Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

A comparison of compensation methods for random input data dropouts in networked iterative learning control system

Lixun Huang[1*], Hanqing Ding[1], Zhe Zhang[1], Qiuwen Zhang[1] and Lijun Sun[2]

*Correspondence: shuhlx@163.com
[1]School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, P.R. China
Full list of author information is available at the end of the article

Abstract

In this paper, for a class of linear networked iterative learning control (ILC) systems, methods to compensate dropped input data in time or iteration domain are compared. Specifically, the transition matrices of input error at the controller side with the two methods are derived first, respectively. After that, the varieties of eigenvalues and elements in the lower triangular of the transition matrices are analyzed. Through analyzing the varieties, it can be easily found that the two methods guarantee the convergence of input error at the controller side, while only the compensation in iteration domain guarantees the convergence of input error at the actuator side. Due to the introduction of networks, the convergence of output error is determined by the input error at the actuator side. Hence, a conclusion could be made naturally that the output error converges to zero with compensation in iteration domain, while compensation in time domain cannot guarantee that. Finally, numerical experiments are given to corroborate the theoretical analysis.

Keywords: Iterative learning control; Networked control systems; Data dropout

1 Introduction

Iterative learning control (ILC) is an effective method when it is used to deal with the system that operates repetitively in a finite time interval [1–4]. The key feature of this method is to use the output error obtained from the last and/or current iteration to enable the output to converge to a desired trajectory. Convergence has been studied in ILC from a number of different perspectives such as stochastic noise [5–7], initial input error [8, 9], model uncertainty [10, 11], disturbance rejection [12], and parameter optimization [13].

In recent times, the system controlled over networks has attracted more and more devotion of researchers [14, 15]. Contrary to classical control systems, networked control systems (NCSs) are closed via wired or wireless networks which transmit output data from sensors to controllers and input data from controllers to actuators. Due to the introduction of various networks, NCSs have advantages such as easy setup and maintenance, reduced weight and wiring. These distinct features of NCSs make them particularly suitable for several emerging applications including remote robot etc.

However, since the output and input data of NCSs need to be transmitted via networks, the analysis and design of such systems render some new challenges due to complex interconnections between sensor, controller, and actuator, such as limited bandwidth, quantization error, input saturations, data dropouts, and so on. Thus, any control technique using undistorted information is inapplicable to NCSs.

As for networked ILC systems, a recent paper summarized the progress on ILC in the presence of data dropouts from three aspects including data dropout model, data dropout position, and convergence meaning [16]. When reviewing the contribution of existing papers from the perspective of data dropout position, authors pointed out it can be easily seen that the lost input data cannot be simply replaced by 0 because it would greatly damage the tracking performance, which means the lost input data must be compensated with suitable data to maintain the operation process of the plant. Authors further indicated there are two kinds of methods to solve the problem of lost input data: Kalman filtering and data compensation.

Based on the Kalman filtering approach, Ahn et al. considered a linear discrete-time ILC system with random packet dropouts in the output and/or input data [17–20]. When the output data is subject to data dropout, in [17], authors presented a mathematical formulation problem of robust ILC design. Furthermore, a method was designed to select learning gain optimally based on Kalman filtering such that the system eventually converges to a desired trajectory as long as the output data is not dropped completely. In [18], they further considered that each component in the output data vector was dropped independently. After [17] and [18], convergence conditions for ILC systems were established in the presence of data dropouts and delays in both input and output data simultaneously in [19]. In [20], the author continued to consider the design of an optimal learning gain matrix for the more realistic case of random data dropout.

As to data compensation, this approach can be further divided into two categories: time domain compensation and iteration domain compensation. The former is similar to the typical method to handle data dropout in general NCSs, which used input data at time instant $t-1$ to compensate the lost one at time instant t during kth iteration. Pan earlier considered a class of ILC systems via networks with communication delay and data dropouts first [21]. In the discussion, an event-driven model was used, which implies the dropped data is compensated in time domain. Through the analysis of a variety of element values in the transition matrix of input vector, convergence of the networked ILC systems with data dropout compensation in time domain was improved. Moreover, this method was then adopted by [22] to consider the problem of ILC over networks for a class of nonlinear systems with random packet dropouts in inputs and outputs simultaneously. This paper proved that under some given conditions, the ILC can guarantee the convergence of the tracking error although some packets are missing. The latter applies the data at the $(k-1)$th iteration to compensate the dropped one at the kth iteration at the same time instant t, which was first used by [23]. In this paper, a discrete-time linear ILC system with random input data dropouts was considered. The convergence property of output error was proved by analyzing the element values of system transition matrix. Similar compensation methods were used to deal with the effect of delay or data dropout in input and/or output data of different networked ILC systems in [24, 25].

A comment should be made that general data dropout environments were considered in the last two years. A derivative-type networked ILC scheme was proposed in [26] for a class of repetitive discrete-time single-input-single-output systems with data dropouts stochastically occurring in input and output communication channels. The compensation methods mended the dropped instant-wise output with the synchronous desired output and drove the plant by refreshing the dropped instant-wise input with the used consensus-instant input at the previous iteration. In [27], the author studied the ILC for a class of nonlinear systems with random data dropouts occurring independently at both measurement and actuator sides. Updated algorithms were proposed for the input data at both the controller side and the actuator side. In [28], the similar compensation method was used for a class of linear ILC systems with general data dropouts at both measurement and actuator sides. In this paper, the sample path behavior along the iteration axis was formulated as a Markov chain. Based on the Markov chain, the recursion of the input error was reformulated as a switching system. The author also used the Markov chain to analyze the convergence of linear ILC stochastic systems under general data dropout environments in [29], and a new analysis method was developed to prove the convergence in both mean square and almost sure senses. A data-driven learning control method for stochastic nonlinear systems under random communication conditions was proposed in [30], including data dropouts, communication delays, and packet transmission disordering. Specifically, the data arriving in the buffer was regulated by a renewal mechanism, and a suitable update data was selected to the controller by a recognition mechanism.

Based on these aforementioned contributions, some results can be easily summarized and listed as follows:

- Most works usually consider output data dropout occurring at sensor-to-controller side, while input data at the controller-to-actuator side is assumed to work well. What is more, the results to deal with data dropout in output cannot be extended to the case that input suffered random data dropouts;
- Under some given conditions, the compensation methods in time or iteration domain improve the convergence of networked ILC systems with random input and/or output data dropout, while the work comparing the difference theoretically between the methods in time or iteration domain to guarantee the convergence of networked ILC systems with random data dropout is rarely seen.

Inspired by these observations, as a further study along this track, we continue to address the convergence performance of networked ILC systems in presence of random input data dropout and to compare the difference in convergence of the networked ILC systems with lost data compensated in different domains. Specifically, our contributions lie in the following aspects:

- We consider a class of linear time-invariant systems controlled by a P-style ILC algorithm over networks and establish a corresponding system model with data dropouts taken into account;
- On the base of a single input data dropout, methods to compensate the dropped input data in time or iteration domain are compared. The comparison is done through analyzing the varieties of eigenvalues and elements in the lower triangular of the transition matrices of input error at the controller side. Based on the analysis, a conclusion is derived that the output error converges to zero with the dropped input data compensated in iteration domain, while compensation in time domain cannot guarantee that.

The remainder of this paper is organized as follows. In the next section, the ILC system with random data dropout taken into account is formulated. After that, the transition matrices of input error at the controller side are derived with compensation in time or iteration domain, respectively. Through analyzing the varieties of eigenvalues and elements in the lower triangular of the transition matrix, some useful results are derived to reveal the difference in convergence of the system with the involved two compensation methods. In Sect. 4, some numerical examples are given to illustrate the correctness of the results derived theoretically in Sect. 3. Finally, some conclusions wrap up this paper in Sect. 5.

2 Problem formulation

Consider the discrete-time, linear, and time-invariant system described as follows:

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

where $k = 0, 1, \ldots$ is an iterative trial number, $t \in [0, 1, \ldots, T]$ denotes discrete time for the periodic trial of the system, $x_k(t) \in R^n$, $u_k(t) \in R^m$, and $y_k(t) \in R^l$ represent the state, input, and output, respectively, $A \in R^{n\times n}$, $B \in R^{n\times m}$, and $C \in R^{l\times n}$ are known parameter matrices.

The system is controlled to track a known desired trajectory $y_d(t)$. For any realizable trajectory and appropriate initial conditions, there exists a unique input $u_d(t)$ generating the trajectory $y_d(t)$, which can be of the form

$$\begin{cases} x_d(t+1) = Ax_d(t) + Bu_d(t), \\ y_d(t) = Cx_d(t), \end{cases} \tag{2}$$

where $x_d(t)$ is the desired state. In order to track $y_d(t)$ accurately, various ILC schemes have been proposed, and the typical one can be represented as

$$u_{k+1}(t) = u_k(t) + \Gamma(t)e_k(t+1), \tag{3}$$

where $\Gamma(t) \in R^{r\times l}$ denotes learning gain, $e_k(t) = y_d(t) - y_k(t)$ is output error, $t \in [0, 1, \ldots, T-1]$. If $\Gamma(t)$ is selected to satisfy the condition $\|\mathrm{I} - \Gamma(t)CB\| \le \rho < 1$, the ILC algorithm converges as the number of iterations goes to infinity. It is shown in [5] that a choice of this learning gain matrix requires the matrix CB to be full-column rank.

The ILC system with output data transmitted from the sensor to the controller and input data transmitted from the controller to the actuator through the network is illustrated in Fig. 1.

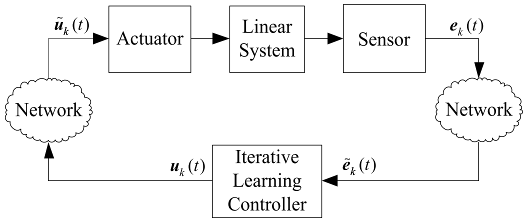


Figure 1 Diagram of networked ILC system

Taking the effect of data dropout in input and output into account, the system can be represented as follows:

$$\begin{cases} x_k(t+1) = Ax_k(t) + B\tilde{u}_k(t), \\ y_k(t) = Cx_k(t), \end{cases} \tag{4}$$

$$u_{k+1}(t) = u_k(t) + \Gamma(t)\tilde{e}_k(t+1), \tag{5}$$

where $\tilde{u}_k(t)$ is the input data received at the actuator side and $\tilde{e}_k(t+1)$ is the output data received at the controller side, which can be expressed as

$$\tilde{u}_k(t) = \xi_k(t)u_k(t), \tag{6}$$

$$\tilde{e}_k(t+1) = \eta_k(t)e_k(t+1). \tag{7}$$

$\xi_k(t)$ and $\eta_k(t)$ are two scalar Bernoulli distributed random variables taking values 0 or 1, which means $\xi_k(t), \eta_k(t) \in \{0,1\}$, $\forall k, t$. Moreover, $\xi_k(t)$ is uncorrelated with $\eta_k(t)$. In this expression, if the variable takes value 0, then the data is dropped correspondingly; otherwise the data is not dropped. It can be easily seen that the lost data would greatly damage the tracking performance of networked ILC systems, which means the lost data must be compensated with a suitable data to guarantee the tracking performance of the plant. According to the characteristic of ILC system which operates in time and iteration domains simultaneously, the lost data could be compensated in time or iteration domain. In order to point out the difference therein, a comparison of compensation methods for dropped input data in networked ILC systems would be done in the next section.

3 A comparison of compensation methods

In this section, the convergence of networked ILC system with the two compensation methods is compared. Specially, the comparison would be done through analyzing the varieties of eigenvalues and the elements in the lower triangular of the transition matrix of input error at the controller side. In the analysis, we assume that the input data $u_k(t)$ is dropped during transmission and compensated by $u_k(t-1)$ in time domain or $u_{k-1}(t)$ in iteration domain, which means $\tilde{u}_k(t) = u_k(t-1)$ or $\tilde{u}_k(t) = u_{k-1}(t)$, respectively. In order to simplify the analysis, the following assumption is made:

Assumption 1 $x_k(0) = x_d(0)$, $\forall k$, $u_1(t) = 0$, $\forall t$.

3.1 Compensation in time domain

When the dropped input data $u_k(t)$ is compensated in time domain, $\tilde{u}_k(t) = u_k(t-1)$ at the actuator side. According to $e_k(t) = y_d(t) - y_k(t)$, the output error $e_k(t)$ can be represented as

$$\begin{aligned} e_k(t+1) &= y_d(t+1) - y_k(t+1) \\ &= C\delta x_k(t+1). \end{aligned} \tag{8}$$

Using (2), (4), and Assumption 1, the relationship between state error $\delta x_k(t)$ and input error $\delta u_k(t)$ at the controller side can be expressed as

$$\begin{aligned}\delta x_k(t+1) &= x_d(t+1) - x_k(t+1)\\ &= A\delta x_k(t) + B\delta\tilde{u}_k(t)\\ &= A\delta x_k(t) + B\bigl(\xi_k(t)\delta u_k(t) + \bigl(1-\xi_k(t)\bigr)\delta u_k(t-1)\bigr)\\ &= \sum_{i=0}^{t-1} A^{t-i}B\bigl(\xi_k(i)\delta u_k(i) + \bigl(1-\xi_k(i)\bigr)\delta u_k(i-1)\bigr).\end{aligned} \tag{9}$$

Now we derive the expression of input error $\delta u_k(t)$ in iteration domain. From (5), (8), (9), and $\tilde{e}_k(t+1) = e_k(t+1)$, we have

$$\begin{aligned}\delta u_{k+1}(t) &= u_d(t) - u_{k+1}(t)\\ &= u_d(t) - u_k(t) - \Gamma(t)\tilde{e}_k(t+1)\\ &= \delta u_k(t) - \Gamma(t)C\delta x_k(t+1)\\ &= \delta u_k(t) - \Gamma(t)\sum_{i=0}^{t} CA^{t-i}B\bigl(\xi_k(i)\delta u_k(i) + \bigl(1-\xi_k(i)\bigr)\delta u_k(i-1)\bigr).\end{aligned} \tag{10}$$

If the input data $u_k(t)$ is dropped, the random parameter $\xi_k(t) = 0$ correspondingly. With the method to compensate the dropped input data in time domain, $u_k(t-1)$ would be used to replace dropped $u_k(t)$, then (10) would be changed into

$$\delta u_{k+1}(t) = \delta u_k(t) - \Gamma(t)CB\delta u_k(t-1) - \sum_{i=1}^{t}\Gamma(t)CA^{i}B\delta u_k(t-i). \tag{11}$$

Furthermore, the norm of input error at the controller side can be rewritten as $\|\psi_{k+1}\| \le \|H_k\|\|\psi_k\|$, where input error vector ψ_k and transition matrix H_k can be represented as (12) and (13):

$$\psi_k = \bigl[\bigl\|\delta u_k(0)\bigr\|, \bigl\|\delta u_k(1)\bigr\|, \ldots, \bigl\|\delta u_k(T-1)\bigr\|\bigr]^T, \tag{12}$$

$$H_k = \begin{bmatrix} I-\Gamma(0)CB & 0 & 0 & 0 & 0 & \cdots & 0\\ \Gamma(1)CAB & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots\\ \vdots & \vdots & \ddots & 0 & \ddots & \vdots & \vdots\\ \Gamma(t)CA^{t}B & \vdots & \Gamma(t)CAB+\Gamma(t)CB & I & 0 & \vdots & \vdots\\ \vdots & \vdots & \Gamma(t)CA^2B+\Gamma(t)CAB & 0 & \ddots & 0 & \vdots\\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0\\ \Gamma(T-1)CA^{T-1}B & \cdots & \Gamma(t)CA^{T-t}B+\Gamma(t)CA^{T-t-1}B & 0 & \cdots & \Gamma(T-1)CAB & I-\Gamma(T-1)CB \end{bmatrix}. \tag{13}$$

When (13) is compared with the expression of transition matrix of input error at the controller side in ideal situation, it can be easily seen that the compensation method brings some changes in the value of transition matrix from $(t+1)$st to (T)st rows. First, these element values in $(t+1)$st column are reduced. Specifically, the eigenvalue in $(t+1)$st column is changed from $I - \Gamma(t)CB$ to I, and the elements in $(t+1)$st column from $(t+2)$st to (T)st rows are reduced into 0. Second, the elements in (t)st column from $(t+1)$st to (T)st rows are increased with $\Gamma(t)CA^iB$, $0 \le i \le T-t-1$. Interestingly, in each row from $(t+1)$st to (T)st, the increment of element in (t)st column just equals to the decrement of the right adjacent element in $(t+1)$st column. This phenomenon is caused by the compensation method in time domain, which uses $u_k(t-1)$ to replace dropped $u_k(t)$. Based on these two discoveries, it can be easily seen that the transition matrix is still a lower triangular, all its eigenvalues are the diagonal elements with one eigenvalue being I and other eigenvalues are $\mathrm{I} - \Gamma(i)CB$, $i \in [0,1,\ldots,t,t+2,\ldots,T-1]$. If the learning gain is satisfied $\|\mathrm{I} - \Gamma(i)CB\| \le \rho < 1$, $i \in [0,1,\ldots,t,t+2,\ldots,T-1]$, the asymptotic convergence property of input error at the controller side can also be guaranteed along the iteration axis.

It is important to point out that this compensation method cannot guarantee that the input error at the actuator side converges to zero, because the converged input data at the controller side would suffer data drop again during transmission from the controller to the actuator. In other words, this method guarantees $\lim_{k\to\infty} \|\delta u_k(t)\| = 0$, $t \in [0, T-1]$ at the controller side, while cannot guarantee $\lim_{k\to\infty} \|\delta \tilde{u}_k(t)\| = 0$, $t \in [0, T-1]$ at the actuator side, because $\lim_{k\to\infty} \|\delta \tilde{u}_k(t)\| = \lim_{k\to\infty} \|u_d(t) - u_k(t-1)\| \ne 0$ when $u_k(t)$ is dropped and compensated in time domain at the actuator side. Due to the introduction of networks, the convergence of output error is determined by the input error at the actuator side. So, this method cannot guarantee that the output error of networked ILC systems converges to zero as iteration number goes on.

3.2 Compensation in iteration domain

When the dropped input data $u_k(t)$ is compensated in iteration domain, $\tilde{u}_k(t) = u_{k-1}(t)$ at the actuator side. With this compensation method, the transition matrix of input error $u_k(i)$, $t \le i \le T-1$ at the controller side would be changed. In this part, the transition matrix of $\delta u_{k+1}(t)$ would be derived first, and then transition matrix of $\delta u_{k+1}(i)$, $t+1 \le i \le T-1$ would be got.

3.2.1 Transition matrix of $\delta u_{k+1}(t)$

If input data $u_k(t)$ is dropped and compensated by $u_{k-1}(t)$, the general expression of input error in (10) can be easily rewritten as

$$\delta u_{k+1}(t) = \delta u_k(t) - \Gamma(t)CB\delta u_{k-1}(t) - \Gamma(t)\sum_{i=0}^{t-1} CA^{t-i}B\delta u_k(i). \tag{14}$$

Using (14), we can show $\delta u_{k+1}(t)$ by means of input error in $(k-1)$st iteration, which is given by

$$\begin{aligned}\delta u_{k+1}(t) = {}& \bigl(I - 2\Gamma(t)CB\bigr)\delta u_{k-1}(t) - \Gamma(t)\sum_{i=0}^{t-1} CA^{t-i}B\delta u_{k-1}(i) \\ & - \Gamma(t)\sum_{i=0}^{t-1} CA^{t-i}B\left(\delta u_{k-1}(i) - \Gamma(i)\sum_{j=0}^{i} CA^{i-j}B\delta u_{k-1}(j)\right).\end{aligned} \tag{15}$$

Furthermore, (15) could be rewritten in the form of transition matrix

$$
\begin{aligned}
\delta u_{k+1}(t) = {}& \begin{bmatrix} \Gamma(t)CA^{t}B \\ \vdots \\ \vdots \\ \Gamma(t)CAB \\ I \end{bmatrix}^{T} \\
&\times \begin{bmatrix} I-\Gamma(0)CB & 0 & \cdots & \cdots & 0 \\ \Gamma(1)CAB & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \Gamma(t-1)CA^{t-1}B & \cdots & \Gamma(t-1)CAB & I-\Gamma(t-1)CB & 0 \\ \Gamma(t)CA^{t}B & \cdots & \cdots & \Gamma(t)CAB & I-2\Gamma(t)CB \end{bmatrix} \\
&\times \begin{bmatrix} \delta u_{k-1}(0) \\ \vdots \\ \vdots \\ \vdots \\ \delta u_{k-1}(t) \end{bmatrix}.
\end{aligned}
\tag{16}
$$

3.2.2 *Transition matrix of $\delta u_{k+1}(i)$, $t+1 \le i \le T-1$*

Similarly, $\delta u_{k+1}(t+1)$ can be expressed as

$$
\begin{aligned}
\delta u_{k+1}(t+1) = {}& \delta u_{k}(t+1) \\
& - \Gamma(t+1)\sum_{i=0}^{t+1} CA^{t+1-i}B\bigl(\xi_k(i)\delta u_k(i) + \bigl(1-\xi_k(i)\bigr)\delta u_{k-1}(i)\bigr).
\end{aligned}
\tag{17}
$$

With the compensation method in iteration domain, (17) could be further rewritten as

$$
\begin{aligned}
\delta u_{k+1}(t+1) = {}& \bigl(I-\Gamma(t+1)CB\bigr)\delta u_k(t+1) \\
& - \Gamma(t)CAB\delta u_{k-1}(t) - \Gamma(t)\sum_{i=0}^{t-1} CA^{t-i}B\delta u_k(i).
\end{aligned}
\tag{18}
$$

Expressing $\delta u_{k+1}(t+1)$ by means of input error in $(k-1)$st iteration, we have

$$
\begin{aligned}
&\delta u_{k+1}(t+1) \\
&\quad = \bigl(I-\Gamma(t+1)CB\bigr)\left(\delta u_{k-1}(t+1) - \Gamma(t+1)\sum_{i=0}^{t+1} CA^{t+1-i}B\delta u_{k-1}(i)\right) \\
&\qquad - \Gamma(t+1)CAB\delta u_{k-1}(t) \\
&\qquad - \Gamma(t+1)\sum_{i=0}^{t-1} CA^{t+1-i}B\left(\delta u_{k-1}(i) - \Gamma(i)\sum_{j=0}^{i} CA^{i-j}B\delta u_{k-1}(j)\right).
\end{aligned}
\tag{19}
$$

In the same way, it can be easy to derive the expression of $\|\delta u_{k+1}(i)\|$, $t+2 \leq i \leq T-1$. According to the expression of $\|\delta u_{k+1}(i)\|$, $t+1 \leq i \leq T-1$, the transition matrices H_k and H_{k-1} can be described as

$$\begin{aligned} H_k &= \begin{bmatrix} \Gamma(t+1)CA^{t+1}B & \cdots & \Gamma(t+1)CAB & I-\Gamma(t+1)CB & 0 & \cdots & 0 \\ \vdots & \cdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \ddots & 0 \\ \Gamma(T-1)CA^{T-1}B & \cdots & \cdots & \cdots & \cdots & \Gamma(T-1)CAB & I-\Gamma(T-1)CB \end{bmatrix}, \end{aligned} \tag{20}$$

$$\begin{aligned} H_{k-1} &= \begin{bmatrix} I-\Gamma(0)CB & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \Gamma(1)CAB & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & 0 & & \ddots & \ddots & \vdots \\ \Gamma(t-1)CA^{t-1}B & \cdots & \Gamma(t-1)CB & I-\Gamma(t-1)CB & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & I & 0 & \ddots & \vdots \\ \Gamma(t+1)CA^{t+1}B & \cdots & \cdots & \Gamma(t)CAB & I-\Gamma(t+1)CB & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ \Gamma(T-1)CA^{T-1}B & \cdots & \cdots & \cdots & \cdots & \cdots & \Gamma(T-1)CAB & I-\Gamma(T-1)CB \end{bmatrix}. \end{aligned} \tag{21}$$

When (16), (20), and (21) are compared with those in ideal situation, it can be easily seen that compensation in iteration domain also brings some changes in the value of transition matrix of input error at the controller side. First, the eigenvalue which determines the convergence speed of $\|\delta u_{k+1}(t)\|$ is changed to $I-2\Gamma(t)CB$ in (16). If learning gain $\Gamma(t)$ is selected to satisfy the condition so that $0 < \rho(\Gamma(t)CB) < 1$ is satisfied, then $\|I-\Gamma(t)CB\| < 1$ and $\|I-2\Gamma(t)CB\| < 1$, which could guarantee the convergence of $\|\delta u_{k+1}(t)\|$. Second, some element values in $(t+1)$st row in (21) are changed. Specifically, the eigenvalue in this row is changed to I, and other elements on the left-hand side of this eigenvalue are all changed into 0. These two changes correspond to compensation method in iteration domain which uses $u_{k-1}(t)$ to replace the dropped $u_k(t)$. Based on these two discoveries, it can be easily seen that the convergence of input error at the controller side is still guaranteed.

Furthermore, the input error at the actuator side also converges to zero, because the used input data at the actuator side is equal to the dropped input data when the input error at the controller side converges. That is to say, compensation in iteration domain not only guarantees $\lim_{k\to\infty} \|\delta u_k(t)\| = 0$, $t \in [0, T-1]$ at the controller side, but also guarantees $\lim_{k\to\infty} \|\delta \tilde{u}_k(t)\| = 0$, $t \in [0, T-1]$ at the actuator side, because $\lim_{k\to\infty} \|\delta \tilde{u}_k(t)\| = \lim_{k\to\infty} \|u_d(t) - u_{k-1}(t)\| = 0$ when $u_k(t)$ is dropped and compensated in iteration domain at the actuator side. Naturally, compensation in iteration domain guarantees that the output error converges to zero as the number of iterations goes to infinity.

4 Simulation

In this section, some numerical examples are given to illustrate the correctness of the results derived theoretically in the last section. Consider system (4) with matrices given by

$$x_k(t+1) = \begin{bmatrix} -0.5 & 0 & 0 \\ 1 & 1.24 & -0.87 \\ 0 & 0.87 & 0 \end{bmatrix} x_k(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tilde{u}_k(t),$$
$$y_k(t) = \begin{bmatrix} 2 & 2.6 & -2.8 \end{bmatrix} x_k(t). \tag{22}$$

The desired trajectory is

$$y_d(t) = 5\sin\bigl[8(t-1)/T\bigr]. \tag{23}$$

The P-style ILC method is described in (5). Initial state error $\delta x_k(0)$ and initial input error $\delta u_0(t)$ are 0, respectively. $T = 200$, $\Gamma(t) = 0.2$, so $0 < \rho(\Gamma(t)CB) = 0.4 < 1$ is satisfied. The mean of output errors in each iteration is used to demonstrate the convergence of output error. Stochastic parameter $\xi_k(t)$ is a Bernoulli variable taking the values of 0 and 1 with $\mathrm{Prob}\{\xi_k(t) = 1\} = E\{\xi_k(t) = 1\} = \alpha$. In this part, the simulation is considered with two cases as $\alpha = 0.97$ and $\alpha = 0.94$, respectively.

In order to compare the convergence of networked ILC systems with dropped input data compensated in time or iteration domains, the convergence of system without compensation is used as a benchmark, which is shown in Fig. 2. From Fig. 2, it can be easily seen that the effect of input data dropouts increases generally with the increase of input data dropout rate. Figure 3 shows the mean of output errors with compensation in time domain. From Fig. 3, it can be easily found that the effect of random input data dropouts on the convergence of the output errors' mean is reduced significantly with two different data dropout rates, while both means of output errors cannot converge to zero. Figure 4 shows the mean of output errors with compensation in iteration domain. From Fig. 4, it can be easily seen that the mean of output errors converges to zero, whether the data dropout

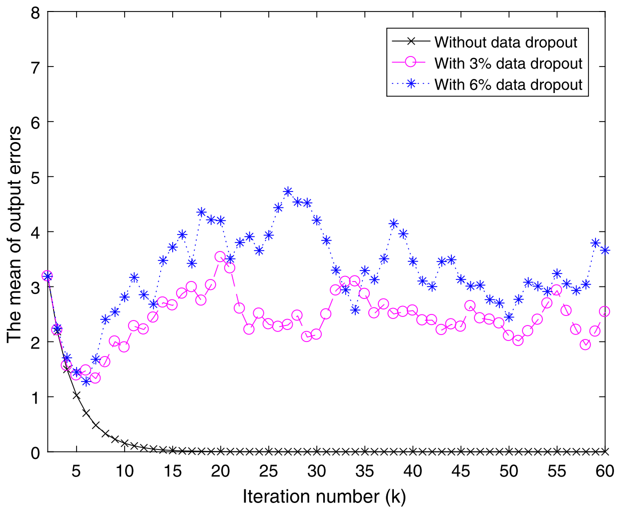


Figure 2 The mean of output errors without compensation

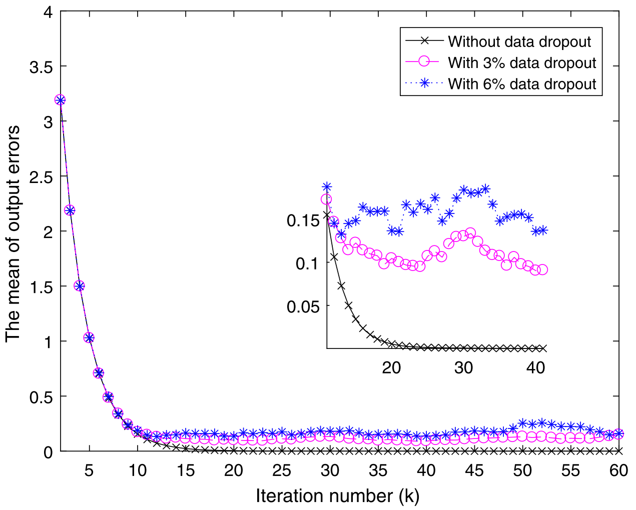


Figure 3 The mean of output errors compensated in time domain

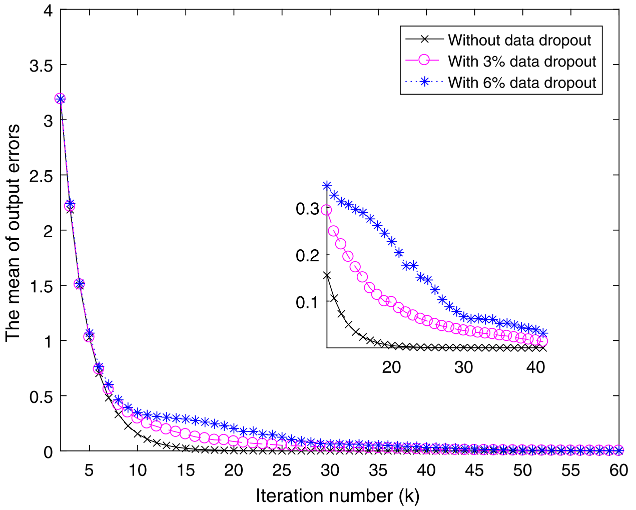
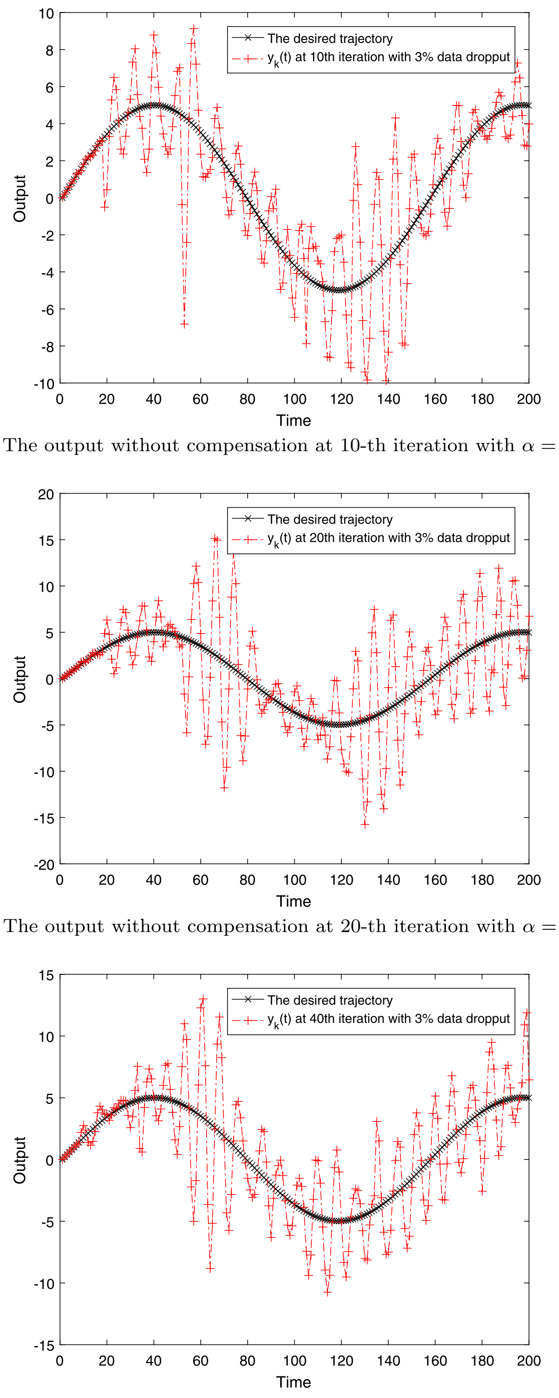


Figure 4 The mean of output errors compensated in iteration domain

rate is $\alpha = 0.97$ or $\alpha = 0.94$. What is more, the convergence speed is decreased with the increase of dropout rate. Through comparing these three figures, it can be easily found that theoretical analysis in the last section is corroborated. Figures 5–7 show the system output trajectories without compensation, with compensation in time domain, and with compensation in iteration domain at three different iterations, respectively, which further verifies the theoretical analysis in the last section from the perspective of trajectories in different iterations.

5 Conclusion

In this paper, two compensation methods for random input data dropouts in the networked ILC systems are compared. Through analyzing the variety of eigenvalues and elements in the lower triangular of the transition matrix of input error at the controller side,



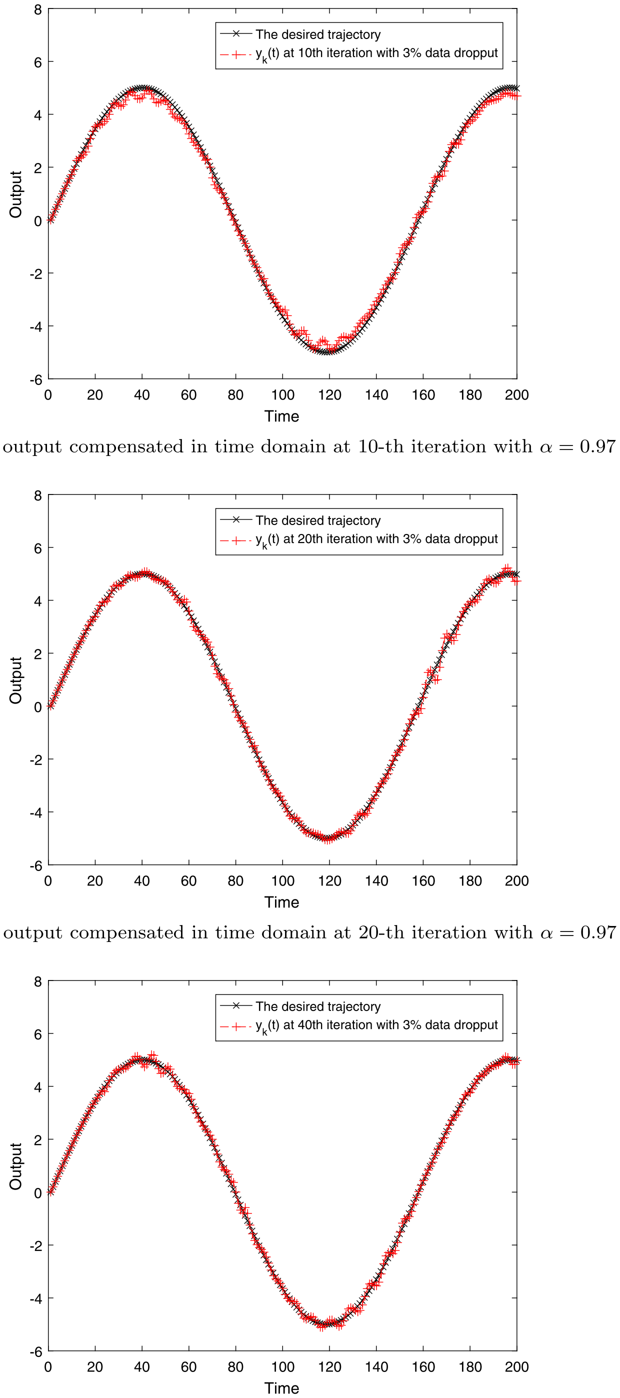
(a) The output without compensation at 10-th iteration with $\alpha = 0.97$

(b) The output without compensation at 20-th iteration with $\alpha = 0.97$

(c) The output without compensation at 40-th iteration with $\alpha = 0.97$

Figure 5 The output without compensation at different iterations with $\alpha = 0.97$

it can be easily found that the two compensation methods both can guarantee the convergence of input errors at the controller side. What is more, the method in iteration domain also guarantees the convergence of input errors at the actuator side, while the method in time domain cannot do that. The reason is that converged input data at the controller side would be dropped again during transmission. The convergence of output error is deter-



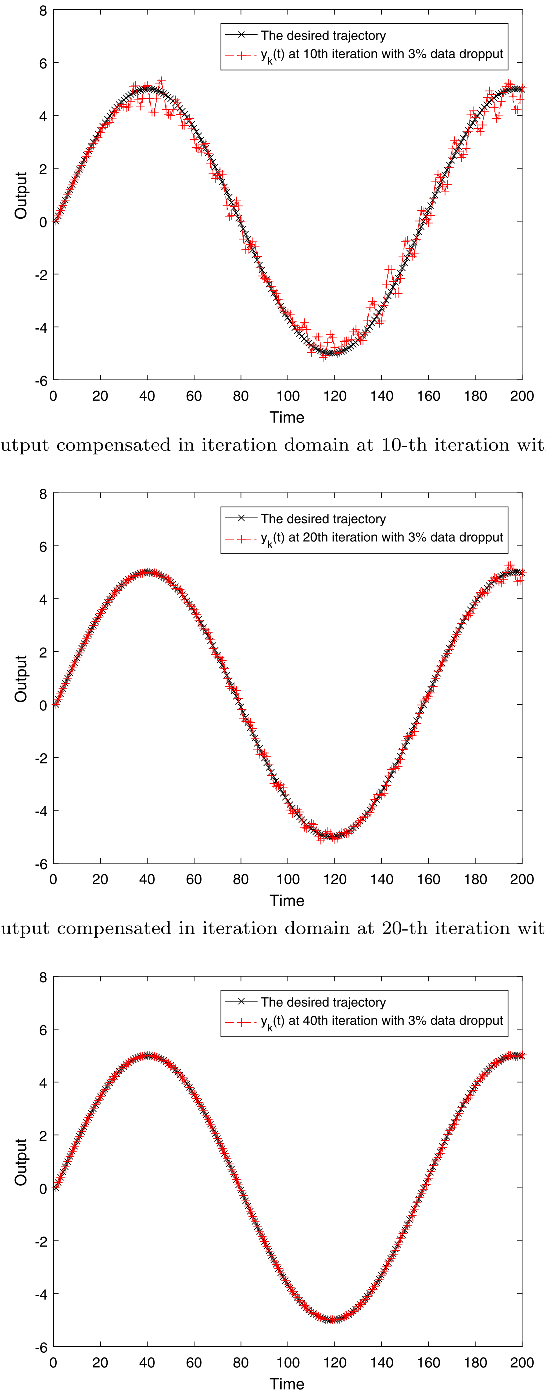
(a) The output compensated in time domain at 10-th iteration with $\alpha = 0.97$

(b) The output compensated in time domain at 20-th iteration with $\alpha = 0.97$

(c) The output compensated in time domain at 40-th iteration with $\alpha = 0.97$

Figure 6 The output compensated in time domain at different iterations with $\alpha = 0.97$

mined by input error at the actuator side; consequently, only the method to compensate the dropped input data in iteration domain guarantees that the output error of networked ILC systems converges to zero.



(a) The output compensated in iteration domain at 10-th iteration with $\alpha = 0.97$

(b) The output compensated in iteration domain at 20-th iteration with $\alpha = 0.97$

(c) The output compensated in iteration domain at 40-th iteration with $\alpha = 0.97$

Figure 7 The output compensated in iteration domain at different iterations with $\alpha = 0.97$

Acknowledgements
The authors wish to express their gratitude for all the reviewers' insightful comments and suggestions to improve this manuscript.

Funding
The authors sincerely appreciate the support of the National Natural Science Foundation of China (Nos. 61771432, 61302118 and U1604151), Scientific and Technological Project of Henan Educational Committee (Nos. 17A510005 and

18B510019), Scientific and Technological Project of Henan Province (182102210610), Doctor Fund of Zhengzhou University of Light Industry (2013BSJJ049), Outstanding Talent Program of Science and Technology Innovation in Henan Province (174200510008), and the Program for Scientific and Technological Innovation Team in Universities of Henan Province (16IRTSTHN029).

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
HLX, DHQ, ZZ, and ZQW carried out the main part of this article. SLJ corrected and revised the manuscript. All authors have read and approved the final manuscript.

Author details
[1]School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, P.R. China. [2]School of Electrical Engineering, Henan University of Technology, Zhengzhou, P.R. China.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 21 June 2018 Accepted: 16 December 2018 Published online: 21 February 2019

References

1. Arimoto, S., Kawamura, S., Miyazaki, F.: Bettering operation of robots by learning. J. Robot. Syst. **1**(2), 123–140 (1984)
2. Bristow, D.A., Tharayil, M., Alleyne, A.G.: A survey of iterative learning control. IEEE Control Syst. **26**(3), 96–114 (2006)
3. Ahn, H.-S., Chen, Y., Moore, K.L.: Iterative learning control: brief survey and categorization. IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev. **37**(6), 1099–1121 (2007)
4. Shen, D., Wang, Y.: Survey on stochastic iterative learning control. J. Process Control **24**(12), 64–77 (2014)
5. Saab, S.S.: A discrete-time stochastic learning control algorithm. IEEE Trans. Autom. Control **46**(6), 877–887 (2001)
6. Huang, L., Fang, Y., Wang, T.: Method to improve convergence performance of iterative learning control systems over wireless networks in presence of channel noise. IET Control Theory Appl. **8**(3), 175–182 (2014)
7. Huang, L., Fang, Y., Wang, T.: Convergence analysis of wireless remote iterative learning control systems with channel noise. Asian J. Control **17**(6), 2374–2381 (2015)
8. Sun, M., Wang, D.: Initial shift issues on discrete-time iterative learning control with system relative degree. IEEE Trans. Autom. Control **48**(1), 144–148 (2003)
9. Park, K.-H., Bien, Z.: A generalized iterative learning controller against initial state error. Int. J. Control **73**(10), 871–881 (2000)
10. Tayebi, A., Zaremba, M.B.: Robust iterative learning control design is straightforward for uncertain LTI systems satisfying the robust performance condition. IEEE Trans. Autom. Control **48**(1), 101–106 (2003)
11. Ahn, H.-S., Moore, K.L., Chen, Y.: Stability analysis of discrete-time iterative learning control systems with interval uncertainty. Automatica **43**(5), 892–902 (2007)
12. Butcher, M., Karimi, A., Longchamp, R.: A statistical analysis of certain iterative learning control algorithms. Int. J. Control **81**(1), 156–166 (2008)
13. Nguyen, D.H., Banjerdpongchai, D.: A convex optimization design of robust iterative learning control for linear systems with iteration-varying parametric uncertainties. Asian J. Control **13**(1), 75–84 (2011)
14. You, K., Li, Z., Quevedo, D.E., Lewis, F.L.: Recent developments in networked control and estimation. IET Control Theory Appl. **8**(18), 2123–2125 (2014)
15. You, K.: Control over communication networks: a personal perspective. In: Intelligent Control and Automation (WCICA), 2014 11th World Congress on, pp. 1304–1309 (2014)
16. Shen, D., Xu, J.-X.: A framework of iterative learning control under random data dropouts: mean square and almost sure convergence. Int. J. Adapt. Control Signal Process. **31**(12), 1825–1852 (2017)
17. Ahn, H.-S., Chen, Y., Moore, K.L.: Intermittent iterative learning control. In: Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE, pp. 832–837 (2006)
18. Ahn, H.-S., Moore, K.L., Chen, Y.: Discrete-time intermittent iterative learning controller with independent data dropouts. In: Proc. the 2008 IFAC World Congress, pp. 12442–12447 (2008)
19. Ahn, H.-S., Moore, K.L., Chen, Y.: Stability of discrete-time iterative learning control with random data dropouts and delayed controlled signals in networked control systems. In: Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on, pp. 757–762 (2008)
20. Ahn, H.-S., Moore, K.L., Chen, Y.: Iterative learning control for batch processes with missing measurements. In: Proc. of the Symposium on Learning Control at CDC2009. Shanghai Jiaotong University, Shanghai (2009)
21. Pan, Y.-J., Marquez, H.J., Chen, T., Sheng, L.: Effects of network communications on a class of learning controlled non-linear systems. Int. J. Syst. Sci. **40**(7), 757–767 (2009)
22. Bu, X., Hou, Z., Yu, F., Wang, F.: H-infinity iterative learning controller design for a class of discrete-time systems with data dropouts. Int. J. Syst. Sci. **45**(9), 1902–1912 (2014)
23. Huang, L., Fang, Y.: Convergence analysis of wireless remote iterative learning control systems with dropout compensation. Math. Probl. Eng. **2013**, 1–9 (2013)
24. Shen, D., Zhang, C., Xu, Y.: Two updating schemes of iterative learning control for networked control systems with random data dropouts. Inf. Sci. **381**, 352–370 (2017)
25. Bu, X., Yu, F., Hou, Z., Wang, F.: Iterative learning control for a class of nonlinear systems with random packet losses. Nonlinear Anal., Real World Appl. **14**(1), 567–580 (2013)

26. Liu, J., Ruan, X.: Networked iterative learning control for discrete-time systems with stochastic packet dropouts in input and output channels. Adv. Differ. Equ. **2017**(53), 1 (2017)
27. Jin, Y., Shen, D.: Iterative learning control for nonlinear systems with data dropouts at both measurement and actuator sides. Asian J. Control **20**(4), 1624–1636 (2018)
28. Shen, D., Jin, Y., Xu, Y.: Learning control for linear systems under general data dropouts at both measurement and actuator sides: a Markov chain approach. J. Franklin Inst. **354**(13), 5091–5109 (2017)
29. Shen, D., Xu, J.-X.: A novel Markov chain based ILC analysis for linear stochastic systems under general data dropouts environments. IEEE Trans. Autom. Control **62**(11), 5850–5857 (2017)
30. Shen, D.: Data-driven learning control for stochastic nonlinear systems: multiple communication constraints and limited storage. IEEE Trans. Neural Netw. Learn. Syst. **29**(6), 2429–2440 (2018)