let $\gamma = 1 - \rho$ denote the data dropout rate. The tracking performance for the last iteration is shown in Fig. 3 with $\gamma = 0.25$, where the dotted line, solid line, and dashed line denote the desired reference, final outputs of I-ILC case and S-ILC case, respectively. It is seen that well tracking is achieved for both algorithms. Moreover, the maximal tracking errors, $\max_t |e_k(t)|$, are shown in Fig. 4 for both algorithms. It should be pointed

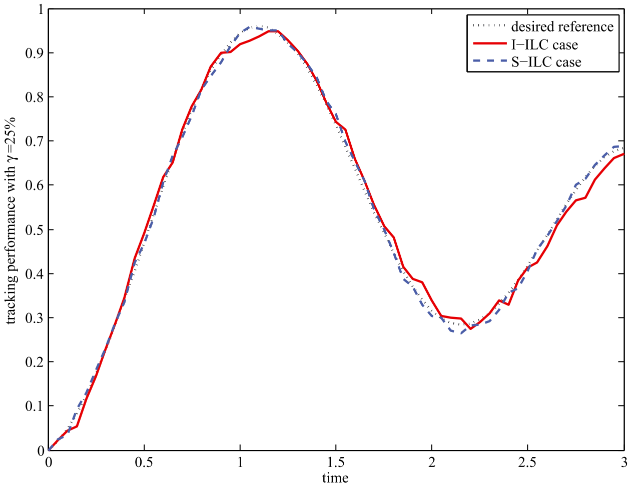

Fig. 3. Tracking Performance of I-ILC and S-ILC with $\gamma = 0.25$. [Color figure can be viewed at wileyonlinelibrary.com]

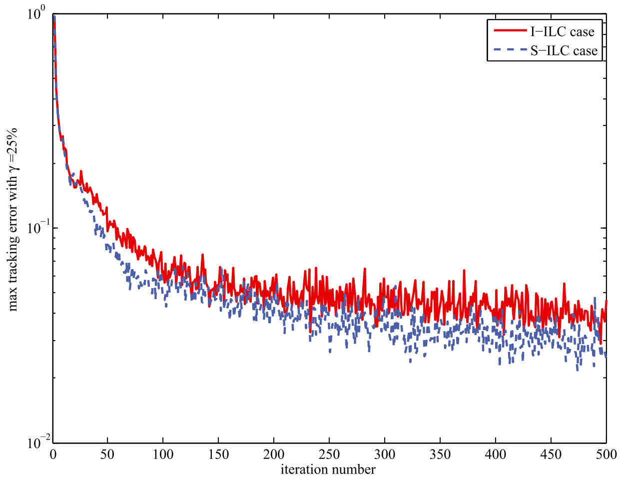

Fig. 4. Maximal errors of I-ILC and S-ILC along iteration axis with $\gamma = 0.25$. [Color figure can be viewed at wileyonlinelibrary.com]

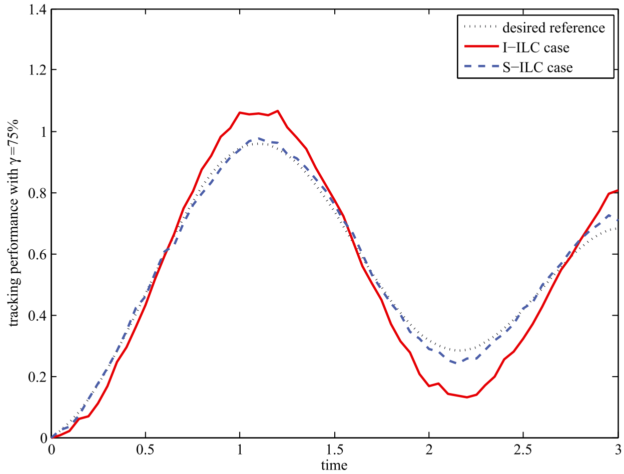

Fig. 5. Tracking performance of I-ILC and S-ILC with $\gamma = 0.75$. [Color figure can be viewed at wileyonlinelibrary.com]

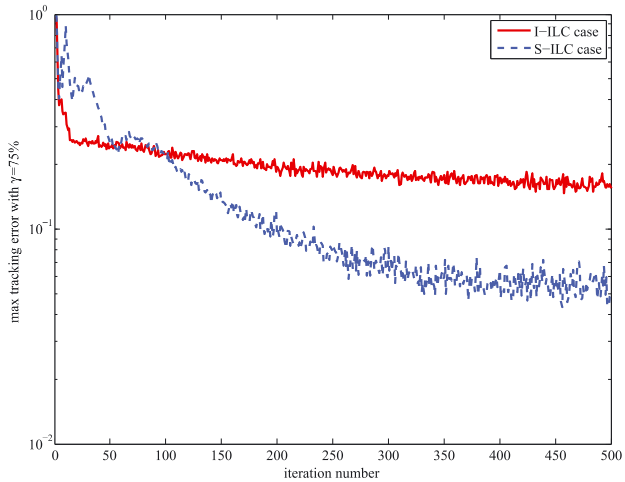

Fig. 6. Maximal errors of I-ILC and S-ILC along iteration axis with $\gamma = 0.75$. [Color figure can be viewed at wileyonlinelibrary.com]

out that due to the existence of stochastic noises, the maximal errors generally do not converge to zero. All results reveal that the performances of I-ILC and S-ILC are similar under low data dropout rate.

Next we set $\rho = 0.25$ or equivalently $\gamma = 0.75$ to further compare the performance of both schemes. It means the transmission function is rather bad. The final outputs of both algorithms are displayed in Fig. 5, while the maximal errors along iteration axis are shown in Fig. 6. It is noticed that under high data dropout rate, the S-ILC algorithm is superior to the I-ILC one. The inherent reason is that the I-ILC scheme would stop updating if the corresponding data is dropped while the S-ILC scheme keeps updating no matter whether the corresponding data is dropped. Thus the S-ILC scheme updates more iterations than the I-ILC scheme within the same iteration amount. Moreover, one may find from Fig. 6 that there is a trade-off between I-ILC and S-ILC schemes. However, this trade-off does not exist in Fig. 4. In conclusion, the trade-off is due to high data dropout rate. To be specific, when the data dropout rate is high and the learning gain is also large, the S-ILC may lead to slightly excessive updating, which further generates a large crest of its maximal error profile along iteration axis.

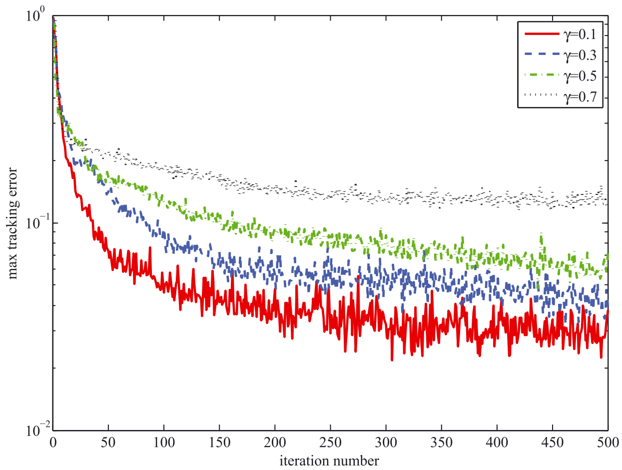

Fig. 7. Tracking performance comparison of I-ILC for different data dropout rate. [Color figure can be viewed at wileyonlinelibrary.com]

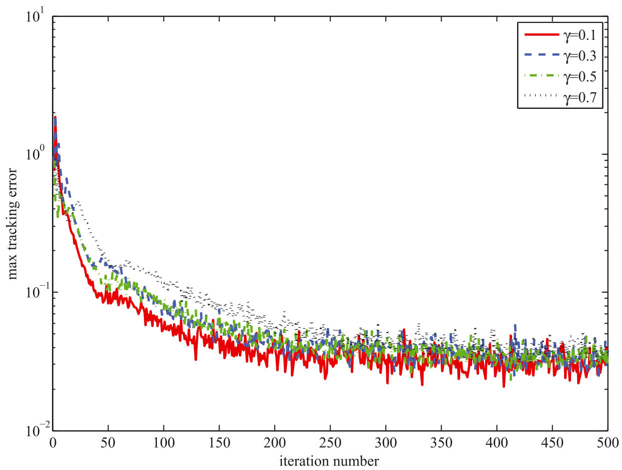

Fig. 8. Tracking performance comparison of S-ILC for different data dropout rate. [Color figure can be viewed at wileyonlinelibrary.com]

To see the influence of data dropout rate, we make comparisons for different data dropout rates. Here we consider for cases of different data dropout rates, 0.1, 0.3, 0.5, and 0.7. The I-ILC case and S-ILC case are shown in Fig. 7 and Fig. 8. It is seen from Fig. 7 that the tracking performance worsens as the data dropout rate increases for the I-ILC case at the same iteration. In contrast, the S-ILC scheme can maintain similar performance after several iterations even though the rate increases, as shown in Fig. 8.

V. CONCLUSIONS

In this paper, two data-driven algorithms, *i.e.*, the intermittent and successive ILC algorithms, are addressed for networked nonlinear systems with random data dropouts. Here, intermittent ILC algorithm only updates the input when the new packet is successfully received, while successive ILC algorithm would keep updating with latest available data no matter whether the data is dropped or not. The Bernoulli random variable is taken to describe the data dropout. Stochastic measurement noises are also considered. The almost sure convergence of the proposed algorithms for any time instance is strictly proved based on mathematical induction method. The simulation on DC-motor is given to verify the theoretical results. For further research, the case of the general nonlinear system is of interest. The detailed performance comparisons between both algorithms are also valuable for practical applications.

VI. APPENDIX A

6.1 Proof of Lemma1

The proof of this lemma can be carried out by induction along the time axis t. By (1) and (2),

$$\begin{aligned}\delta x_k(t+1) &= f_d(t) - f_k(t) + \boldsymbol{b}_d(t)u_d(t) - \boldsymbol{b}_k(t)u_k(t)\\ &= \delta f_k(t) + \delta\boldsymbol{b}_k(t)u_d(t) + \boldsymbol{b}_k(t)\delta u_k(t)\end{aligned} \tag{13}$$

Thus for $t = 0$, noticing A1 and A3, one has $\delta f_k(0) = f_d(0) - f_k(0) \xrightarrow[k\to\infty]{} 0$, $\delta\boldsymbol{b}_k(0) = \boldsymbol{b}_d(0) - \boldsymbol{b}_k(0) \xrightarrow[k\to\infty]{} 0$, which imply that the first two terms at the right-hand of (13) tend to zero as $k \to \infty$. Since $\|\boldsymbol{b}_k(0)\| \leq \|\boldsymbol{b}_d(0)\| + \|\delta\boldsymbol{b}_k(0)\|$, it follows that $\boldsymbol{b}_k(0)$ is bounded. Thus if $\delta u_k(0) \xrightarrow[k\to\infty]{} 0$, then the third term at the right-hand of (13) also tends to zero. It further implies that $\delta x_k(1) \xrightarrow[k\to\infty]{} 0$ and then by A1 again, $\delta f_k(1) \xrightarrow[k\to\infty]{} 0$ and $\delta\boldsymbol{b}_k(1) \xrightarrow[k\to\infty]{} 0$. That is, the conclusion is valid for $t = 0$.

Now assume the conclusions of the lemma are true for $s = 0, 1, \cdots, t-1$, it suffices to show that the conclusions hold for t, *i.e.*, $\|\delta x_k(t+1)\| \xrightarrow[k\to\infty]{} 0$, $\|\delta f_k(t+1)\| \xrightarrow[k\to\infty]{} 0$, $\|\delta\boldsymbol{b}_k(t+1)\| \xrightarrow[k\to\infty]{} 0$. This could be done through the same argument as used above. This completes the proof.

6.2 Proof of Lemma 2

Let $\mathcal{F}_k \triangleq \sigma\{y_j(t), x_j(t), w_j(t), 0 \leq j \leq k, t \in \{0,\cdots,N\}\}$ be the σ-algebra generated by $y_j(t)$, $x_j(t)$, $w_j(t)$, $0 \leq t \leq N$, $0 \leq j \leq k$. It is evident that $u_{k+1}(t) \in \mathcal{F}_k$ by noticing the design of update laws. According to A4 and the definition of $\mathcal{F}_k$, it follows that $\mathcal{F}_k$ is independent of $\{w_l(t), l = k+i, i = 1, 2, \cdots, \forall t\}$, thus $\{w_k(t), \mathcal{F}_k\}$ is a martingale difference sequence. Meanwhile, input, output, and state vectors are all adapted to $\mathcal{F}_k$. Therefore by (1) and A1

$$\begin{aligned}\limsup_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\|y_k(t)-y_d(t)\|^2 &= \limsup_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\|\boldsymbol{c}(t)(x_k(t)-x_d(t))+w_k(t)\|^2\\ &= \limsup_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\|\boldsymbol{c}(t)\delta x_k(t)\|^2(1+o(1))\\ &\quad+\limsup_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\|w_k(t)\|^2\\ &\geq \limsup_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\|w_k(t)\|^2 = R_w^t\end{aligned}$$

The sufficient and necessary condition to achieve the minimum is $\limsup_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\|\boldsymbol{c}(t)\delta x_k(t)\|^2=0$, which is true when $\boldsymbol{c}(t)\delta x_k(t)\xrightarrow[k\to\infty]{}0$. While the latter holds if $\delta u_k(s)\xrightarrow[k\to\infty]{}0$, $s=0,1,\cdots,t-1$ by Lemma 1. The proof is completed. □

VII. APPENDIX B

7.1 A technical lemma

The proof of the following technical lemma can be found in [30].

Lemma 3. Let $\{h_k\}$ be a sequence with $h_k\to h$ where h is a negative constant. Let a_k satisfy the conditions in (8) and both $\{\mu_k\}$ and $\{\nu_k\}$ satisfy the following conditions

$$\sum_{k=1}^{\infty}a_k\mu_k<\infty,\quad \nu_k\xrightarrow[k\to\infty]{}0 \tag{14}$$

then $\{\alpha_k\}$ generated by the following recursion with arbitrary initial value α_0 converges to zero a.s.

$$\alpha_{k+1}=\alpha_k+a_kh_k\alpha_k+a_k(\mu_k+\nu_k) \tag{15}$$

VIII. APPENDIX C

8.1 Proof of Theorem 1

The proof is carried out by mathematical induction along the time axis t. It should be indicated that the steps for $t=1,2,\cdots,N-1$ are identical to the case of $t=0$, which will be expressed in detail in the following.

Step 1 (Base Step). Consider the case $t=0$.

For $t=0$, the input error recursion could be rewritten as

$$\begin{aligned}\delta u_{k+1}(0)=&(1-a_k\rho\boldsymbol{c}^+\boldsymbol{b}_k(0))\delta u_k(0)\\ &-a_k(\gamma_k(1)-\rho)\boldsymbol{c}^+\boldsymbol{b}_k(0)\delta u_k(0)\\ &-a_k\gamma_k(1)[\boldsymbol{c}^+\delta f_k(0)+\boldsymbol{c}^+\delta\boldsymbol{b}_k(0)u_d(0)]\\ &+a_k\gamma_k(1)w_k(1)\end{aligned} \tag{16}$$

Note that $\boldsymbol{b}_k(0)$ is continuous in the initial state by A3, one has that $\boldsymbol{b}_k(0)\to\boldsymbol{b}_d(0)$ as $k\to\infty$ by A2. In addition, the coupling value $\boldsymbol{c}^+\boldsymbol{b}_k(0)$ would converge to $\boldsymbol{c}^+\boldsymbol{b}_d(0)$ by A2. Thus, it follows that $\rho\boldsymbol{c}^+\boldsymbol{b}_k(0)>\varepsilon$ for sufficient large k, say $k\geq k_0$, where $\varepsilon>0$ is a suitable constant.

Note that the first term on the right-hand side of (16) is the main recursion term, while the others are structural and measurement noises. According to Lemma 3 given in Appendix B, it is sufficient to show that these noises satisfy the condition (14).

By A1 and A3, it is easy to derive that $\delta f_k(0)\xrightarrow[k\to\infty]{}0$ and $\delta\boldsymbol{b}_k(0)\xrightarrow[k\to\infty]{}0$. Notice that both $\gamma_k(1)$ and $u_d(0)$ are bounded. Therefore, the third term on the right-hand side of (16) converges to 0 as $k\to\infty$.

Further, the sequence $\{w_k(1)\}$ is an i.i.d. sequence with zero mean and finite second moments. In addition, $w_k(1)$ is independent of $\gamma_k(1)$. Thus, it is obvious that $\sum_{k=1}^{\infty}\mathbb{E}[a_k\gamma_k(1)w_k(1)]^2\leq\sup_k\mathbb{E}w_k^2(1)\cdot\mathbb{E}\gamma_k^2(1)\sum_{k=1}^{\infty}a_k^2\leq\rho\|R_w^1\|\sum_{k=1}^{\infty}a_k^2<\infty$ where $\|\cdot\|$ is a suitable matrix norm. This further leads to that $\sum_{k=1}^{\infty}a_k\gamma_k(1)w_k(1)<\infty$, a.s. by Khintchine-Kolmogorov convergence theorem [31]. In other words, the last term of (16) satisfies (14).

Now it comes to the second term on the right-hand side of (16), $a_k(\gamma_k(1)-\rho)\boldsymbol{c}^+\boldsymbol{b}_k(0)\delta u_k(0)$. The sequence of this term is no longer mutual independent. To deal with this term, let $\mathcal{F}_k$ be the increasing σ-algebra generated by $y_j(t)$, $w_j(t)$, $\gamma_j(t)$, $x_j(0)$, $0\leq j\leq k$, $\forall t$. That is, $\mathcal{F}_k\triangleq\sigma\{y_j(t), w_j(t), \gamma_j(t), x_j(0), 0\leq j\leq k, \forall t\}$. Then according to the learning law (5), it is easy to find that $u_k(t)\in\mathcal{F}_{k-1}$ and $\boldsymbol{b}_k(0)\in\mathcal{F}_{k-1}$. In addition, $\gamma_k(1)$ is independent of $\mathcal{F}_{k-1}$ and thus is independent of $\delta u_k(0)$ and $\boldsymbol{b}_k(0)$. Therefore, $\mathbb{E}\{(\gamma_k(1)-\rho)\boldsymbol{c}^+\boldsymbol{b}_k(0)\delta u_k(0)|\mathcal{F}_{k-1}\}=\boldsymbol{c}^+\boldsymbol{b}_k(0)\delta u_k(0)\mathbb{E}\{\gamma_k(1)-\rho|\mathcal{F}_{k-1}\}=0$. This means that $((\gamma_k(1)-\rho)\boldsymbol{c}^+\boldsymbol{b}_k(0)\delta u_k(0),\mathcal{F}_k,k\geq 1)$ is a martingale difference sequence [31]. In addition, $\sum_{k=1}^{\infty}\mathbb{E}\{[a_k(\gamma_k(1)-\rho)\boldsymbol{c}^+\boldsymbol{b}_k(0)\delta u_k(0)]^2|\mathcal{F}_{k-1}\}\leq\sup_k[\boldsymbol{c}^+\boldsymbol{b}_k(0)\delta u_k(0)]^2\sum_{k=1}^{\infty}a_k^2\mathbb{E}\{(\gamma_k(1)-\rho)^2|\mathcal{F}_{k-1}\}\leq c_1\sum_{k=1}^{\infty}a_k^2<\infty$ where $c_1>0$ is a suitable constant. Then by Chow convergence theorem of martingale

[31], we have $\sum_{k=1}^{\infty} a_k(\gamma_k(1)-\rho)\boldsymbol{c}^+\boldsymbol{b}_k(0)\delta u_k(0) < \infty$. In other words, the second term on the right-hand side of (16) satisfies (14).

Then applying Lemma 3 in Appendix B to (16), we are now able to have that $\delta u_k(0) \to 0$ as $k \to \infty$ a.s.

Step 2 (Inductive Step). Assume that the convergence of $u_k(t)$ has been proved for $t = 0, 1, \cdots, s-1$ and the target is to show the convergence for $t = s$. From the inductive assumptions and Lemma 1, we have $\delta x_k(s) \xrightarrow[k\to\infty]{} 0$ and therefore $\delta f_k(s) \xrightarrow[k\to\infty]{} 0$ and $\delta \boldsymbol{b}_k(s) \xrightarrow[k\to\infty]{} 0$. On the other hand, the recursion for $t = s$ is as follows.

$$\begin{aligned}\delta u_{k+1}(s) = \delta u_k(s) &- a_k\gamma_k(s+1)\boldsymbol{c}^+\boldsymbol{b}_k(s)\delta u_k(s)\\ &- a_k\gamma_k(s+1)[\boldsymbol{c}^+\delta f_k(s) + \boldsymbol{c}^+\delta\boldsymbol{b}_k(s)u_d(s)]\\ &+ a_k\gamma_k(s+1)w_k(s+1)\end{aligned}$$

Then following similar steps of the case $t = 0$, we are with no further efforts to conclude that $\delta u_k(s) \to 0$ as $k \to \infty$ a.s.. This completes the proof.

IX. APPENDIX D

9.1 Proof of Theorem 3

Comparing (10) and (11), we find that the effect of the random data dropouts acts as an additional error $e_{k-\tau_k^{t+1}}(t+1) - e_k(t+1)$. Taking the main idea of the proof for convergence into account and recalling the preliminary result of Theorem 2, it is sufficient to show that this error satisfies the condition (14). Specifically, we have

$$\begin{aligned}&e_{k-\tau_k^{t+1}}(t+1) - e_k(t+1)\\ &= y_k(t+1) - y_{k-\tau_k^{t+1}}(t+1) + w_k(t+1) - w_{k-\tau_k^{t+1}}(t+1)\\ &= \boldsymbol{c}^+\boldsymbol{b}_k(t)[u_k(t) - u_{k-\tau_k^{t+1}}(t)] + [\boldsymbol{c}^+f_k(t) - \boldsymbol{c}^+f_{k-\tau_k^{t+1}}(t)]\\ &\quad + [\boldsymbol{c}^+\boldsymbol{b}_k(t) - \boldsymbol{c}^+\boldsymbol{b}_{k-\tau_k^{t+1}}(t)]u_{k-\tau_k^{t+1}}(t)\\ &\quad + w_k(t+1) - w_{k-\tau_k^{t+1}}(t+1)\end{aligned} \tag{17}$$

There is no doubt that the last term satisfies the condition (14). In addition, it could be proved by mathematical induction, similar to the proofs of Theorem 1, that the second and the third terms on the right-hand side of (17) satisfy the condition (14) with the help of Lemma 1. Thus only the first term, *i.e.*, $\boldsymbol{c}^+\boldsymbol{b}_k(t)[u_k(t) - u_{k-\tau_k^{t+1}}(t)]$, is left for further analysis. It is also easy to prove boundedness and convergence of $\boldsymbol{c}^+\boldsymbol{b}_k(t)$ by the mathematical induction principle.

Recalling the learning algorithm (10), we find that the difference is expanded as

$$\begin{aligned}u_k(t) - u_{k-\tau_k^{t+1}}(t) = &\sum_{m=k-\tau_k^{t+1}}^{k-1} a_m\boldsymbol{c}^+\boldsymbol{b}_m(t)\delta u_{m-\tau_m^{t+1}}(t)\\ &+ \sum_{m=k-\tau_k^{t+1}}^{k-1} a_m\boldsymbol{c}^+\delta f_{m-\tau_m^{t+1}}(t)\\ &- \sum_{m=k-\tau_k^{t+1}}^{k-1} a_m w_{m-\tau_m^{t+1}}(t+1)\end{aligned} \tag{18}$$

In order to analyze the effect of (18), we need to give an estimation on the number of successive data dropout iterations, *i.e.*, τ_k^t. Noticing that the data dropouts are modeled by a Bernoulli random variable, we find that τ_k^t obeys the geometric distribution. Here, for concise notations, we let τ denote a random variable satisfying the same distribution, *i.e.*, $\tau \sim G(\rho)$. Then it is obvious that $\mathbb{E}\tau = 1/\rho$ and $\mathrm{Var}(\tau) = (1-\rho)/\rho^2$. Then we further have that $\mathbb{E}\tau^2 = 1/\rho$. Using direct calculations, we have that $\sum_{n=1}^{\infty}\mathbb{P}\{\tau \geq n^{\frac{1}{2}}\} = \sum_{n=1}^{\infty}\mathbb{P}\{\tau^2 \geq n\} = \sum_{n=1}^{\infty}\sum_{j=n}^{\infty}\mathbb{P}\{j \leq \tau^2 < j+1\} = \sum_{j=1}^{\infty} j\mathbb{P}\{j \leq \tau^2 < j+1\} \leq \mathbb{E}\tau^2 < \infty$. By the Borel-Cantelli lemma, it further leads that $\mathbb{P}\{\tau \geq n^{\frac{1}{2}}\ i.o.\} = 0$. Consequently, we have $\tau_n^t/n \xrightarrow[n\to\infty]{} 0$, a.s., $\forall t$. That is, $(n-\tau_n^t)/n \xrightarrow[n\to\infty]{} 1$ and $n - \tau_n^t \to \infty$, a.s., $\forall t$.

Based on this observation, now we can prove that the terms on the right-hand side of (18) satisfy the condition (14).

Using similar steps of the proof of Theorem 1, it is concluded that $\sum_{m=0}^{k} a_m w_{m-\tau_m^{t+1}}(t+1)$ converges to an unknown constant, a.s., $\forall t$. Therefore, noticing that $n - \tau_n^t \to \infty$, we have that $\| \sum_{m=k-\tau_k^{t+1}}^{k-1} a_m w_{m-\tau_m^{t+1}}(t+1)\| = o(1)$. This further leads that the last term of (18) satisfy the condition (14). On the other hand, by mathematical induction principle, it can be proved that the state function error, $\delta f_{m-\tau_m^{t+1}}(t)$, in the second term on the right-hand side of (18), converges to zero as iteration number goes to infinity. That is, the condition (14) is also satisfied for the second term.

Therefore, only the first term on the right-hand side of (18) is left to discuss. As a matter of fact, this term can be almost surely bounded by a sample path dependent constant timing $\sum_{m=k-\tau_k^{t+1}}^{k-1} a_m$ by noticing A1. Further, the selection of a_k leads to that this term is bounded by $c_0 a_{k-\tau_k^{t+1}}\tau_k^{t+1}$ where c_0 is a suitable constant. Thus it is sufficient to show that $c_0 a_{k-\tau_k^{t+1}}\tau_k^{t+1} = o(1)$. For expressions easy to understand, here we select $a_k = 1/k$. The general case is similar but with more complicated derivations on the quantity estimation. For this case, we can directly calculate the term as follows. $a_{k-\tau_k^{t+1}}\tau_k^{t+1} = 1/(k-\tau_k^{t+1})\cdot\tau_k^{t+1} = k/(k-\tau_k^{t+1})\times\tau_k^{t+1}/k = O(1/k)O(k^{\frac{1}{2}}) = O(1/k^{\frac{1}{2}}) \to 0$ as $k\to\infty$. Therefore, the first term is also verified.

In sum, we have proved that the effect of random data dropouts, *i.e.*, (17), satisfies the condition (14). Then the convergence proof of this theorem could be completed by similar steps of Theorem 1.

REFERENCES

1. Arimoto, S., S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robotic Syst.*, Vol. 1, No. 2, pp. 123–140 (1984).
2. Bristow, D. A., M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control: A learning-based method for high-performance tracking control," *IEEE Control Syst. Mag.*, Vol. 26, No. 3, pp. 96–114 (2006).
3. Ahn, H. S., 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).
4. Shen, D. and Y. Wang, "Survey on stochastic iterative learning control," *J. Process Control*, Vol. 24, No. 12, pp. 64–77 (2014).
5. Bifaretti, S., P. Tomei, and C. M. Verrelli, "A global robust iterative learning position control for current-fed permanent magnet step motors," *Automatica*, Vol. 47, No. 1, pp. 227–234 (2011).
6. Xu, W., B. Chu, and E. Rogers, "Iterative learning control for robotic-assisted upper limb stroke rehabilitation in the presence of muscle fatigue," *Control Eng. Practice*, Vol. 31, pp. 63–72 (2014).
7. Zhao, Y., Y. Lin, F. Xi, and S. Guo, "Calibration-based iterative learning control for path tracking of industrial robots," *IEEE Trans. Ind. Electron.*, Vol. 62, No. 5, pp. 2921–2929 (2015).
8. Gupta, R. A. and M.-Y. Chow, "Networked control system: overview and research trends," *IEEE Trans. Ind. Electron.*, Vol. 57, No. 7, pp. 2527–2535 (2010).
9. Sinopoli, B., L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, Vol. 49, No. 9, pp. 1453–1464 (2004).
10. Xu, H., S. Jagannathan, and F. L. Lewis, "Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses," *Automatica*, Vol. 48, No. 6, pp. 1017–1030 (2012).
11. Hespanha, J. P., P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. IEEE*, Vol. 95, No. 1, pp. 138–162 (2007).
12. Ahn H. S., Y. Q. Chen, and K. L. Moore, "Intermittent iterative learning control," *Proc. IEEE Int. Symp. Intell. Control*, Munich, Germany, pp. 832–837 (2006).
13. Ahn, H. S., K. L. Moore, and Y. Q. Chen, "Discrete-time intermittent iterative learning controller with independent data dropouts," *Proc. 17th IFAC World Congr.*, Coex, South Korea, pp. 12442–12447 (2008).
14. Ahn, H. S., 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," *Proc. 10th Int. Conf. Control Autom. Robotics Vision*, Hanoi, Vietnam, pp. 757–762 (2008).
15. Bu, X., Z.-S. Hou, and F. Yu, "Stability of first and high order iterative learning control with data dropouts," *Int. J. Control Autom. Syst.*, Vol. 9, No. 5, pp. 843–849 (2011).
16. Bu, X., 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).
17. Bu, X., 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).
18. Shen, D. 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).
19. Shen, D. 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).
20. Zhao, D., Z. Xia, and D. Wang, "Model-free optimal control for affine nonlinear systems with convergence analysis," *IEEE Trans. Autom. Sci. Eng.*, Vol. 12, No. 4, pp. 1461–1468 (2015).
21. Yu, M., D. Huang, and W. He, "Robust adaptive iterative learning control for discrete-time nonlinear systems with both parametric and nonparametric uncertainties," *Int. J. Adapt. Control Signal Process.*, Vol. 30, pp. 972–985 (2016).
22. Zhao, Q., H. Xu, and S. Jagannathan, "Neural network-based finite-horizon optimal control of uncertain affine nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, Vol. 26, No. 3, pp. 486–499 (2015).

23. Sun, M. and D. Wang, “Analysis of nonlinear discrete-time systems with higher-order iterative learning control,” *Dyn. Control*, Vol. 11, pp. 81–96 (2001).
24. Tan, Y., H.-H. Dai, D. Huang, and J.-X. Xu, “Unified iterative learning control schemes for nonlinear dynamic systems with nonlinear input uncertainties,” *Automatica*, Vol. 48, No. 12, pp. 3173–3182 (2012).
25. Chen, Y., C. Wen, Z. Gong, and M. Sun, “An iterative learning controller with initial state learning,” *IEEE Trans. Autom. Control*, Vol. 44, No. 2, pp. 371–376 (1999).
26. Yang, S., J.-X. Xu, and D. Huang, “Iterative learning control for multi-agent systems consensus tracking,” *Proc. 51st IEEE Conf. Decis. Control*, Maui, HI, USA, pp. 4672–4677 (2012).
27. Xu, J.-X. and R. Yan, “On initial conditions in iterative learning control,” *IEEE Trans. Autom. Control*, Vol. 50, No. 9, pp. 1349–1354 (2005).
28. Saab, S. S., “A discrete-time stochastic learning control algorithm,” *IEEE Trans. Autom. Control*, Vol. 46, No. 6, pp. 877–887 (2001).
29. Wang, D., “Convergence and robustness of discrete time nonlinear systems with iterative learning control,” *Automatica*, Vol. 34, No. 11, pp. 1445–1448 (1998).
30. Chen, H. F., *Stochastic Approximation and Its Applications*, Kluwer, Dordrecht (2002).
31. Chow Y. S. and H. Teicher, *Probability Theory: Independence, Interchangeability, Martingales*, Springer Verlag, New York (1978).

Dong Shen received the B.S. degree in mathematics from Shandong University, Jinan, China, in 2005. He received the Ph.D. degree in mathematics from the Academy of Mathematics and System Science, Chinese Academy of Sciences (CAS), Beijing, China, in 2010. From 2010 to 2012, he was a Post-Doctoral Fellow with the Institute of Automation, CAS. Since 2012, he has been an associate professor with College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China. His current research interests include iterative learning control, stochastic control and optimization. He has published more than 40 refereed journal and conference papers. He is the author of *Stochastic Iterative Learning Control* (Science Press, 2016, in Chinese) and co-author of *Iterative Learning Control for Multi-Agent Systems Coordination* (Wiley, 2017). Dr. Shen received IEEE CSS Beijing Chapter Young Author Prize in 2014 and Wentsun Wu Artificial Intelligence Science and Technology Progress Award in 2012.

Chao Zhang received the B.S. degree in automation from Beijing University of Chemical Technology, Beijing, China, in 2016. Now he is pursuing a M.S. degree at Beijing University of Chemical Technology. His research interests include iterative learning control and its applications on motion robots.

Yun Xu received the B.S. degree in automation from Beijing Institute of Petrochemical Technology, China, in 2014. Now she is pursuing a M.S. degree at College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China. Her research interests are in the area of sampled-data iterative learning control and adaptive iterative learning control.