International Journal of Control, Automation and Systems 15(2) (2017) 647-655
http://dx.doi.org/10.1007/s12555-015-0273-y
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Almost Sure Convergence of ILC for Networked Linear Systems with Random Link Failures

Dong Shen

Abstract: The iterative learning control (ILC) problem for networked linear system with random link failure is addressed in this paper. The link failures arise both from the controller to the plant and from the plant to the controller. The random link failure is modeled by an independent Bernoulli random variable. Two P-type update laws are designed and critically analyzed. The almost sure convergence property is established according to this case for the first time. Illustrative simulations show the effectiveness of the proposed algorithms.

Keywords: Almost sure convergence, iterative learning control, link failures, networked linear systems.

1. INTRODUCTION

In our daily lives, one knows that a task could be completed better and better if we could repeat the task again and again. This is a basic principle of learning, which motivates us to design control algorithms that could learn from trying and correcting. Iterative learning control (ILC) is such a kind control strategy, greatly suitable for repetitive systems that could complete some given tracking reference in a finite time interval and repeat the process [1–3]. For these repetitive systems, it is called an iteration when the system completes the whole tracking reference once. ILC generates control input for the current iteration using the input and tracking error information from previous iterations, and thus the tracking performance could be improved asymptotically along the iteration axis. This inherent mechanism mimics human learning principle as one could learn from experiences. However, it should be specifically pointed out that much repeatability is usually required for this control method. Thus one would like to explore how to maintain an acceptable performance under harsh conditions.

On the other hand, networked control systems (NCS) are widely implemented in practical applications due to the flexibility, facility, and robustness with the help of fast developments of communication and network technology. However, in NCS, the controller and the plant are placed in different sites and communicated through wire or wireless networks. Thus the communication links from controller to plant and from plant to controller may sustain random link failures. Then the data during failure period could not be transmitted successfully and thus might cause critical influence of the control performance. In addition, the link failures arise randomly, which motivates us to study stochastic ILC [4].

As one could see, the link failure could be formulated as a data dropout problem which has attracted some preliminary research. In most previous ILC literature, a Bernoulli random variable is used to describe the randomness of data dropout. Ahn *et al* gives a first attempt on this topic [5–7]. In [5], the measurement output was assumed to be randomly lost during transmission from plant to the controller. It was required that the output vector should be lost or successfully transmitted as a whole. While the case that only part of the output vector is randomly lost was handled in [6]. Then [7] proceeded to the case that data dropouts happened to the control signals as well as output signals. Overall, in these results, mean square convergence of input sequences is derived by using the Kalman filtering techniques.

Bu *et al*. considered ILC under data dropout from the statistics point of view [8–10]. In [8], the ILC for linear time invariant system under data dropout was discussed and the stability analysis was given by taking mathematical expectations to both sides of the iteration equation of tracking errors directly, so that the randomness in iterative evolution were eliminated. The nonlinear system case was addressed in [9] and [10] with similar techniques except replacing iteration equation by iteration inequality. In short, the mathematical expectation is first taken to iterative equations and then the analysis is given based on the deterministic recursion in these reports. Moreover, it should be pointed out that data dropout only happens at the measurement side in [2,6,8,9] while the case that data

Manuscript received July 12, 2015; revised March 1, 2016; accepted April 5, 2016. Recommended by Associate Editor Do Wan Kim under the direction of Editor PooGyeon Park. This work is supported by National Natural Science Foundation of China (61304085), Beijing Natural Science Foundation (4152040).

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

dropout at the input side was attempted in [7, 10].

In sum, the convergence in the sense of mean square and mathematical expectation was obtained with suitable design of ILC algorithms in the above publications. However, is it possible to derive almost sure convergence for a simple ILC algorithm? In [11, 12], the authors adopted the conventional P-type ILC algorithm to handle nonlinear systems under data dropout. An arbitrary stochastic sequence model with finite length requirement was used to express the random data dropout. The almost sure convergence analysis was given following stochastic approximation techniques. However, it is noticed that the model on data dropout hints that it is not totally stochastic. In other words, the randomness is limited due to technical analysis [12]. Thus we would like to relax this requirement in this note. Similar problems are discussed in other topics, such as multirate multiple-access wireless system [13], multi-agent networks with random directional link failure [14], and stochastic synchronization for Markovian coupled neural networks [15]. These papers show a potential background of this note, where the ILC topic is considered.

This note considers ILC for networked linear systems with random link failures. The link failures arise both from controller to plant and from plant to controller. The random link failure is modeled by Bernoulli random variable, which is totally stochastic. The conventional P-type update law is used in this paper with suitable selection of learning gain matrix. The almost sure convergence property is built for this simplest algorithm, which reveals that the P-type ILC algorithm possesses well robustness against to harsh conditions. Our contribution differs from [5–10] in terms of convergence sense and differs from [11, 12] in terms of random model.

The rest of the paper is arranged as follows. Section 2 gives the system setup and problem formulation; Section 3 provides convergence analysis for the case that only link failure at the measurement side is considered; the general case that link failures at both the measurement side and control side are discussed in Section 4; Section 5 makes illustrative simulations to verify theoretical analysis; some concluding remarks are presented in Section 6.

2. PROBLEM FORMULATION

Consider the following MIMO linear discrete system

$$\begin{aligned} x_k(t+1) &= A(t)x_k(t) + B(t)u_k(t), \\ y_k(t) &= C(t)x_k(t), \end{aligned} \tag{1}$$

where $t = 0, 1, \cdots, N$ denotes the time instant of an iteration and $k = 1, 2, \cdots$ labels different iteration number. $u_k(t) \in \mathbb{R}^p$, $x_k(t) \in \mathbb{R}^n$, and $y_k(t) \in \mathbb{R}^q$ denote the input, state, and output, respectively. $A(t)$, $B(t)$, and $C(t)$ are system matrices with appropriate dimensions.

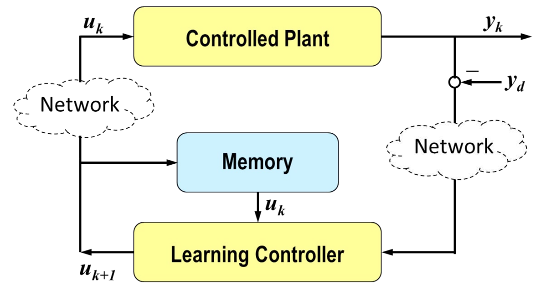

Fig. 1. Block diagram of networked ILC.

Denote $y_d(t)$, $t \in \{0, 1, \cdots, N\}$ as the reference trajectory. Let the tracking error be $e_k(t) = y_d(t) - y_k(t)$. Then the conventional P-type learning controller is given as follows

$$u_{k+1}(t) = u_k(t) + L_t e_k(t+1), \tag{2}$$

where L_t is a learning gain matrix with appropriate dimension. However, in many practical applications, the controlled plant and learning controller are placed in different sites and communicated by networks with each other as illustrated in Fig. 1. In such kind of implementation, the plant is controlled by signals that are transmitted through the wire or wireless network. If the control signal is successfully transmitted, there is an input coming into the system, otherwise there is no input or the input is zero. Due to complex network conditions, the links from controller to plant and from plant to controller would suffer random failures. In order to model the link failures, we introduce two random variables, $\gamma_k(t)$ and $\sigma_k(t)$, for the above two cases, respectively. Both $\gamma_k(t)$ and $\sigma_k(t)$ satisfy binary Bernoulli distribution. That is, $\forall t, k$, $\gamma_k(t) = 1$ if no link failure from controller to plant, otherwise $\gamma_k(t) = 0$, while $\sigma_k(t) = 1$ if no link failure from plant to controller, otherwise $\sigma_k(t) = 0$. In addition, $\mathbb{P}\{\gamma_k(t) = 1\} = \overline{\gamma}(t)$, $\mathbb{P}\{\gamma_k(t) = 0\} = 1 - \overline{\gamma}(t)$, $\mathbb{P}\{\sigma_k(t) = 1\} = \overline{\sigma}(t)$, $\mathbb{P}\{\sigma_k(t) = 0\} = 1 - \overline{\sigma}(t)$, $0 < \overline{\gamma}(t) < 1$, $0 < \overline{\sigma}(t) < 1$. Here $\mathbb{P}\{\cdot\}$ denotes probability of an event.

Then the networked control system is

$$\begin{aligned} x_k(t+1) &= A(t)x_k(t) + \gamma_k(t)B(t)u_k(t), \\ y_k(t) &= C(t)x_k(t), \end{aligned} \tag{3}$$

and the P-type learning update law (2) is rewritten as

$$u_{k+1}(t) = u_k(t) + \sigma_k(t)L_t e_k(t+1). \tag{4}$$

The control purpose is to find the control sequence $\{u_k(t), k = 1, 2, \cdots\}$ under random link failures such that the reference trajectory is asymptotically tracked. Moreover, one would like to address the tracking performance of the conventional P-type algorithm (4) under random failures. In addition, the relationship between convergence speed and failure rate is also an interesting problem.

Without loss of any generality, we assume that $C(t+1)B(t)$ is of full-column rank for all t. This further implies that the relative degree is one and the dimension of input is not larger than the dimension of output. Besides, let $y_d(t), t \in \{0,1,\cdots,N\}$ be realizable, which means that there exists a control $u_d(t)$ and an initial state value $x_d(0)$ such that

$$\begin{aligned} x_d(t+1) &= A(t)x_d(t) + B(t)u_d(t), \\ y_d(t) &= C(t)x_d(t). \end{aligned} \tag{5}$$

Then it is easy to obtain

$$\begin{aligned} u_d(t) =&[(C(t+1)B(t))^T C(t+1)B(t)]^{-1}(C(t+1)B(t))^T \\ &\times (y_d(t+1) - C(t+1)A(t)x_d(t)). \end{aligned} \tag{6}$$

For notations concise, denote $C^+B(t) \triangleq C(t+1)B(t)$ in the rest of the paper. Sometimes we would omit the argument t from the expressions when no confusion arises.

3. LINK FAILURES AT MEASUREMENT SIDE

In this section, we consider the case that link failures only happen at the measurement side, i.e., only the network from plant to controller may interrupt randomly, while the network from controller to plant works well. That is, $\gamma_k(t) = 1, \forall k,t$.

In order to analyze the convergence of the traditional P-type algorithm (4) under random link failures, we first rewrite the system into a super-vector form by the so-called lifting technique as follows,

$$\begin{aligned} U_k &= [u_k^T(0), u_K^T(1), \cdots, u_k^T(N-1)]^T \in \mathbb{R}^{pN}, \\ Y_k &= [y_k^T(1), y_K^T(2), \cdots, y_k^T(N)]^T \in \mathbb{R}^{qN}. \end{aligned}$$

Let H be given in equation (7), where $\prod_{i=n}^{m} A(i) \triangleq A(m)\cdots A(n)$. Then we have

$$Y_k = \mathbf{H}U_k + Y_k^0, \tag{8}$$

where $Y_k^0 = [(C(1)A(0)x_k(0))^T,\ (C(2)A(1)A(0)x_k(0))^T, \cdots, (C(N)\prod_{i=0}^{N-1} A(i)x_k(0))^T]^T$ is the initial value. Using similar derivation we have

$$Y_d = \mathbf{H}U_d + Y_d^0, \tag{9}$$

where Y_d, U_d, and Y_d^0 are defined similar to Y_k, U_k, and Y_k^0 by replacing k with d. For expression concise, it is always assumed that $Y_k^0 = Y_d^0$ in this note. In other words, the system is identically reset for all the iterations. Then we lift the tracking error as E_k and obviously, $E_k = Y_d - Y_k$.

Denote

$$\Sigma_k = \operatorname{diag}\{\sigma_k(1)I_{q\times q}, \sigma_k(2)I_{q\times q}, \cdots, \sigma_k(N)I_{q\times q}\}, \tag{10}$$

and

$$\mathbf{L} = \operatorname{diag}\{L_0, L_1, \cdots, L_{N-1}\}. \tag{11}$$

Then the update law (4) could be lifted as

$$U_{k+1} = U_k + \mathbf{L}\Sigma_k E_k. \tag{12}$$

Theorem 1: Consider system (8) and update law (12). If the following condition is satisfied

$$\rho(I - L_t C^+ B(t)) < 1 \tag{13}$$

for all t, where $\rho(M)$ denotes the spectral radius of a matrix M, then the input U_k would converge almost surely.

Proof: Subtracting both sides of (12) from U_d and denoting $\delta U_k \triangleq U_d - U_k$, one has

$$\begin{aligned} \delta U_{k+1} =&\delta U_k - \mathbf{L}\Sigma_k E_k \\ =&\delta U_k - \mathbf{L}\Sigma_k \mathbf{H}\delta U_k \\ =&(I - \mathbf{L}\Sigma_k \mathbf{H})\delta U_k. \end{aligned} \tag{14}$$

Note that both $\mathbf{L}$ and Σ_k are block-diagonal matrix, while $\mathbf{H}$ is a block-lower-triangular matrix, thus the product $\mathbf{L}\Sigma_k\mathbf{H}$ also is a block-lower-triangular matrix with diagonal block being $\sigma_k(t)L_tC^+B(t)$, $t = 0,1,\cdots,N-1$. If there is no link failure, i.e., $\sigma_k(t) = 1$, then $\rho(I - \sigma_k(t)L_tC^+B(t)) < 1$, while if there is a link failure, i.e., $\sigma_k(t) = 0$, then $\rho(I - \sigma_k(t)L_tC^+B(t)) = 1$. Thus it is obvious that

$$\rho(I - \mathbf{L}\Sigma_k\mathbf{H}) \le 1. \tag{15}$$

Taking norms on both sides of (14), we have

$$\|\delta U_{k+1}\| \le \|I - \mathbf{L}\Sigma_k\mathbf{H}\| \cdot \|\delta U_k\| \le \|\delta U_k\|.$$

That is, δU_k or U_k is bounded for all iterations. More specifically, $\|\delta U_k\|$ is non-increasing.

Now come to (14), and rewrite it as

$$\delta U_{k+1} = (I - \mathbf{L}\bar{\Sigma}\mathbf{H})\delta U_k + \mathbf{L}(\bar{\Sigma} - \Sigma_k)\mathbf{H}\delta U_k,$$

where $\mathbb{E}$ is mathematical expectation and $\bar{\Sigma} = \mathbb{E}\Sigma_k$. Thus $\bar{\Sigma} - \Sigma_k$ is with zero mean and finite second moment. Recursively, we have

$$\begin{aligned} \delta U_{k+1} =&(I - \mathbf{L}\bar{\Sigma}\mathbf{H})^k \delta U_1 \\ &+ \sum_{j=1}^{k} (I - \mathbf{L}\bar{\Sigma}\mathbf{H})^{k-j}\mathbf{L}(\bar{\Sigma} - \Sigma_j)\mathbf{H}\delta U_j. \end{aligned}$$

When the condition $\rho(I - L_tC^+B(t)) < 1$ is satisfied, it is easy to find that $\rho(I - L_t\bar{\sigma}(t)C^+B(t)) < 1$ and thereby $\rho(I - \mathbf{L}\bar{\Sigma}\mathbf{H}) < 1$. Consequently, the first term on the right side of last equation tends to zero as k goes to infinity. Note that U_k is generated by signal at the $(k-1)$th iteration, thus it is independent of Σ_k. As a result

$$\begin{aligned} &\sum_{j=1}^{k} \mathbb{E}\|(I - \mathbf{L}\bar{\Sigma}\mathbf{H})^{k-j}\mathbf{L}(\bar{\Sigma} - \Sigma_j)\mathbf{H}\delta U_j\|^2 \\ &\le \sum_{j=1}^{k} \|(I - \mathbf{L}\bar{\Sigma}\mathbf{H})^{k-j}\|^2 \|\mathbf{L}\|^2 \mathbb{E}\|\bar{\Sigma} - \Sigma_j\|^2 \|\mathbf{H}\|^2 \mathbb{E}\|\delta U_j\|^2 \end{aligned}$$

$$\mathbf{H}=\begin{bmatrix} C^{+}B(0) & 0 & \cdots & 0 \\ C(2)A(1)B(0) & C^{+}B(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C(N)(\prod_{i=1}^{N-1}A(i))B(0) & C(N)(\prod_{i=2}^{N-1}A(i))B(1) & \cdots & C^{+}B(N-1) \end{bmatrix} \quad (7)$$

$$\leq \eta \sum_{j=1}^{k} \|(I-\mathbf{L}\overline{\Sigma}\mathbf{H})^{k-j}\|^2 < \infty,$$

where $\eta \triangleq \sup_j \|\mathbf{L}\|^2\mathbb{E}\|\overline{\Sigma}-\Sigma_j\|^2\|\mathbf{H}\|^2\mathbb{E}\|\delta U_j\|^2$. By probability theory, one has

$$\sum_{j=1}^{k}(I-\mathbf{L}\overline{\Sigma}\mathbf{H})^{k-j}\mathbf{L}(\overline{\Sigma}-\Sigma_j)\mathbf{H}\delta U_j < \infty.$$

In other words, δU_k converges almost surely. This completes the proof. □

As one could see, the almost sure convergence of U_k has been obtained. However, it is not straightforward whether $\delta U_k \xrightarrow[k\to\infty]{} 0$ hold. To make more freedom of the design on learning gain matrix L_t and to achieve zero convergence of δU_k, we could make a small modification to (12) by adding a decreasing step-size a_k. Then, the update law would be

$$U_{k+1} = U_k + a_k\mathbf{L}\Sigma_k E_k, \quad (16)$$

where a_k satisfies $a_k \xrightarrow[k\to\infty]{} 0$, $\sum_{k=1}^{\infty} a_k = \infty$, $\sum_{k=1}^{\infty} a_k^2 < \infty$. It is easy to see that $a_k = 1/k$ meets all these requirements.

Theorem 2: Consider system (8) and update law (16). If the coupling matrix $-L_tC^{+}B(t)$ is stable for all t, then the input U_k would converge to U_d almost surely. By stability of a matrix we mean that all its eigenvalues are with negative real parts.

Proof: Following similar steps of the proof for Theorem 1, we get the boundedness of δU_k and

$$\delta U_{k+1} = (I-a_k\mathbf{L}\overline{\Sigma}\mathbf{H})\delta U_k + a_k\mathbf{L}(\overline{\Sigma}-\Sigma_k)\mathbf{H}\delta U_k,$$

or

$$\delta U_{k+1} = \Phi_{k,1}\delta U_1 + \sum_{j=1}^{k}\Phi_{k,j+1}a_j\mathbf{L}(\overline{\Sigma}-\Sigma_j)\mathbf{H}\delta U_j, \quad (17)$$

where $\Phi_{k,j} \triangleq (I-a_k\mathbf{L}\overline{\Sigma}\mathbf{H})\times\cdots\times(I-a_j\mathbf{L}\overline{\Sigma}\mathbf{H})$, $j\leq k$ and $\Phi_{k,k+1}=I$.

We first give an estimate of $\Phi_{k,j}$. To be specific, let us show that there exist constants $c_0>0$ and $c>0$ such that

$$\|\Phi_{k,j}\| \leq c_0 \exp\left(-c\sum_{i=j}^{k} a_i\right). \quad (18)$$

Note that both $\mathbf{L}$ and $\overline{\Sigma}$ are block-diagonal matrix, thus they are commutative, whence $\mathbf{L}\overline{\Sigma}\mathbf{H} = \overline{\Sigma}\mathbf{L}\mathbf{H}$. $\overline{\Sigma}\mathbf{L}\mathbf{H}$ is a block lower triangular matrix with diagonal blocks being $\overline{\sigma}(t)L_tC^{+}B(t)$. If all eigenvalues of $L_tC^{+}B(t)$ has positive real parts for all t by the convergence condition, then $-\mathbf{L}\overline{\Sigma}\mathbf{H}$ is stable. This further implies that there is a positive definite matrix P such that

$$P(-\mathbf{L}\overline{\Sigma}\mathbf{H}) + (-\mathbf{L}\overline{\Sigma}\mathbf{H})^TP \leq -I. \quad (19)$$

Denote $\Lambda \triangleq -\mathbf{L}\overline{\Sigma}\mathbf{H}$. Then,

$$\begin{aligned}&\Phi_{k,j}^TP\Phi_{k,j}\\ &=\Phi_{k-1,j}^T(I+a_k\Lambda)^TP(I+a_k\Lambda)\Phi_{k-1,j}\\ &\leq\Phi_{k-1,j}^T(P+a_k^2\Lambda^TP\Lambda+a_k\Lambda^TP+a_kP\Lambda)\Phi_{k-1,j}\\ &\leq\Phi_{k-1,j}^T(P+a_k^2\Lambda^TP\Lambda-a_kI)\Phi_{k-1,j}\\ &\leq\Phi_{k-1,j}^TP^{\frac{1}{2}}(I-a_kP^{-1}+a_k^2P^{-\frac{1}{2}}\Lambda^TP\Lambda P^{-\frac{1}{2}})P^{\frac{1}{2}}\Phi_{k-1,j},\end{aligned}$$

where for large enough k, say $k\geq k_0$,

$$\|I-a_kP^{-1}+a_k^2P^{-\frac{1}{2}}\Lambda^TP\Lambda P^{-\frac{1}{2}}\| \leq 1-2ca_k < e^{-2ca_k}$$

with suitable $c>0$. It leads to

$$\Phi_{k,j}^TP\Phi_{k,j} \leq \left(\exp\left(-2c\sum_{i=j}^{k}a_i\right)\right)I, \quad (20)$$

and hence

$$\|\Phi_{k,j}\| \leq \lambda_{\min}^{-\frac{1}{2}}\exp\left(-c\sum_{i=j}^{k}a_i\right) \quad (21)$$

which verifies (18).

Now come back to (17). It is obvious that the first term converges to zero by (21) as k goes to infinity. Thus it is sufficient to prove zero-convergence of the last term of (17). Following similar steps of the proof for Theorem 1, one has

$$\sum_{j=1}^{\infty}a_j\mathbf{L}(\overline{\Sigma}-\Sigma_j)\mathbf{H}\delta U_j < \infty.$$

Then using similar steps of Lemma 3.1.1 of [16], one could obtain that $\sum_{j=1}^{k}\Phi_{k,j+1}a_j\mathbf{L}(\overline{\Sigma}-\Sigma_j)\mathbf{H}\delta U_j \xrightarrow[k\to\infty]{} 0$. This completes the proof. □

Remark 1: Comparing Theorems 1 and 2, we find that for the former theorem the convergence condition is $\rho(I - L_t C^+ B(t)) < 1$, while for the latter it requires $-L_t C^+ B(t)$ to be stable. Thus it is obvious that the condition for the latter case is much more relaxed than the former one and provides more freedom for design of L_t. However, it is worth mentioning that the sacrifice for freedom of L_t is convergence speed of the proposed algorithm. That is, update law (16) would converge slower than (12). To be specific, from (14) one can find that the convergence speed is exponential, where the exponential parameter depends on the rate of random failures. On the other hand, the modified algorithm (16) is a stochastic approximation algorithm, whence the convergence speed is not faster than the descending speed of the parameter a_k [16]. Thus there is a trade off between the selection of learning gain matrix and convergence speed.

Remark 2: As one could see, the convergence condition is placed on $L_t C^+ B(t)$ and is irrelevant with link failure rate $\overline{\sigma}(t)$. This implies that the conventional P-type algorithm could guarantee almost sure convergence as long as the link from plant to controller is not completely broken.

Remark 3: The selection of learning matrix L_t should satisfy the conditions given in Theorem 1 or 2 for different algorithms. As a matter of fact, it can be derived by solving LMIs $-I < I - L_t C^+ B(t) < I$ or $L_t C^+ B(t) > 0$, respectively. A special case of L_t is the transpose of $C^+ B(t)$ multiplied with suitable coefficient, i.e., $L_t = \alpha(C^+ B(t))^T$ where α is a positive constant. For the former case, α should be small to meet the condition of Theorem 1, while for the latter case, an arbitrary positive α is sufficient to derive Theorem 2.

Remark 4: It is noticed that the analysis is given on the basis of lifted models, whence the dimension of the matrices are quite large due to the long iteration length of practical applications. However, the lifted form is only used to simplify the derivations and would not increase the computation burden. As a matter of fact, the convergence condition and update laws are given to each time instance in an iteration. From this point of view, the computation burden for the proposed algorithm is not tremendous.

4. GENERAL LINK FAILURES CASE

In this section, we will consider the case that link failures occur both from plant to controller and from controller to plant in detail. It should be specially pointed out that this is not a trivial extension of last section. When only link failure at the measurement side is taken into account, the system would successively improve its tracking performance. This could be seen from (4), where the input would hold the value of previous iterations if a link failure happens. In other words, if $\sigma_k(t) = 0$, then $u_{k+1}(t) = u_k(t)$. However, when the link failure from controller to plant is involved, the tracking performance would be complicated as there is a possibility that the system lacks control signal, which definitely make corresponding output deviate from the desired reference. Recalling system (3), if a link failure happens, i.e., $\gamma_k(t) = 0$, the system becomes autonomous, $x_k(t+1) = A(t)x_k(t)$. Thus more details are to be discussed.

Similar to last section, we first make a lifted model. To this end, denote

$$\Gamma_k = \mathrm{diag}\{\gamma_k(0)I_{p\times p}, \gamma_k(1)I_{p\times p}, \cdots, \gamma_k(N-1)I_{p\times p}\}. \quad (22)$$

Then we have

$$Y_k = \mathbf{H}\Gamma_k U_k + Y_k^0, \quad (23)$$

and then the tracking error E_k used for (12) is given as

$$E_k = \mathbf{H}U_d - \mathbf{H}\Gamma_k U_k = \mathbf{H}(U_d - \Gamma_k U_k). \quad (24)$$

Due to the existence of Γ_k, it is no longer able to get $Y_k \xrightarrow[k\to\infty]{} Y_d$. As a link failure happens, the system becomes autonomous and the corresponding output would deviate from the reference, which further results in a disturbance to the input for the next iteration. Instead of making the expectation of tracking error to be zero, i.e., $\mathbb{E}E_k = 0$, now the control objective is to minimize the following tracking performance index,

$$V_t = \mathbb{E}(\|Y_d - Y_k\|^2 | U_k). \quad (25)$$

By simple calculations,

$$\begin{aligned} V_t =& \mathbb{E}(\|Y_d - Y_k\|^2 | U_k) \\ =& \mathbb{E}(\|\mathbf{H}(U_d - \Gamma_k U_k)\|^2 | U_k) \\ =& \mathbb{E}(\|\mathbf{H}(U_d - \overline{\Gamma}U_k + \overline{\Gamma}U_k - \Gamma_k U_k)\|^2 | U_k) \\ =& \|\mathbf{H}(U_d - \overline{\Gamma}U_k)\|^2 + \mathbb{E}\|\overline{\Gamma} - \Gamma_k\|^2 \cdot \|U_k\|^2. \end{aligned}$$

Thus our objective is to show $U_d - \overline{\Gamma}U_k$ converges to zero almost surely.

Using update law (12), we have the following theorem. However, the boundedness of this case is not straightforward as the one given in the proof of Theorem 1. Thus we will assume that the iteration of (12) keeps bounded for expression concise. Some comments are given later in following remarks.

Theorem 3: Consider system (23) and update law (12). If the following condition is satisfied

$$\rho(I - L_t C^+ B(t)) < 1 \quad (26)$$

for all t and $\sup_k \|U_k\| < \infty$, then the input U_k and E_k converges almost surely.

Proof: From (12) we have $U_{k+1} = U_k + \mathbf{L}\Sigma_k \mathbf{H}(U_d - \Gamma_k U_k) = U_k + \mathbf{L}\Sigma_k \mathbf{H}(U_d - \overline{\Gamma}U_k + \overline{\Gamma}U_k - \Gamma_k U_k)$. Multiplying $\overline{\Gamma}$ from left and subtracting from U_d leads to

$$\Delta U_{k+1} = (I - \overline{\Gamma}\mathbf{L}\Sigma_k \mathbf{H})\Delta U_k - \overline{\Gamma}\mathbf{L}\Sigma_k \mathbf{H}(\overline{\Gamma} - \Gamma_k)U_k$$

$$=(I-\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})\Delta U_k+\overline{\Gamma}\mathbf{L}(\overline{\Sigma}-\Sigma_k)\mathbf{H}\Delta U_k-\overline{\Gamma}\mathbf{L}\Sigma_k\mathbf{H}(\overline{\Gamma}-\Gamma_k)U_k, \quad (27)$$

where $\Delta U_k=U_d-\overline{\Gamma}_kU_k$. This further leads to

$$\begin{aligned}\Delta U_{k+1}=&(I-\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})^k\Delta U_1\\&+\sum_{j=1}^{k}(I-\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})^{k-j}\overline{\Gamma}\mathbf{L}(\overline{\Sigma}-\Sigma_j)\mathbf{H}\Delta U_j\\&-\sum_{j=1}^{k}(I-\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})^{k-j}\overline{\Gamma}\mathbf{L}\Sigma_j\mathbf{H}(\overline{\Gamma}-\Gamma_j)U_j,\end{aligned}$$

where the first term converges to zero as k goes to infinity as long as $\rho(I-\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})<1$, which is true because $\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H}$ is a block lower triangular matrix with diagonal block being $\overline{\sigma}\overline{\gamma}L_tC^+B(t)$. While for the latter terms, it is noticed both $\overline{\Sigma}-\Sigma_k$ and $\overline{\Gamma}-\Gamma_k$ are a random matrices with zero-mean and bounded second moment. Besides, $\sigma_k(t)$ is independent of $\gamma_k(t)$, $\forall t$ and they are independent of U_k meanwhile. Thus using similar techniques in the proof for Theorem 1, it follows that

$$\sum_{j=1}^{k}(I-\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})^{k-j}\overline{\Gamma}\mathbf{L}(\overline{\Sigma}-\Sigma_j)\mathbf{H}\Delta U_j<\infty, \quad (28)$$

$$\sum_{j=1}^{k}(I-\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})^{k-j}\overline{\Gamma}\mathbf{L}\Sigma_j\mathbf{H}(\overline{\Gamma}-\Gamma_j)U_j<\infty. \quad (29)$$

This shows the convergence of ΔU_k, and the convergence of U_k and E_k is obvious.

Now let us draw some derivations for E_k. Since $E_k=\mathbf{H}U_d-\mathbf{H}\Gamma_kU_k$, it follows

$$\begin{aligned}E_{k+1}-E_k=&\mathbf{H}\Gamma_kU_k-\mathbf{H}\Gamma_{k+1}U_{k+1}\\=&\mathbf{H}\Gamma_kU_k-\mathbf{H}\Gamma_{k+1}(U_k+\mathbf{L}\Sigma_kE_k)\\=&\mathbf{H}(\Gamma_k-\Gamma_{k+1})U_k-\mathbf{H}\Gamma_{k+1}\mathbf{L}\Sigma_kE_k,\end{aligned}$$

whence

$$\begin{aligned}E_{k+1}=&(I-\mathbf{H}\Gamma_{k+1}\mathbf{L}\Sigma_k)E_k+\mathbf{H}(\Gamma_k-\Gamma_{k+1})U_k\\=&(I-\mathbf{H}\overline{\Gamma}\mathbf{L}\overline{\Sigma})E_k+\mathbf{H}(\Gamma_k-\Gamma_{k+1})U_k\\&+\mathbf{H}(\overline{\Gamma}\mathbf{L}\overline{\Sigma}-\Gamma_{k+1}\mathbf{L}\Sigma_k)E_k.\end{aligned}$$

Thus following similar techniques, the convergence of E_k is also obtained. This completes the proof. □

Similarly, we would like to further derive a zero convergence and make more freedom of the design of L_t. Now considering the update law (16), we have the following results.

Theorem 4: Consider system (23) and update law (16). If the coupling matrix $-L_tC^+B(t)$ is stable for all t and $\sup_k\|U_k\|<\infty$, then the input error $U_d-\overline{\Gamma}U_k$ converges to zero almost surely.

Proof: Similar to (27) the following recursion is observed

$$\begin{aligned}\Delta U_{k+1}=&(I-a_k\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})\Delta U_k+a_k\overline{\Gamma}\mathbf{L}(\overline{\Sigma}-\Sigma_k)\mathbf{H}\Delta U_k\\&-a_k\overline{\Gamma}\mathbf{L}\Sigma_k\mathbf{H}(\overline{\Gamma}-\Gamma_k)U_k,\end{aligned} \quad (30)$$

and then

$$\begin{aligned}\Delta U_{k+1}=&\Psi_{k,1}\Delta U_1+\sum_{j=1}^{k}\Psi_{k,j+1}a_j\overline{\Gamma}\mathbf{L}(\overline{\Sigma}-\Sigma_j)\mathbf{H}\Delta U_j\\&-\sum_{j=1}^{k}a_j\overline{\Gamma}\mathbf{L}\Sigma_j\mathbf{H}(\overline{\Gamma}-\Gamma_j)U_j,\end{aligned} \quad (31)$$

where $\Psi_{k,j}\triangleq(I-a_k\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})\cdots(I-a_j\overline{\Gamma}\mathbf{L}\overline{\Sigma}\mathbf{H})$, $\forall j\le k$ and $\Psi_{k,k+1}=I$. By using similar analysis techniques used for estimation of $\Phi_{k,j}$ in the proof of Theorem 2, there exist constants $c_0>0$ and $c>0$ such that

$$\|\Psi_{k,j}\|\le c_0\exp\left(-c\sum_{i=j}^{k}a_i\right). \quad (32)$$

The rest of the proof is completely same to Theorem 2 and thus is omitted for saving space. □

Remark 5: Unlike those in Theorrems 1 and 2, an additional boundedness requirement of iterations is added to the convergence conditions in Theorems 3 and 4. This is a technique requirement for convergence analysis. In practical applications, we could use projection algorithm to guarantee this condition. For example, instead of (4), one might use $u_{k+1}(t)=\Xi(u_k(t)+L_te_k(t+1))$ where $\Xi(\cdot)$ is a projection to a prescribed finite space G. That is, Ξ is the identity map for points in the interior of G, and maps a point outside G to the point in G closest to it w.r.t. Euclidean distance.

Remark 6: As have been explained at the beginning of this section, the requirement on boundedness originates from the loss of input signal, which further leads to large derivation in tracking errors. An intuitive solution to this problem is the so-called hold-strategy for input signal. That is, if the input signal is lost due to link failure, the system could detect this fact and then use the one of previous iteration to ensure that an input is put into the system. For example, assume $u_k(t)$ is lost, then the system will use $u_{k-1}(t)$ instead to drive itself. While if $u_{k-1}(t)$ is also lost, the system would track back to the last available one.

5. ILLUSTRATIVE SIMULATIONS

Let us consider the following linear discrete-time system

$$x_k(t+1)=\begin{bmatrix}-0.8+0.02\sin t & -0.22\\ 1 & 0\end{bmatrix}x_k(t)+\begin{bmatrix}0.5 & 0.05\\ 1-0.05\cos(t/10) & 0.5\end{bmatrix}u_k(t), \quad (33)$$

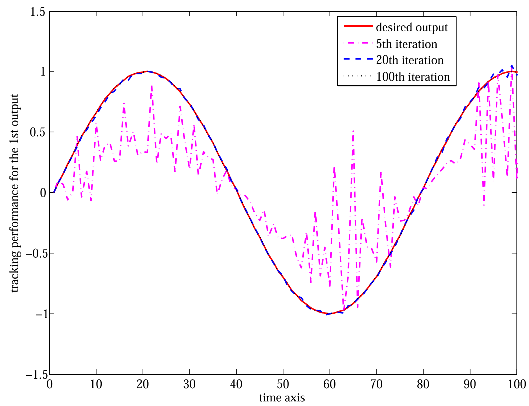

Fig. 2. Tracking performance for the 1st output: update law (12).

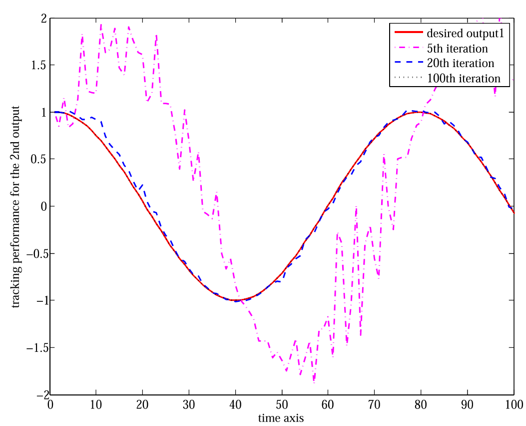

Fig. 3. Tracking performance for the 2nd output: update law (12).

$$y_k(t) = \begin{bmatrix} 1 & 0.05t^2 \\ 0 & 1 \end{bmatrix} x_k(t),$$

where $N = 100$, i.e., $t \in [0, 100]$. The state, input, and output are all of two-dimension.

The tracking reference is $y_d(t) = [y_d^{(1)}(t)\ y_d^{(2)}(t)]^T$, where $y_d^{(1)}(t) = \sin(4t/50)$ and $y_d^{(2)}(t) = \cos(4t/50)$. For each case, the algorithm is run for 100 iterations.

We first consider the case that the link failures only happen at the measurement side. Here the parameter of link failures is set as follows. As has been stated in the formulation section, the expectation of the random variable can vary at different time instances. Therefore, we set $\overline{\sigma}(t) = 1 - 0.3\beta_t$, where β_t is randomly generated following uniform distribution in $[0, 1]$. Then, $\mathbb{P}\{\sigma_k(t) = 0\} = 0.3\beta_t, \forall t$. Generally speaking, about 15% of the measurement data is lost due to link failures.

Consider the algorithm (12), where the learning gain is selected as $L_t = 0.8I$. It is easy to verify that the condition given in Theorem 1 is satisfied. The output tracking performance are shown in Figs. 2 and 3 for the 1st and 2nd outputs, respectively. As one could see, the outputs track the desired references effectively after several iterations.

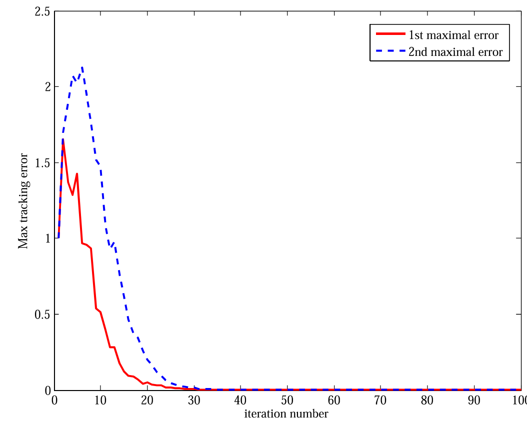

Fig. 4. Maximal tracking error along iterations: update law (12).

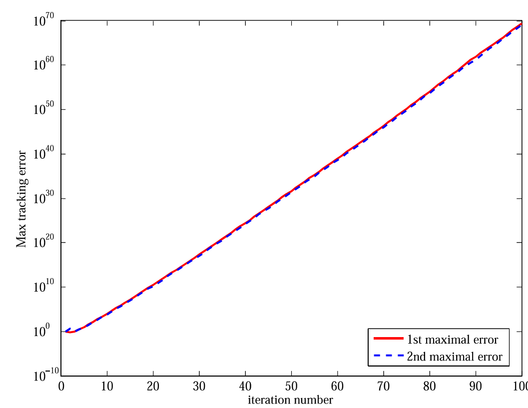

Fig. 5. Maximal tracking error along iterations: update law (12) with $L_t = 3I$.

The maximal tracking errors, $\max_t e_k(t)$, for the 1st and 2nd outputs along iteration axis are displayed in Fig. 4, where the error is quite near zero after 30 iterations. This shows the effectiveness of the algorithm (12).

To see the conservative selection on learning gain matrix L_t, we give another selection $L_t = 3I$. It is easy to verify that $-L_t C^+ B(t)$ is stable but the eigenvalues of $I - L_t C^+ B(t)$ do not lie in the unit circle. We simulate algorithm (12) again and the maximal tracking error along iteration axis is shown Fig. 5, where one could see that the tracking error diverges to infinity.

However, when algorithm (16) is used with $L_t = 3I$ and $a_k = 4/(k+10)$, a perfect tracking performance could be still obtained. As displayed in Figs. 6 and 7 for the first and second output, respectively, the actual output could achieve a satisfactory tracking after several iterations. Particularly, the outputs at 100 iteration almost coincide with the desired references.

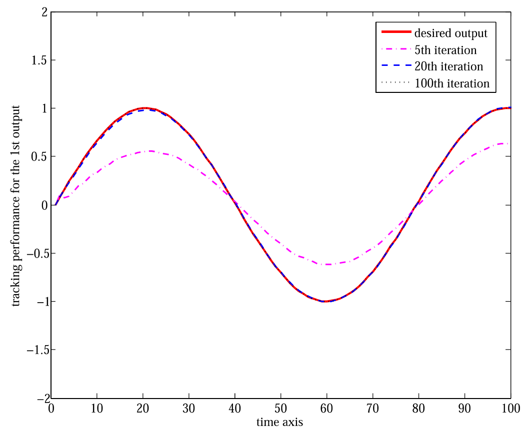

Fig. 6. Tracking performance for the 1st output: update law (16).

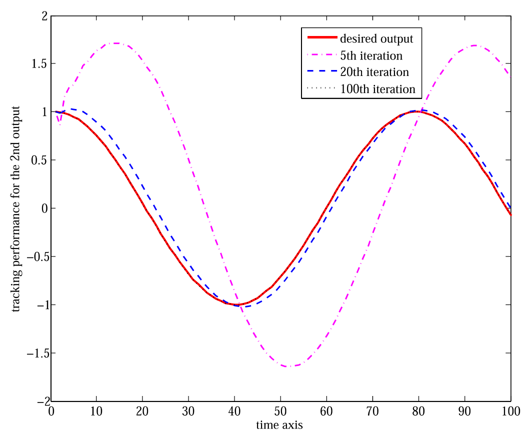

Fig. 7. Tracking performance for the 2nd output: update law (16).

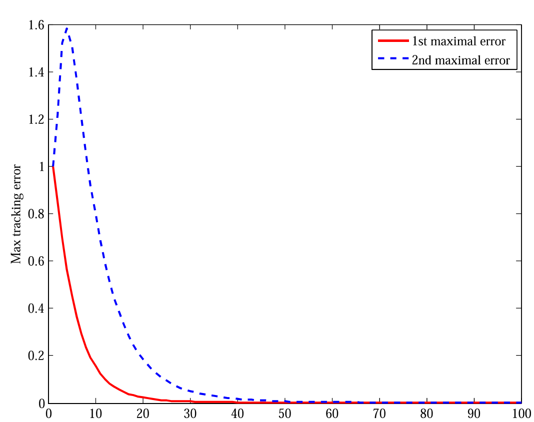

Fig. 8. Maximal tracking error along iterations: update law (16).

The maximal tracking errors, $\max_t e_k(t)$, for the 1st and 2nd outputs along iteration axis are displayed in Fig. 8. It can be seen from this figure that the convergence speed

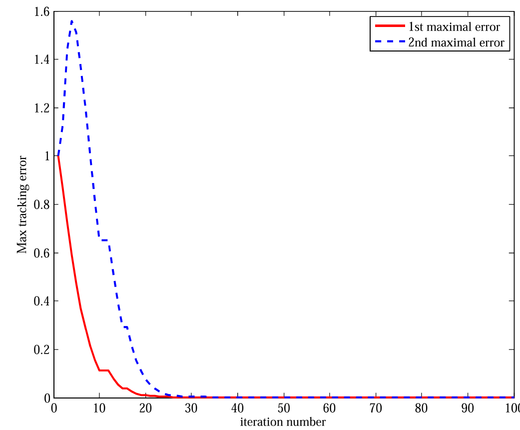

Fig. 9. Maximal tracking error for general case: update law (12).

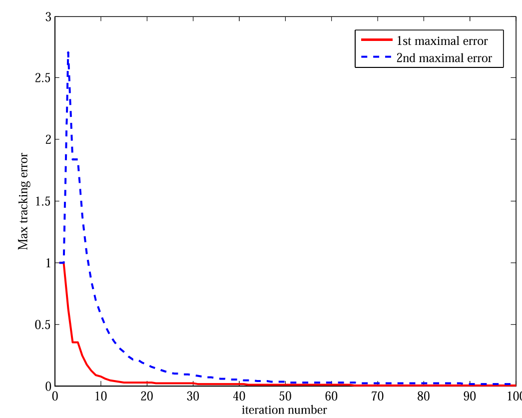

Fig. 10. Maximal tracking error for general case: update law (16).

might be slow due to the decreasing design technique on a_k. However, this technique leaves much freedom on the selection of learning gain matrix L_t for us. The latter is much more important from the view point of practical applications. In addition, a possible way to solve the slow convergence speed is to set a constant learning gain for the first several iterations.

It is worth pointing out that the algorithms run for each time instance rather than for the whole iteration together. Thus, the computation burden for each step is little, although a hundred of time instances are taken into account for an iteration.

For the general case, we use a simple model of the link failure at both sides. We simply set that $\overline{\gamma}(t) = \overline{\sigma}(t) = 0.9$. In other words, about 10% data at each side would be lost during the transmission. The maximal tracking errors, $\max_t e_k(t)$, for (12) and (16) are shown in Figs. 9 and 10, respectively. Similar behavior can be observed for the general case.

6. CONCLUSIONS

Two P-type ILC update laws have been designed in this paper for networked linear system with random link failures, which appear both from the controller to the plant and from the plant to the controller. It is the latter case that is usually considered in previous studies, while the former case is found to be essentially different from the latter case. Here link failures are modeled by independent Bernoulli random variables. The almost sure convergence properties of the proposed algorithms are critically proved. For further research, the case of networked nonlinear systems with random link failures is of great interest, which definitely requires much more efforts since it is more complex than the linear case considered in this paper.

REFERENCES

[1] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control: A learning-based method for high-performance tracking control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96-114, 2006.

[2] H. S. Ahn, Y. Q. Chen, and K.L. Moore, "Iterative learning control: survey and categorization from 1998 to 2004," *IEEE Trans. Systems, Man, and Cybernetics-Part C*, vol. 37, no. 6, pp. 1099-1121, 2007.

[3] Q.-Y. Xu, X.-D. Li, and M.-M. Lv, "Adaptive ILC for tracking non-repetitive reference trajectory of 2-D FMM under random boundary condition," *International Journal of Control, Automation, and Systems*, vol. 14, no. 2, pp. 478-485, 2016. [click]

[4] D. Shen, Y. Wang, "Survey on stochastic iterative learning control," *Journal of Process Control*, vol. 24, no. 12, pp. 64-77, 2014. [click]

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

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

[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," *Proc. the IEEE Int. Conf. Control Automation, Robotics, and Vision*, pp. 757-762, 2008.

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

[9] X. Bu, F. Yu, Z.-S. Hou, and F. Wang, "Iterative learning control for a c1ass of nonlinear systems with measurement dropouts," *Control Theory and Applications* (In Chinese), vol. 29, no. 11, pp. 1458-1464, 2012.

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

[11] D. Shen and Y. Wang, "Iterative learning control for networked stochastic systems with random data dropouts," *International Journal of Control*, vol. 88, no. 5, pp. 959-968, 2015. [click]

[12] D. Shen and Y. Wang, "ILC for networked nonlinear systems with random output dropouts and unknown control direction," *System & Control Letters*, vol. 77, pp. 30-39, 2015. [click]

[13] K. Li and X. Wang, "Cross-layer optimization for LDPC-coded multirate multiuser systems with QoS constraints," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2567-2578, 2006.

[14] H. Li, X. Liao, T. Huang, W. Zhu, and Y. Liu, "Second-order global consensus in multiagent networks with random directional link failure," *IEEE Transactions on Neural Networks And Learning Systems*, vol. 26, no. 3, pp. 565-575, 2015.

[15] H. Zhang, J. Wang, Z. Wang, and H. Liang, "Mode-dependent stochastic synchronization for markovian coupled neural networks with time-varying mode-delays," *IEEE Transactions on Neural Networks And Learning Systems*, vol. 26, no. 11, pp. 2621-2634, 2015.

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

Dong Shen received his B.S. degree in mathematics from Shandong University, Jinan, China, in 2005. He received his 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 (BUCT), Beijing, China. His current research interests include iterative learning control, stochastic control and optimization. He has published more than 30 refereed journal and conference papers. Dr. Shen received IEEE CSS Beijing Chapter Young Author Prize in 2014 and Wentsun Wu Artificial Intelligence Science and Technology Progress Award in 2012.