

# Kernel-based auto-associative P-type iterative learning control strategy

LAN Tianyi, LIN Hui\*, and LI Bingqiang

School of Automation, Northwestern Polytechnical University, Xi'an 710129, China

**Abstract:** In order to accelerate the convergence speed of iterative learning control (ILC), taking the P-type learning algorithm as an example, a correction algorithm with kernel-based auto-associative is proposed for the linear system. The learning mechanism of human brain associative memory is introduced to the traditional ILC. The control value of the subsequent time is pre-corrected with the current time information by association in each iterative learning process. The learning efficiency of the whole system is improved significantly with the proposed algorithm. Through the rigorous analysis, it shows that under this new designed ILC scheme, the uniform convergence of the state tracking error is guaranteed. Numerical simulations illustrate the effectiveness of the proposed associative control scheme and the validity of the conclusion.

**Keywords:** iterative learning control (ILC), associative learning, convergence speed, tracking convergence.

**DOI:** 10.23919/JSEE.2020.000015

## 1. Introduction

Iterative learning control (ILC) [1] has many attractive characteristics: (i) it is a very simple control learning algorithm, (ii) it can track any linear or nonlinear unknown periodic systems, and (iii) the control performance will be better and better with the increasing of the number of trials. The theoretical results are fruitful in the past 40 years, such as the convergence of learning algorithms [2,3], initial state problems [4,5], robustness problems [6], new learning algorithms [7,8] and so on. Recently, it has aroused great interests of control engineers, and has been paid more and more attention in the control field [9–14].

Although a series of theoretical research results have been obtained, ILC has not been widely used. In order to make ILC widely used in industry, many experts and

scholars improve the ILC from different perspectives, which include two aspects, namely, the model-free method and the model-based approach. Model-free improvement methods include adaptive ILC [14–16], PID-type control [17,18], accelerated learning algorithm [19,20], and high-order ILC [9,21]. Model-free improvement methods can retain the advantages of the traditional ILC without relying on model parameters. It has good robustness, but the controller parameters need to be adjusted according to experience and the controller only includes the output error so that it is difficult to obtain the optimal control effect. The model-based approach mainly includes the feedback learning algorithm [22,23], and predictive control [24]. In [22], an ILC method based on optimal feedback and feed-forward was proposed, the inverse system of the model was used in iterative learning controller design to obtain a better control performance, but the robust performance of the controller was reduced due to the introduction of model parameters. According to the predictive control method [24], the algorithm updates the control input by learning not only from the past trials but also from the predicted future trials by using the knowledge of the system model. This algorithm includes information from the predicted future trials, and the designed ILC controller properties reveal potentially substantial benefit in terms of the convergence speed.

The mechanism of human learning indicates that humans have a strong ability to learn, which is inseparable with the associative function of the human brain [25]. Association is able to obtain the complete information model of objects by using the partial information. Conceiving overall part information can draw analogies, which effectively accelerates the learning process. In accordance with the mechanism of human association, some new learning schemes were proposed based on a kernel-based auto-associative method combined with the ILC to improve the learning speed of the tradition algorithm [26–28]. The algorithms have the ability of association in each iterative learning process and will greatly accelerate the conver-

---

Manuscript received May 27, 2019.

\*Corresponding author.

This work was supported by the National Natural Science Foundation of China (51777170), the Aeronautical Science Foundation of China (20162853026), and the Project Supported by Natural Science Basic Research Plan in Shanxi Province of China (2019JM-462; 2020JM-151).

gence of the learning speed.

This paper is organized as follows. Section 2 describes the kernel-based auto-association ILC algorithm in general. The mathematical proof of the convergence of the proposed algorithm is given in Section 3. Further, the proposed ILC law is extended to linear time-varying systems in Section 4. Two simulation examples are carried out to compare the proposed algorithm with the traditional algorithm, and the simulation results are given in Section 5 to verify the validity and correctness of the proposed algorithm. The final conclusions are obtained in Section 6.

## 2. Kernel-based auto-association ILC

In this section, the ILC algorithm is formulated by an abstract function. As an example (and for simplicity), these formulations are presented for discrete time linear time-invariant and linear time-varying systems, however, the abstract problem setting is applied to more general linear systems in Hilbert spaces. This includes many situations of interest such as nonlinear time-varying systems, and nonlinear differential models [29–31].

Consider a class of the linear time-invariant (LTI) discrete system

$$\begin{cases} \mathbf{x}_k(j+1) = \mathbf{A}\mathbf{x}_k(j) + \mathbf{B}\mathbf{u}_k(j) \\ \mathbf{y}_k(j+1) = \mathbf{C}\mathbf{x}_k(j+1) \end{cases} \quad (1)$$

where  $N$  is the total number of samples;  $j \in S_{N-1}$  is the sampling step;  $S_N = \{0, 1, \dots, N\}$  is the discrete time interval; subscript  $k$  is the number of iterations (trials); and  $\mathbf{x}_k(j) \in \mathbf{R}^n$ ,  $\mathbf{u}_k(j) \in \mathbf{R}^r$ , and  $\mathbf{y}_k(j) \in \mathbf{R}^m$  are, respectively, the state, input, and output of the system;  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  are constant matrices of the corresponding dimension satisfying the condition, i.e.,  $\mathbf{CB}$  is of full column rank.

**Assumption 1** The initial state is equal to the ideal initial state, that is,  $\mathbf{x}_k(0) = \mathbf{x}_d(0)$ .

**Remark 1** The identical initialization condition is a standard assumption in the ILC design to ensure the perfect tracking performance. Note that without perfect initial conditions, perfect tracking can never be achieved. More discussions on various initial conditions in the learning context could be found in [32].

**Assumption 2** For any given desired trajectory  $\mathbf{y}_d(j+1)$ , there exists a desired state  $\mathbf{x}_d(j)$  and a desired control signal  $\mathbf{u}_d(j)$ , such that

$$\begin{cases} \mathbf{x}_d(j+1) = \mathbf{A}\mathbf{x}_d(j) + \mathbf{B}\mathbf{u}_d(j) \\ \mathbf{y}_d(j+1) = \mathbf{C}\mathbf{x}_d(j+1) \end{cases} \quad (2)$$

where  $\forall j \in S_{N-1}$ .

**Remark 2** Since  $\mathbf{u}_d(j)$  exists uniquely, the uniform convergence of the control profile  $\mathbf{u}_k(j)$  to  $\mathbf{u}_d(j)$  implies

that the state and output tracking errors will vanish. It is a reasonable assumption that the task for control should be feasible.

### 2.1 Kernel associative learning function

We present a new ILC algorithm which can greatly improve the learning speed of linear systems. The associative function is generated by the product of the predefined kernel function with the control correction value.

Define the kernel associative function

$$\mathbf{C}\mathbf{o}_k(i) = d(i, j)\Delta\mathbf{u}_k(i) \quad (3)$$

where  $i$  is the current time,  $k$  is the iteration number;  $\mathbf{C}\mathbf{o}_k(i)$  is the kernel associative function;  $d(i, j)$  is the kernel function;  $\Delta\mathbf{u}_k(i)$  is the correction value (in this paper, the correction value is error  $e_k(i)$ ). Product  $d(i, j)\Delta\mathbf{u}_k(i)$  denotes the correction value of  $\Delta\mathbf{u}_k(i)$  at point  $j$ .

In this paper,  $d(i, j) = e^{-K_E(\frac{j-i}{N})^2}$  is the negative exponential function,  $j \in S_N$ . Obviously, it can be found that the closer  $i$  to  $j$ , the bigger the value of the kernel function, and vice versa. According to the characteristics of association, the function is defined as follows:

$$\mathbf{C}\mathbf{o}_k(i) = \begin{cases} 0, & j \leq i \\ \mathbf{K}_L e^{-K_E(\frac{j-i}{N})^2} \Delta\mathbf{u}_k(i), & j > i \end{cases} \quad (4)$$

where  $K_E$  is a weight coefficient, and  $\mathbf{K}_L$  is an associative gain matrix.

The kernel function  $\mathbf{C}\mathbf{o}_k(i)$  is modified as shown in Fig. 1. Among them, the abscissa represents the time axis, and the learning interval  $[0, T]$  is divided into  $N$  segments, each segment length is  $\Delta t = T/N$ , the nodes are  $0, \Delta t, 2\Delta t, \dots, N\Delta t$ . For simplicity, it is briefly recorded as  $0, 1, 2, \dots, N$ .

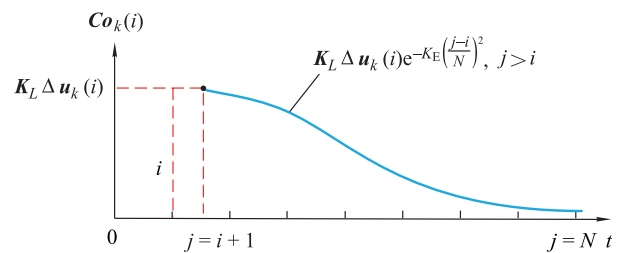


Fig. 1 Kernel associative learning function

### 2.2 P-type ILC associative algorithm design problem

Through analysis, it is clear that if a learning error occurs in the  $k$ th trial, the learning algorithm will produce a correction value  $\Delta\mathbf{u}_k(j)$  (in the traditional ILC, the correction  $\Delta\mathbf{u}_k(j)$  is the error  $e_k(j)$  or the derivative of error  $e_k(j) - e_k(j-1)$ ) and modify the value of the corresponding time point  $j$  for the next input, namely,  $\mathbf{u}_{k+1}(j) =$

$u_k(j) + \Delta u_k(j)$  (traditional P-type or D-type ILC algorithms). Therefore, the modified control input  $u_{k+1}(j)$  is used as a control input during the  $(k + 1)$ th trial.

From the above analysis, we can see that the traditional ILC is only a single point amendment to the current control point, and has no effect on other time points. Fig. 2 shows the structure of the P-type ILC associative algorithm.

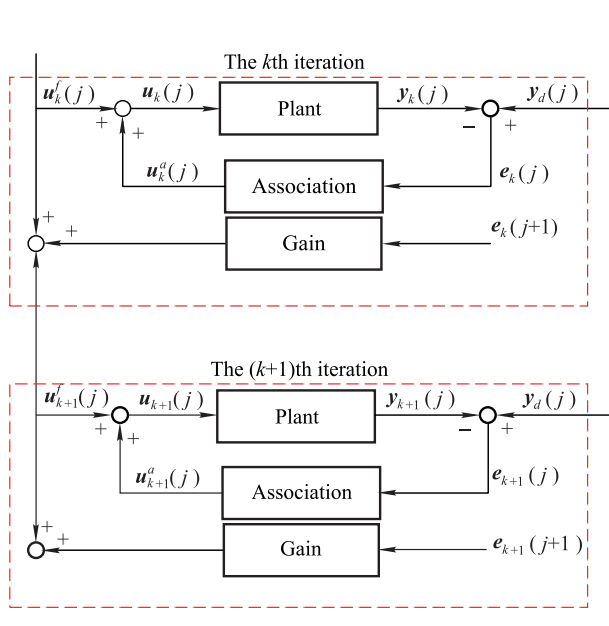


Fig. 2 Structure of P-type ILC associative algorithm

At the time point  $i = j$  of the  $k$ th trial, the error and the control input are computed by the traditional ILC. When  $j > i$ , the control value at points  $j + 1, j + 2, \dots, N$  will

be pre-corrected according to (3). The pre-correction values of this time are superimposed into the corresponding point, without waiting for the  $(k + 1)$ th trial.

The ILC algorithm design problem can be stated as finding a control updating law

$$u_k(j) = u_k^f(j) + u_k^a(j) \tag{5}$$

$$u_k^f(j) = u_{k-1}(j) + K_P e_{k-1}(j+1) \tag{6}$$

$$u_k^a(j) = \sum_{i=0}^{j-1} d(i, j) e_k(i), \quad u_k^a(0) \equiv 0 \tag{7}$$

where  $u_k^f(j)$  denotes the feed-forward ILC part, and  $u_k^a(j)$  is an associative controller;  $K_P$  is a proportional learning gain matrix. The output tracking error is defined to be  $e_k(j) = y_d(j) - y_k(j)$ .

A simple example is illustrated in Fig. 3 to show the modification value of each point in the iteration domain. In the  $k$ th trial, the learning process is from point 0 to point  $N$ .

When running to  $i$ , the error of this point is  $e_k(i)$ , and the point of  $j > i$  needs to be pre-corrected, namely,  $j = i + 1, i + 2, \dots, N$ . Fig. 3 is the predictive corrected diagram of point  $i$ . Table 1 shows the predictive corrected values corresponding to Fig. 3.

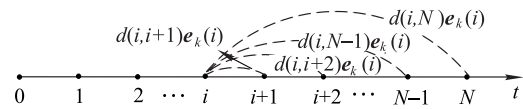


Fig. 3 The pre-associative diagram

Table 1 All of the predictive corrected values

| Error        | 1               | 2               | ...      | $i$             | $i + 1$             | ...      | $N$                     |
|--------------|-----------------|-----------------|----------|-----------------|---------------------|----------|-------------------------|
| $e_k(0)$     | $d(0, 1)e_k(0)$ | $d(0, 2)e_k(0)$ | ...      | $d(0, i)e_k(0)$ | $d(0, i + 1)e_k(0)$ | ...      | $d(0, N)e_k(0)$         |
| $e_k(1)$     | 0               | $d(1, 2)e_k(1)$ | ...      | $d(1, i)e_k(1)$ | $d(1, i + 1)e_k(1)$ | ...      | $d(1, N)e_k(1)$         |
| $e_k(2)$     | 0               | 0               | ...      | $d(2, i)e_k(2)$ | $d(2, i + 1)e_k(2)$ | ...      | $d(2, N)e_k(2)$         |
| $\vdots$     | $\vdots$        | $\vdots$        | $\ddots$ | $\vdots$        | $\vdots$            | $\ddots$ | $\vdots$                |
| $e_k(i)$     | 0               | 0               | ...      | 0               | $d(i, i + 1)e_k(i)$ | ...      | $d(i, N)e_k(i)$         |
| $\vdots$     | $\vdots$        | $\vdots$        | ...      | $\vdots$        | $\vdots$            | ...      | $\vdots$                |
| $e_k(N - 1)$ | 0               | 0               | ...      | 0               | 0                   | 0        | $d(N - 1, N)e_k(N - 1)$ |

Through the above description of associative learning, the control input of each point in the  $k$ th learning process can be easily obtained. For example, when  $j = 3$ , the associative algorithm of this point is expressed as follows:

$$u_k(3) = u_k^f(3) + u_k^a(3) = u_{k-1}(3) + K_P e_{k-1}(4) + d(0, 3)e_k(0) + d(1, 3)e_k(1) + d(2, 3)e_k(2) =$$

$$u_{k-1}(3) + K_P e_{k-1}(4) + \sum_{i=0}^2 d(i, 3)e_k(i). \tag{8}$$

**Remark 3** It can be seen that the ‘‘association’’ iterative learning algorithm proposed in this paper is based on the given kernel function (monotone decreasing function), using the error  $e_k(j)$  of the current time point to fine-tune the subsequent non-occurrence time, so as to achieve the rapid convergence.

To prove the convergence of the proposed associative

learning algorithm, the following important lemmas are given.

**Lemma 1** [33] Assume the matrix  $M \in \mathbf{R}^{n \times n}$ , for  $\forall \varepsilon > 0$ , then there exists a matrix norm  $\|\cdot\|$  induced by the vector norm on  $\mathbf{R}^n$ , it makes  $\|M\| \leq \rho(M) + \varepsilon$ , where the spectral radius  $\rho(M) = \max_{1 \leq i \leq n} \{|\lambda_i|\}$ ,  $\lambda_i$  ( $i = 1, 2, \dots, n$ ) is the eigenvalue of  $M$ .

**Lemma 2** If the matrices  $M \in \mathbf{R}^{n \times n}$  and  $N \in \mathbf{R}^{n \times n}$  are all upper-triangular matrices and the principal diagonal elements are equal, then the spectral radius of  $MN$  is less than or equal to the spectral radius of the product of  $M$  and  $N$ , that is,

$$\rho(MN) \leq \rho(M)\rho(N). \tag{9}$$

**Proof** Set matrices  $M$  and  $N$  to be

$$M = \begin{bmatrix} a & 0 & 0 & \cdots & 0 & 0 \\ a_{21} & a & 0 & \cdots & 0 & 0 \\ a_{31} & a_{32} & a & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & a_{n-1,3} & \cdots & a & 0 \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n-1} & a \end{bmatrix}$$

and

$$N = \begin{bmatrix} b & 0 & 0 & \cdots & 0 & 0 \\ b_{21} & b & 0 & \cdots & 0 & 0 \\ b_{31} & b_{32} & b & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{n-1,1} & b_{n-1,2} & b_{n-1,3} & \cdots & b & 0 \\ b_{n,1} & b_{n,2} & b_{n,3} & \cdots & b_{n,n-1} & b \end{bmatrix}$$

It can be seen from the characteristic of the proper value that the proper value of  $M$  is  $a$ , and the proper value of  $N$  is  $b$ .

$$MN =$$

$$\begin{bmatrix} a & 0 & 0 & \cdots & 0 & 0 \\ a_{21} & a & 0 & \cdots & 0 & 0 \\ a_{31} & a_{32} & a & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & a_{n-1,3} & \cdots & a & 0 \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n-1} & a \end{bmatrix} \times \begin{bmatrix} b & 0 & 0 & \cdots & 0 & 0 \\ b_{21} & b & 0 & \cdots & 0 & 0 \\ b_{31} & b_{32} & b & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{n-1,1} & b_{n-1,2} & b_{n-1,3} & \cdots & b & 0 \\ b_{n,1} & b_{n,2} & b_{n,3} & \cdots & b_{n,n-1} & b \end{bmatrix} =$$

$$\begin{bmatrix} ab & 0 & 0 & \cdots & 0 & 0 \\ * & ab & 0 & \cdots & 0 & 0 \\ * & * & ab & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & * & \cdots & ab & 0 \\ * & * & * & \cdots & * & ab \end{bmatrix}$$

According to the above, the proper value of  $MN$  is  $ab$ . According to the spectral radius definition, this obtains  $\rho(MN) \leq \rho(M)\rho(N)$ .  $\square$

### 3. Convergence analysis

**Theorem 1** Assume that the LTI system (1) satisfies Assumption 1 and Assumption 2, an ILC algorithm with kernel association (5) is adopted, if the condition holds:  $\rho(\mathbf{I} - \mathbf{K}_p \mathbf{C} \mathbf{B}) < 1$ . Then the output trajectory of the learning algorithm (5) converges uniformly to the desired trajectory; that is, when  $k \rightarrow \infty$ , such that  $\mathbf{y}_k(j) \rightarrow \mathbf{y}_d(j)$ ,  $j \in S_N$ .

**Proof** In this paper, let  $d(i, j) = e^{-K_E(\frac{j-i}{N})^2}$  prove the convergence of associative algorithm,  $j \in S_N$ .

Define

$$\delta \mathbf{u}_k(j) = \mathbf{u}_d(j) - \mathbf{u}_k(j), \tag{10}$$

$$\delta \mathbf{u}_k^f(j) = \mathbf{u}_d(j) - \mathbf{u}_k^f(j). \tag{11}$$

Thus, using (5) and (10) can give

$$\begin{aligned} \delta \mathbf{u}_k(j) &= \mathbf{u}_d(j) - \mathbf{u}_k(j) = \\ &= \mathbf{u}_d(j) - \mathbf{u}_k^f(j) - \mathbf{u}_k^a(j) = \\ &= \delta \mathbf{u}_k^f(j) - \mathbf{u}_k^a(j) = \\ &= \delta \mathbf{u}_{k-1}(j) - \mathbf{K}_P \mathbf{e}_{k-1}(j+1) - \mathbf{u}_k^a(j) \end{aligned} \tag{12}$$

and

$$\begin{aligned} \delta \mathbf{u}_k^f(j) &= \mathbf{u}_d(j) - \mathbf{u}_k^f(j) = \\ &= \mathbf{u}_d(j) - \mathbf{u}_{k-1}(j) - \mathbf{K}_P \mathbf{e}_{k-1}(j+1) = \\ &= \delta \mathbf{u}_{k-1}(j) - \mathbf{K}_P \mathbf{e}_{k-1}(j+1). \end{aligned} \tag{13}$$

According to the LTI system (1),

$$\begin{aligned} \mathbf{y}_k(j+1) &= \mathbf{C} \mathbf{x}_k(j+1) = \\ &= \mathbf{C} \mathbf{A}^{j+1} \mathbf{x}_k(0) + \sum_{i=0}^j \mathbf{C} \mathbf{A}^{j-i} \mathbf{B} \mathbf{u}_k(i). \end{aligned} \tag{14}$$

Using (2) and (14), we have

$$\mathbf{e}_k(j+1) = \mathbf{C} \mathbf{A}^{j+1} \delta \mathbf{x}_k(0) + \sum_{i=0}^j \mathbf{C} \mathbf{A}^{j-i} \mathbf{B} \delta \mathbf{u}_k(i). \tag{15}$$

According to Assumption 1, we can obtain

$$\mathbf{e}_k(j+1) = \sum_{i=0}^j \mathbf{C} \mathbf{A}^{j-i} \mathbf{B} \delta \mathbf{u}_k(i). \tag{16}$$

For the case  $j = 0$ , substituting (16) into (12) yields

$$\begin{aligned}\delta \mathbf{u}_k(0) &= \delta \mathbf{u}_k^f(0) - \mathbf{u}_k^a(0) = \\ &\delta \mathbf{u}_{k-1}(0) - \mathbf{K}_P \mathbf{e}_{k-1}(1) = \\ &\delta \mathbf{u}_{k-1}(0) - \mathbf{K}_P \mathbf{C} \mathbf{B} \delta \mathbf{u}_{k-1}(0) = \\ &(\mathbf{I} - \mathbf{K}_P \mathbf{C} \mathbf{B}) \delta \mathbf{u}_{k-1}(0).\end{aligned}\quad (17)$$

where  $\mathbf{I}$  is the unit matrix.

For the case  $j = 1$ , similar to (17), we can obtain

$$\begin{aligned}\delta \mathbf{u}_k(1) &= \delta \mathbf{u}_k^f(1) - \mathbf{u}_k^a(1) = \\ &\delta \mathbf{u}_{k-1}(1) - \mathbf{K}_P \mathbf{e}_{k-1}(2) - \mathbf{u}_k^a(1) = \\ &\delta \mathbf{u}_{k-1}(1) - \mathbf{K}_P \sum_{i=0}^1 \mathbf{C} \mathbf{A}^{1-i} \mathbf{B} \delta \mathbf{u}_{k-1}(i) - \\ &\sum_{i=0}^0 \mathbf{K}_L \mathbf{e}_k(i) e^{-K_E \left(\frac{i-1}{N}\right)^2} = \delta \mathbf{u}_{k-1}(1) - \\ &\mathbf{K}_P \mathbf{C} \mathbf{B} \delta \mathbf{u}_{k-1}(1) - \mathbf{K}_P \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_{k-1}(0) - \\ &\mathbf{K}_L \mathbf{e}_k(0) e^{-K_E \left(\frac{1}{N}\right)^2} = (\mathbf{I} - \mathbf{K}_P \mathbf{C} \mathbf{B}) \delta \mathbf{u}_{k-1}(1) - \\ &\mathbf{K}_P \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_{k-1}(0).\end{aligned}\quad (18)$$

For the case  $j = 2$ , similar to the above,

$$\begin{aligned}\delta \mathbf{u}_k(2) &= \delta \mathbf{u}_k^f(2) - \mathbf{u}_k^a(2) = \\ &\delta \mathbf{u}_{k-1}(2) - \mathbf{K}_P \mathbf{e}_{k-1}(3) - \mathbf{u}_k^a(2) = \\ &\delta \mathbf{u}_{k-1}(2) - \mathbf{K}_P \mathbf{e}_{k-1}(3) - \sum_{i=0}^1 \mathbf{K}_L \mathbf{e}_k(i) e^{-K_E \left(\frac{i-1}{N}\right)^2} = \\ &\delta \mathbf{u}_{k-1}(2) - \mathbf{K}_P \mathbf{e}_{k-1}(3) - \mathbf{K}_L \mathbf{e}_k(0) e^{-K_E \left(\frac{1}{N}\right)^2} - \\ &\mathbf{K}_L \mathbf{e}_k(1) = \delta \mathbf{u}_{k-1}(2) - \mathbf{K}_P \{ \mathbf{C} \mathbf{B} \delta \mathbf{u}_{k-1}(2) + \\ &\mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_{k-1}(1) + \mathbf{C} \mathbf{A}^2 \mathbf{B} \delta \mathbf{u}_{k-1}(0) \} - \\ &\mathbf{K}_L \mathbf{C} \mathbf{B} \delta \mathbf{u}_k(0) = (\mathbf{I} - \mathbf{K}_P \mathbf{C} \mathbf{B}) \delta \mathbf{u}_{k-1}(2) - \\ &\mathbf{K}_P \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_{k-1}(1) - \mathbf{K}_P \mathbf{C} \mathbf{A}^2 \mathbf{B} \delta \mathbf{u}_{k-1}(0) - \\ &\mathbf{K}_L \mathbf{C} \mathbf{B} \delta \mathbf{u}_k(0).\end{aligned}\quad (19)$$

Summarizing (19), we can obtain

$$\begin{aligned}\delta \mathbf{u}_k(2) + \mathbf{K}_L \mathbf{C} \mathbf{B} \delta \mathbf{u}_k(0) &= \\ &(\mathbf{I} - \mathbf{K}_P \mathbf{C} \mathbf{B}) \delta \mathbf{u}_{k-1}(2) - \\ &\mathbf{K}_P \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_{k-1}(1) - \mathbf{K}_P \mathbf{C} \mathbf{A}^2 \mathbf{B} \delta \mathbf{u}_{k-1}(0).\end{aligned}\quad (20)$$

For the case  $j = 3$ , similar to (19)

$$\begin{aligned}\delta \mathbf{u}_k(3) &= \delta \mathbf{u}_k^f(3) - \mathbf{u}_k^a(3) = \delta \mathbf{u}_{k-1}(3) - \\ &\mathbf{K}_P \mathbf{e}_{k-1}(4) - \mathbf{u}_k^a(3) = \delta \mathbf{u}_{k-1}(3) - \mathbf{K}_P \mathbf{e}_{k-1}(4) -\end{aligned}$$

$$\begin{aligned}\sum_{i=0}^2 \mathbf{K}_L \mathbf{e}_k(i) e^{-K_E \left(\frac{i-2}{N}\right)^2} &= \delta \mathbf{u}_{k-1}(3) - \mathbf{K}_P \mathbf{e}_{k-1}(4) - \\ &\mathbf{K}_L \mathbf{e}_k(0) e^{-K_E \left(\frac{2}{N}\right)^2} - \mathbf{K}_L \mathbf{e}_k(1) e^{-K_E \left(\frac{1}{N}\right)^2} - \mathbf{K}_L \mathbf{e}_k(2) = \\ &\delta \mathbf{u}_{k-1}(3) - \mathbf{K}_L \mathbf{C} \mathbf{B} e^{-K_E \left(\frac{1}{N}\right)^2} \delta \mathbf{u}_k(0) - \\ &\mathbf{K}_L \{ \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_k(0) + \mathbf{C} \mathbf{B} \delta \mathbf{u}_k(1) \} - \\ &\mathbf{K}_P \{ \mathbf{C} \mathbf{A}^3 \mathbf{B} \delta \mathbf{u}_{k-1}(0) + \mathbf{C} \mathbf{A}^2 \mathbf{B} \delta \mathbf{u}_{k-1}(1) + \\ &\mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_{k-1}(2) + \mathbf{C} \mathbf{B} \delta \mathbf{u}_{k-1}(3) \} = \\ &(\mathbf{I} - \mathbf{K}_P \mathbf{C} \mathbf{B}) \delta \mathbf{u}_{k-1}(3) - \mathbf{K}_L \mathbf{C} \mathbf{B} e^{-K_E \left(\frac{1}{N}\right)^2} \delta \mathbf{u}_k(0) - \\ &\mathbf{K}_L \{ \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_k(0) + \mathbf{C} \mathbf{B} \delta \mathbf{u}_k(1) \} - \\ &\mathbf{K}_P \{ \mathbf{C} \mathbf{A}^3 \mathbf{B} \delta \mathbf{u}_{k-1}(0) + \mathbf{C} \mathbf{A}^2 \mathbf{B} \delta \mathbf{u}_{k-1}(1) + \\ &\mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_{k-1}(2) \}.\end{aligned}\quad (21)$$

Coordinating (21), we can obtain

$$\begin{aligned}\delta \mathbf{u}_k(3) + (\mathbf{K}_L \mathbf{C} \mathbf{B} e^{-K_E \left(\frac{1}{N}\right)^2} + \\ &\mathbf{K}_L \mathbf{C} \mathbf{A} \mathbf{B}) \delta \mathbf{u}_k(0) + \mathbf{K}_L \mathbf{C} \mathbf{B} \delta \mathbf{u}_k(1) = \\ &(\mathbf{I} - \mathbf{K}_P \mathbf{C} \mathbf{B}) \delta \mathbf{u}_{k-1}(3) - \mathbf{K}_P \mathbf{C} \mathbf{A}^3 \mathbf{B} \delta \mathbf{u}_{k-1}(0) - \\ &\mathbf{K}_P \mathbf{C} \mathbf{A}^2 \mathbf{B} \delta \mathbf{u}_{k-1}(1) - \mathbf{K}_P \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_{k-1}(2).\end{aligned}\quad (22)$$

Proceeding in the same fashion, the following equation can be obtained for  $j = N - 1$ .

$$\begin{aligned}\delta \mathbf{u}_k(N-1) &= \delta \mathbf{u}_k^f(N-1) - \mathbf{u}_k^a(N-1) = \\ &\delta \mathbf{u}_{k-1}(N-1) - \mathbf{K}_P \mathbf{e}_{k-1}(N) - \mathbf{u}_k^a(N-1) = \\ &\delta \mathbf{u}_{k-1}(N-1) - \mathbf{K}_P \mathbf{e}_{k-1}(N) - \\ &\sum_{i=0}^{N-2} \mathbf{K}_L \mathbf{e}_k(i) e^{-K_E \left(\frac{i-N+2}{N}\right)^2} = \\ &\delta \mathbf{u}_{k-1}(N-1) - \mathbf{K}_P \mathbf{e}_{k-1}(N) - \mathbf{K}_L \mathbf{e}_k(0) e^{-K_E \left(\frac{N-2}{N}\right)^2} - \\ &\mathbf{K}_L \mathbf{e}_k(1) e^{-K_E \left(\frac{N-3}{N}\right)^2} - \mathbf{K}_L \mathbf{e}_k(2) e^{-K_E \left(\frac{N-4}{N}\right)^2} - \dots - \\ &\mathbf{K}_L \mathbf{e}_k(N-2) = (\mathbf{I} - \mathbf{K}_P \mathbf{C} \mathbf{B}) \delta \mathbf{u}_{k-1}(N-1) - \\ &\mathbf{K}_L e^{-K_E \left(\frac{N-3}{N}\right)^2} \mathbf{C} \mathbf{B} \delta \mathbf{u}_k(0) - \\ &\mathbf{K}_L e^{-K_E \left(\frac{N-4}{N}\right)^2} \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_k(0) - \\ &\mathbf{K}_L e^{-K_E \left(\frac{N-4}{N}\right)^2} \mathbf{C} \mathbf{B} \delta \mathbf{u}_k(1) - \dots - \\ &\mathbf{K}_L \mathbf{C} \mathbf{A}^{N-3} \mathbf{B} \delta \mathbf{u}_k(0) - \\ &\mathbf{K}_L \mathbf{C} \mathbf{A}^{N-4} \mathbf{B} \delta \mathbf{u}_k(1) - \mathbf{K}_L \mathbf{C} \mathbf{A}^{N-5} \mathbf{B} \delta \mathbf{u}_k(2) - \dots \\ &- \mathbf{K}_L \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_k(N-4) - \mathbf{K}_L \mathbf{C} \mathbf{B} \delta \mathbf{u}_k(N-3) - \\ &\mathbf{K}_P \mathbf{C} \mathbf{A}^{N-1} \mathbf{B} \delta \mathbf{u}_{k-1}(0) - \mathbf{K}_P \mathbf{C} \mathbf{A}^{N-2} \mathbf{B} \delta \mathbf{u}_{k-1}(1) - \\ &\mathbf{K}_P \mathbf{C} \mathbf{A}^{N-3} \mathbf{B} \delta \mathbf{u}_{k-1}(2) - \dots - \\ &\mathbf{K}_P \mathbf{C} \mathbf{A} \mathbf{B} \delta \mathbf{u}_{k-1}(N-2)\end{aligned}\quad (23)$$

Summarizing (23), we can obtain

$$\begin{aligned} & \delta \mathbf{u}_k(N-1) + \mathbf{K}_L(\mathbf{C}\mathbf{B}e^{-K_E(\frac{N-3}{N})^2} + \\ & \mathbf{C}\mathbf{A}\mathbf{B}e^{-K_E(\frac{N-4}{N})^2} + \dots + \mathbf{C}\mathbf{A}^{N-3}\mathbf{B})\delta \mathbf{u}_k(0) + \\ & (\mathbf{K}_L\mathbf{C}\mathbf{B}e^{-K_E(\frac{N-4}{N})^2} + \dots + \mathbf{K}_L\mathbf{C}\mathbf{A}^{N-4}\mathbf{B})\delta \mathbf{u}_k(1) + \\ & (\mathbf{K}_L\mathbf{C}\mathbf{B}e^{-K_E(\frac{N-5}{N})^2} + \dots + \mathbf{K}_L\mathbf{C}\mathbf{A}^{N-5}\mathbf{B})\delta \mathbf{u}_k(2) + \\ & \dots + (\mathbf{K}_L\mathbf{C}\mathbf{B}e^{-K_E(\frac{1}{N})^2} + \mathbf{K}_L\mathbf{C}\mathbf{A}\mathbf{B})\delta \mathbf{u}_k(N-4) + \\ & \mathbf{K}_L\mathbf{C}\mathbf{B}\delta \mathbf{u}_k(N-3) = (\mathbf{I} - \mathbf{K}_P\mathbf{C}\mathbf{B})\delta \mathbf{u}_{k-1}(N-1) - \\ & \mathbf{K}_P\mathbf{C}\mathbf{A}^{N-1}\mathbf{B}\delta \mathbf{u}_{k-1}(0) - \mathbf{K}_P\mathbf{C}\mathbf{A}^{N-2}\mathbf{B}\delta \mathbf{u}_{k-1}(1) - \\ & \mathbf{K}_P\mathbf{C}\mathbf{A}^{N-3}\mathbf{B}\delta \mathbf{u}_{k-1}(2) - \dots - \mathbf{K}_P\mathbf{C}\mathbf{A}\mathbf{B}\delta \mathbf{u}_{k-1}(N-2). \end{aligned} \quad (24)$$

Define

$$\mathbf{G}_0 = (\mathbf{I} - \mathbf{K}_P\mathbf{C}\mathbf{B}), \quad (25)$$

$$\mathbf{G}_m = -\mathbf{K}_P\mathbf{C}\mathbf{A}^m\mathbf{B}, \quad m \in S_{N-1} - \{0\}, \quad (26)$$

$$\tilde{\mathbf{G}}_m = \mathbf{K}_L \sum_{i=0}^m \mathbf{C}\mathbf{A}^i\mathbf{B}e^{-K_E(\frac{m-i}{N})^2}, \quad m \in S_{N-3}. \quad (27)$$

According to (17), (18), (20), (22) and (24), (12) can be further rewritten in the following composite form:

$$\tilde{\mathbf{G}}\Delta \mathbf{U}_k = \mathbf{G}\Delta \mathbf{U}_{k-1} \quad (28)$$

$\Delta \mathbf{U}_k = \mathbf{U}_d - \mathbf{U}_k = [\delta \mathbf{u}_k^T(0), \delta \mathbf{u}_k^T(1), \dots, \delta \mathbf{u}_k^T(N-1)]^T$  and

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & 0 & 0 & \dots & 0 & 0 \\ \mathbf{G}_1 & \mathbf{G}_0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{G}_{N-2} & \mathbf{G}_{N-3} & \mathbf{G}_{N-4} & \dots & \mathbf{G}_0 & 0 \\ \mathbf{G}_{N-1} & \mathbf{G}_{N-2} & \mathbf{G}_{N-3} & \dots & \mathbf{G}_1 & \mathbf{G}_0 \end{bmatrix},$$

$$\tilde{\mathbf{G}} = \begin{bmatrix} \mathbf{I} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & \dots & 0 & 0 & 0 \\ \tilde{\mathbf{G}}_0 & 0 & \mathbf{I} & \dots & 0 & 0 & 0 \\ \tilde{\mathbf{G}}_1 & \tilde{\mathbf{G}}_0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{G}}_{N-4} & \tilde{\mathbf{G}}_{N-5} & \tilde{\mathbf{G}}_{N-6} & \dots & 0 & \mathbf{I} & 0 \\ \tilde{\mathbf{G}}_{N-3} & \tilde{\mathbf{G}}_{N-4} & \tilde{\mathbf{G}}_{N-5} & \dots & \tilde{\mathbf{G}}_0 & 0 & \mathbf{I} \end{bmatrix}.$$

According to (28), we can obtain

$$\Delta \mathbf{U}_k = \tilde{\mathbf{G}}^{-1}\mathbf{G}\Delta \mathbf{U}_{k-1}. \quad (29)$$

From (29) and Lemma 2, we have

$$\|\Delta \mathbf{U}_k\| \leq \|\tilde{\mathbf{G}}^{-1}\mathbf{G}\| \|\Delta \mathbf{U}_{k-1}\| \leq \rho(\mathbf{G}) \|\Delta \mathbf{U}_{k-1}\|. \quad (30)$$

Since  $\rho(\mathbf{I} - \mathbf{K}_P\mathbf{C}\mathbf{B}) < 1$ , according to Lemma 1, then

$$\lim_{k \rightarrow \infty} \|\Delta \mathbf{U}_k\| = 0. \quad (31)$$

This implies that

$$\lim_{k \rightarrow \infty} \|\delta \mathbf{u}_k(j)\| = 0, \quad j \in S_{N-1}. \quad (32)$$

According to (16) and (32), we have

$$\lim_{k \rightarrow \infty} \|\mathbf{e}_k(j+1)\| = 0, \quad j \in S_{N-1}. \quad (33)$$

Therefore, when  $k \rightarrow \infty$ , such that  $\mathbf{y}_k(j) \rightarrow \mathbf{y}_d(j)$ ,  $j \in S_N$ .  $\square$

**Remark 4** Although the proposed associative iterative learning algorithm is similar in form to the traditional higher-order discrete learning algorithm, the learning process is completely different from the traditional higher-order iterative learning algorithm. The traditional high-order ILC is the algebraic overlay of the control information of the previous two or more trials at the corresponding time. The new iterative learning algorithm is to pre-correct the subsequent unoccurred time with the error value of the current time in the same trial. They are essentially different.

**Remark 5** When  $\mathbf{K}_L = 0$ , the iterative learning law with associative properties (5) is reduced to the traditional P-type ILC. Then the convergence condition of the control algorithm is  $\rho(\mathbf{I} - \mathbf{C}\mathbf{B}\mathbf{K}_P) < 1$ . It shows that the traditional P-type ILC transforming into an associative learning algorithm is theoretically based.

**Remark 6** The traditional P-type ILC belongs to the the feed-forward learning control. As long as the dynamic system of the controlled object and the control objective remains the same, the system will be sure to converge in the iteration period as the number of learning times  $k$  increases. The associative P-type iterative learning law (5) is a generalization of the traditional P-type. It is equivalent to increasing the real-time feedback and correcting the information, which can effectively compensate for the lack of prior knowledge of the conventional P-type ILC, and improve the convergence of the traditional P-type algorithm.

#### 4. Extension to time-varying systems

In this section, the proposed ILC scheme is extended to linear time-varying systems

$$\begin{cases} \mathbf{x}_k(j+1) = \mathbf{A}(j)\mathbf{x}_k(j) + \mathbf{B}(j)\mathbf{u}_k(j) \\ \mathbf{y}_k(j) = \mathbf{C}(j)\mathbf{x}_k(j) \end{cases} \quad (34)$$

where  $\mathbf{A}(j)$ ,  $\mathbf{B}(j)$  and  $\mathbf{C}(j)$  are time-varying matrices with appropriate dimensions and  $\mathbf{C}(j+1)\mathbf{B}(j)$  is of full

column rank. The result is summarized in the following corollary.

**Corollary 1** For the discrete-time linear time-varying system (34), an ILC algorithm with kernel association (5) is adopted, choosing the learning gain matrix  $\mathbf{K}_P$ , if the condition holds:  $\rho(\mathbf{I} - \mathbf{K}_P(j)\mathbf{C}(j+1)\mathbf{B}(j)) < 1$ . Then the output trajectory of the learning algorithm (5) converges uniformly to the desired trajectory; that is, when  $k \rightarrow \infty$ , such that  $\mathbf{y}_k(j) \rightarrow \mathbf{y}_d(j), j \in S_N$ .

**Proof** The proof can be performed similarly as Theorem 1.

Considering the time-varying systems (34), we have

$$\mathbf{y}_k(j+1) = \mathbf{C}(j+1) \left( \prod_{i=0}^j \mathbf{A}(i) \right) \mathbf{x}_k(0) + \mathbf{C}(j+1) \sum_{i=0}^j \left( \prod_{l=0}^{j-i-1} \mathbf{A}(j-l) \right) \mathbf{B}(i) \mathbf{u}_k(i). \quad (35)$$

Since a similar relationship also holds in the  $j$ th trial, it follows that

$$\mathbf{e}_k(j+1) = \mathbf{C}(j+1) \left( \prod_{i=0}^j \mathbf{A}(i) \right) \delta \mathbf{x}_k(0) + \mathbf{C}(j+1) \sum_{i=0}^j \left( \prod_{l=0}^{j-i-1} \mathbf{A}(j-l) \right) \mathbf{B}(i) \delta \mathbf{u}_k(i). \quad (36)$$

Replacing (14) and (15) in the proof of Theorem 1 with (35) and (36), we can obtain  $\rho(\mathbf{I} - \mathbf{K}_P(j)\mathbf{C}(j+1)\mathbf{B}(j)) < 1$ , it can be concluded that  $\lim_{k \rightarrow \infty} \|\delta \mathbf{u}_k(j)\| = 0, j \in S_{N-1}$ , and then  $\lim_{k \rightarrow \infty} \|\delta \mathbf{x}_k(j)\| = 0$ , so that  $\mathbf{y}_k(j)$  tends to  $\mathbf{y}_d(j)$ .  $\square$

**Remark 7** In Theorem 1 and Corollary 1, the identical initialization condition is identical.

### 5. Illustrative examples

To illustrate the effectiveness of the proposed learning algorithm, we introduce the following example, and compare the performance of the proposed new algorithm with the traditional P-type ILC.

**Case 1** Time-invariant system [34]: consider the system

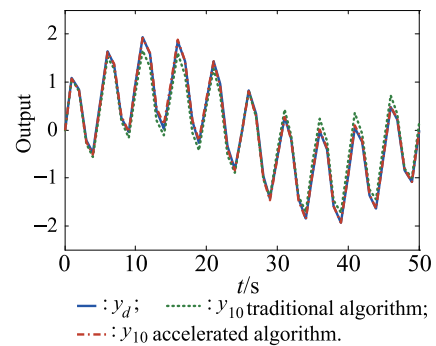
$$\begin{cases} \mathbf{x}_k(j+1) = \begin{pmatrix} 0.5 & 0 & 1 \\ 0.15 & 0.3 & 0 \\ -0.75 & 0.25 & -0.25 \end{pmatrix} \mathbf{x}_k(j) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \mathbf{u}_k(j) \\ \mathbf{y}_k(j) = (0 \ 0 \ 1) \mathbf{x}_k(j) \end{cases} \quad (37)$$

where  $\mathbf{x}_k(0) = [0, 0, 0]^T$ , the sample point  $j \in S_{50}$  is the time interval of each trial, and the desired trajectory is set to

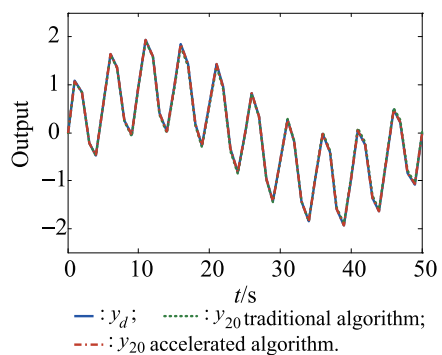
$$\mathbf{y}_d(j) = \sin(2\pi j/50) + \sin(2\pi j/5) + \sin(50\pi j). \quad (38)$$

The values of the initial iteration of the controller  $\mathbf{u}_0(j)$  are generated randomly by the rand function. The learning gain  $\mathbf{K}_P = 0.4$  is selected (the simulation example is a single input single output system, so the gain matrix is a real number). In the new learning algorithm, the convergence condition of the learning algorithm  $\rho(\mathbf{I} - \mathbf{K}_P \mathbf{C} \mathbf{B}) = 0.6 < 1$  is calculated by the exponential correction parameters  $K_E = 1$  and  $K_L = 0.1$ .

Fig. 4(a) and Fig. 4(b) respectively represent the process of tracking the desired output of the system under the 10th and the 20th trials by the traditional algorithm and the new learning algorithm. As can be seen from Fig. 4, the system output tracks the expected output completely in the whole interval with the increase of trials. The proposed association algorithm only needs 10 times, while the traditional algorithm needs 20 times.



(a) The 10th trial



(b) The 20th trial

**Fig. 4** Tracking profile of the system

To facilitate the comparative analysis, the new algorithm and the traditional P-type iterative learning algorithm are studied through the numerical simulation under the same

simulation environment. The iterative maximal tracking errors of the two algorithms are shown in Fig. 5, that is,  $\max_j |e_k|$ .

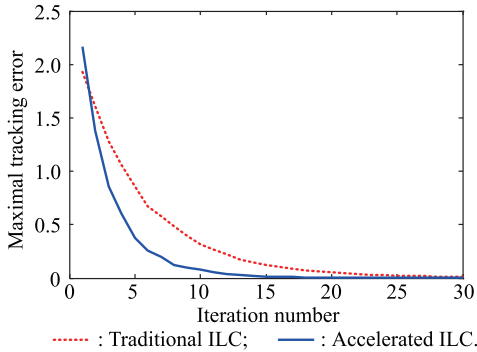


Fig. 5 Comparison of tracking errors in Case 1

It can be seen from Fig. 5 that the modified P-type control law is convergent to the error band (0.008 3) when it is iterated 10 times. The traditional control law requires 24 iterations to converge to the above error band.

**Case 2** Time-varying system [35]: in order to show the effectiveness of the new algorithm of P-type ILC, a time-varying system is given as follows:

$$\begin{cases} \mathbf{x}_k(j+1) = \begin{pmatrix} 0.2e^{-t/100} & -0.6 & 0 \\ 0 & 0.5 & \sin t \\ 0 & 0 & 0.7 \end{pmatrix} \mathbf{x}_k(j) + \\ \begin{pmatrix} 0 \\ 0.3 \sin t \\ 1 \end{pmatrix} \mathbf{u}_k(j), \\ \mathbf{y}_k(j) = (0 \ 0.1 \ 1 + 0.1 \cos t) \mathbf{x}_k(j). \end{cases} \quad (39)$$

The initial state is set as  $\mathbf{x}_k(0) = [0, 0, 0]^T$ , let the desired trajectory be

$$\mathbf{y}_d(j) = \sin(2\pi j/50) + \sin(2\pi j/5) \quad (40)$$

where  $j \in S_{50}$ . Without loss of generality, the input of the initial iteration is simply set to zero, namely,  $\mathbf{u}_0(j) = 0, j \in S_{50}$ . The learning gain  $\mathbf{K}_P = 0.4$  is selected. In the new learning algorithm, the convergence condition of the learning algorithm

$$\sup_{j \in [0, 49]} \|\mathbf{I} - \mathbf{K}_p(j)\mathbf{C}(j+1)\mathbf{B}(j)\| = 0.6 < 1$$

is calculated by the exponential correction parameters  $K_E = 1$  and  $K_L = 0.1$ . The performance of the maximal tracking error  $\max_j |e_k|$  is presented in Fig. 6, where the modified P-type control algorithm is convergent to the error band (0.018 3) when it is learning 55 times. The traditional control law requires 126 trials to converge to the above error band.

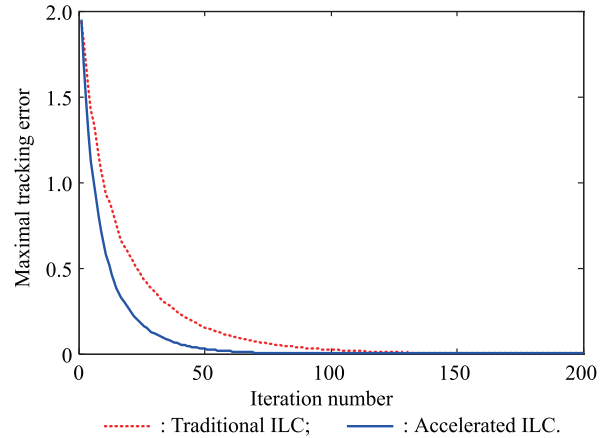


Fig. 6 Comparison of tracking errors in Case 2

In this paper, it can be presented that the learning speed of the proposed control algorithm is much better, compared with the traditional method at the same iteration.

**Remark 8** It can be seen that the convergence condition of the proposed algorithm is ultimately unrelated to the kernel function. We think that this is exactly the advantage of the algorithm in this paper, and the proposed algorithm does not lead to more constraints. For the most conventional linear discrete systems, there is no need to change their convergence conditions, the stability of the algorithm can be guaranteed by using the algorithm proposed in this paper. The appropriate association algorithms will greatly speed up the learning convergence rate, which shows that the kernel-based self-association iterative learning idea is correct and feasible.

### 6. Conclusions

We design an algorithm of P-type ILC strategy with the kernel-based auto-associative memory for a linear discrete system. During each iterative learning process, the control value of the current time is corrected. Meanwhile, the control value in the subsequent time is pre-corrected by association, which speeds up the learning process. The convergence speed of the association algorithm is illustrated by simulation results, which is related with the proportional learning gain, the learning interval and the exponential factor. In addition, how to choose the proportional learning gain and the exponential factor are also an expecting topic to discuss. Further, the investigation on these issues will come forth.

### References

- [1] ARIMOTO S, KAWAMURA S, MIYAZAKI F. Bettering operation of robotics by learning. *Journal of Robotic System*, 1984, 12(2): 123–140.
- [2] SON T D, PIPELEERS G, SWEVERS J. Robust monotonic convergent iterative learning control. *IEEE Trans. on Auto-*



- matic Control, 2017, 61(4): 1063–1068.
- [3] RUN X E, ZHAO J Y. Convergence monotonicity and speed comparison of iterative learning control algorithms for nonlinear systems. *IMA Journal of Mathematical Control and Information*, 2013, 30(4): 473–486.
  - [4] WEI Y S, LI X D. Iterative learning control for linear discrete-time systems with high relative degree under initial state vibration. *IET Control Theory and Applications*, 2016, 10(10): 1115–1126.
  - [5] CHI R H, HUANG B, WANG D W, et al. Data-driven optimal terminal iterative learning control with initial value dynamic compensation. *IET Control Theory and Applications*, 2016, 10(12): 1357–1364.
  - [6] ZHANG L, CHEN W H, LIU J M, et al. A robust adaptive iterative learning control for trajectory tracking of permanent magnet spherical actuator. *IEEE Trans. on Industrial Electronics*, 2016, 63(1): 291–301.
  - [7] LUO A, XU X Y, FANG L, et al. Feedback-feedforward PI-type iterative learning control strategy for hybrid active power filter with injection circuit. *IEEE Trans. on Industrial Electronics*, 2010, 57(11): 3767–3779.
  - [8] CHI R H, YU L, HOU Z S, et al. Data-driven terminal iterative learning control with high-order learning law for a class of non-linear discrete-time multiple-input-multiple output systems. *IET Control Theory and Applications*, 2015, 9(7): 1075–1082.
  - [9] CHI R H, HOU Z S, JIN S T, et al. Computationally efficient data-driven higher order optimal iterative learning control. *IEEE Trans. on Neural Networks and Learning Systems*, 2018, 29(12): 5971–5980.
  - [10] ROSOLIA U, BORRELLI F. Learning model predictive control for iterative tasks: a computationally efficient approach for linear system. *IEEE Trans. on Automatic Control*, 2018, 63(7): 1883–1896.
  - [11] GLUCK T, BLANK M, BUCHL D, et al. Convex constrained iterative learning control using projection: application to a smart power switch. *IEEE Trans. on Control Systems Technology*, 2018, 26(5): 1818–1825.
  - [12] JIAN Y P, HUANG D Q, LIU J B, et al. High-precision tracking of piezoelectric actuator using iterative learning control and direct inverse compensation of hysteresis. *IEEE Trans. on Industrial Electronics*, 2019, 66(1): 368–377.
  - [13] DEVASIA S. Iterative machine learning for output tracking. *IEEE Trans. on Control Systems Technology*, 2019, 27(2): 516–526.
  - [14] BU X H, HOU Z H. Adaptive iterative learning control for linear systems with binary-valued observations. *IEEE Trans. on Neural Networks and Learning Systems*, 2018, 29(1): 232–237.
  - [15] GU Q, HAO X H. Adaptive iterative learning control based on particle swarm optimization. *The Journal of Supercomputing*. <https://doi.org/10.1007/s11227-018-2566-4>.
  - [16] BU X H, CUI L Z, HOU Z S, et al. Formation control for a class of nonlinear multi-agent systems using model-free adaptive iterative learning. *International Journal of Robust and Nonlinear Control*, 2018, 28(4): 1402–1412.
  - [17] LIU T, WANG X Z, CHEN J H. Robust PID based indirect-type iterative learning control for batch processes with time-varying uncertainties. *Journal of Process Control*, 2014, 24(12): 95–106.
  - [18] MA F L, LI C D. Open-closed-loop PID-type iterative learning control for linear systems with initial state error. *Journal of Vibration and Control*, 2011, 17(12): 1791–1797.
  - [19] KOCAN O, MANECY A, POYSSOT-VASSAL C. A practical method to speed-up the experimental procedure of iterative learning controllers. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018: 6411–6416.
  - [20] WU J H, DING H. Reference adjustment for a high-acceleration and high-precision platform via a-type of iterative learning control. *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2007, 221(5): 781–789.
  - [21] WANG X, CHU B, ROGERS E. New results on higher-order iterative learning control for discrete linear systems. *Proc. of the 10th International Workshop on Multidimensional Systems*, 2017: 8070632.
  - [22] AMANN N, OWENS D H, ROGERS E. Iterative learning control using optimal feedback and feedforward actions. *International Journal of Control*, 1996, 65(2): 277–293.
  - [23] ALEXANDRIDIS A, CHONDRODIMA E, GIANNOPOULOS N, et al. A fast and efficient method for training categorical radial basis function networks. *IEEE Trans. on Neural Networks and Learning Systems*, 2017, 28(11): 2831–2836.
  - [24] CHU B, OWENS D H, FREEMAN C T. Iterative learning control with predictive trial information: convergence, robustness, and experimental verification. *IEEE Trans. on Control Systems Technology*, 2016, 24(3): 1101–1108.
  - [25] BRISTOW D A, THARAYIL M, ALLEYNE A G. A survey of iterative learning control: a learning-based method for high-performance tracking control. *IEEE Control Systems Magazine*, 2006, 26(3): 96–114.
  - [26] LOWE D G. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 1995, 7(1): 72–85.
  - [27] NIAZMARDI S, SAFARI A, HOMAYOUNI S. Similarity-based multiple kernel learning algorithms for classification of remotely sensed images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2017, 10(5): 2012–2021.
  - [28] FENG X J, WANG Y J, LIU B D, et al. Class specific dictionary learning based kernel collaborative representation for fine-grained image classification. *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, 2016: 4737–4742.
  - [29] CHU B, OWENS D H. Accelerated norm-optimal iterative learning control algorithms using successive projection. *International Journal of Control*, 2009, 82(8): 1469–1484.
  - [30] ZSIGA N, DOOREN S V, ELBERT P, et al. A new method for analysis and design of iterative learning control algorithms in the time-domain. *Control Engineering Practice*, 2016, 57: 39–49.
  - [31] MENG D Y, MOORE K L. Robust iterative learning control for non-repetitive uncertain systems. *IEEE Trans. on Automatic Control*, 2007, 62(2): 907–913.
  - [32] XU J X, YAN R. On initial conditions in iterative learning control. *IEEE Trans. on Automatic Control*, 2005, 50(9): 1349–1354.
  - [33] HORN R A, JOHNSON C R. *Matrix analysis*. Cambridge: Cambridge University Press, 1985.
  - [34] LI X F, XU J X, HUANG D Q. An iterative learning control approach for linear systems with randomly varying trial lengths. *IEEE Trans. on Automatic Control*, 2014, 59(7): 1954–1960.
  - [35] SHEN D, ZHANG W, WANG Y Q, et al. On almost sure and mean square convergence of P-type ILC under randomly varying iteration lengths. *Automatica*, 2016, 63: 359–365.

## Biographies



**LAN Tianyi** was born in 1981. He received his B.S. and M.S. degrees both in applied mathematics and Ph.D. degree in control science and engineering from Northwestern Polytechnical University in 2004, 2007 and 2017, respectively. He is currently a postdoctoral fellow with the School of Automation, Northwestern Polytechnical University. His research interests include iterative learning control, fractional order control, control theory, and multi-agent system.

E-mail: iamty1111@163.com



**LIN Hui** was born in 1957. He received his B.S. and M.S. degrees in electrical engineering and his Ph.D. degree in control science engineering from Northwestern Polytechnical University in 1982, 1985, and 1993, respectively. He is a currently a professor at the Department of Electrical Engineering, Northwestern Polytechnical University. His research interests include power electronics, electric vehicles,

motion control, and control theory.

E-mail: linhui@nwpu.edu.cn



**LI Bingqiang** was born in 1982. He received his B.E. and M.E. degrees both in electrical engineering and Ph.D. degree in control science and engineering from Northwestern Polytechnical University in 2004, 2007 and 2010, respectively. He is currently an associate professor and the dean of the Department of Electrical Engineering, Northwestern Polytechnical University, a postdoctoral fellow with Xi'an Aviation Brake Technology Co., Ltd.. His research interests include iterative learning control, fractional order control, servo control, multi-agent system and their applications.

E-mail: libingqiang@nwpu.edu.cn