Journal of Process Control 24 (2014) 64–77

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Survey on stochastic iterative learning control

Dong Shen*, Youqing Wang*

College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, PR China

ARTICLE INFO

Article history:
Received 29 September 2013
Received in revised form 1 April 2014
Accepted 14 April 2014
Available online 10 May 2014

Keywords:
Iterative learning control
Stochastic signals
Linear systems
Nonlinear systems

ABSTRACT

Iterative learning control (ILC) is suitable for systems that are able to repeatedly complete several tasks over a fixed time interval. Since it was first proposed, ILC has been further developed through extensive efforts. However, there are few related results on systems with stochastic signals, where by stochastic signal we mean one that is described by a random variable. Stochastic iterative learning control (SILC) is defined as ILC for systems that contain stochastic signals including system noises, measurement noises, random packet losses, etc. This manuscript surveys the current state of the art in SILC from the perspective of key techniques, which are divided into three parts: SILC for linear stochastic systems, SILC for nonlinear stochastic systems, and systems with other stochastic signals. In addition, three promising directions are also provided, namely stochastic ILC for point-to-point control, stochastic ILC for iteration-varying reference tracking, and decentralized/distributed coordinated stochastic ILC, respectively.

1. Introduction

In our daily lives, the ability to repeatedly work on a given task would lead to constant improvements. For example, in basketball set shooting, as the number of attempts increases, the shooter is able to increase the hit ratio since he/she may adjust the angle and speed to reduce the shooting deviation shot by shot. The basic reason for this is that we are able to learn from experiences and subsequently improve our behaviors.

This basic cognition has motivated research on iterative learning control (ILC). That is, ILC is a control method that improves its control performance by learning from previous control performance. Specifically, ILC is usually designed for systems that are able to complete some task over a fixed time interval and perform them repeatedly. In such systems, the input and output information of past cycles, as well as the tracking objective, are used to formulate the input signal for the next iteration, so that the tracking performance can be improved as the number of cycles increases to infinity. Thus, ILC has the following features: (1) the system can finish a task in a limited time, (2) the system can be reset to the same initial value, and (3) the tracking objective is iteration-invariant. The main idea of ILC is shown in Fig. 1.

In Fig. 1, y_d denotes the reference trajectory. Based on the input of the kth iteration, u_k, as well as the tracking error $e_k = y_d - y_k$, the input u_{k+1} for the next iteration, i.e., the $(k+1)$th iteration, is constructed. Meanwhile, the input u_{k+1} is also stored into the memory for the $(k+2)$th iteration. Thus, a closed loop feedback is formed along the iteration index.

By comparing ILC with our daily lives, we find that the previous information on inputs and outputs of the plant corresponds to the experiences faced in our daily lives. Persons usually decide on a strategy for a given task based on previous experiences, while the strategy here is equivalent to the input signal of ILC. Note that the previous experiences would help us to improve our behavior; thus, it is reasonable to believe that information on the previous operation may help to improve the control performance to some extent.

The major advantage of ILC is that the design of control law only requires the tracking references and input/output signals. In other words, not much information about the plant is required and it may even be completely unknown. However, the algorithm is simple and effective.

It is important to note that ILC adjusts the control along the iteration index rather than the time index, which is the main difference with other control methods such as proportional-integral-derivative (PID) control. PID control is a widely used feedback control. However, for iteration type systems, PID generates the same tracking error during each iteration since no previous information is used, while ILC reduces the tracking error iteration by iteration. Additionally, ILC differs from adaptive control, which also learns from previous operation information. Adaptive control aims to adjust the parameter of a given controller, while ILC aims to construct the input signal directly.

* Corresponding authors. Tel.: +86 10 64419847.
E-mail addresses: shendong@mail.buct.edu.cn (D. Shen), wang.youqing@ieee.org (Y. Wang).

http://dx.doi.org/10.1016/j.jprocont.2014.04.013
0959-1524/

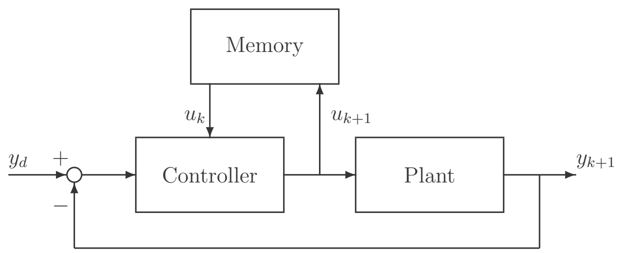

Fig. 1. Framework of ILC.

The concept of ILC may be traced back to a paper published in 1978 by Uchiyama [1]. However, this paper failed to attract widespread attention as it was written in Japanese. Three papers that were published in 1984 [2–4] resulted in further the research on ILC. Subsequently, large amounts of literature have been published on various related issues, such as research monographs [5–9], survey papers [10–12], and special issues of academic journals [13–16]. ILC has recently become an important branch of intelligent control, and its use is widespread in many practical applications such as robotics [17–20], hard disk drives [21,22], and industrial processes [23,24].

1.1. Background of ILC

In this subsection, basic formulations of ILC are given, followed by some traditional convergence results. Consider the following discrete-time linear time-invariant system

$$\begin{aligned} x(t+1,k) &= Ax(t,k)+Bu(t,k) \\ y(t,k) &= Cx(t,k) \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, and $y \in \mathbb{R}^q$ denote the system state, input, and output, respectively. Matrices A, B, and C are system matrices with appropriate dimensions. t denotes an arbitrary time instance in an operation iteration, $t=0, 1, \ldots, N$, where N is the length of the operation iteration. For simplicity, $t \in [0, N]$ is used in the following. $k=0, 1, 2, \ldots$ denote different iterations.

Because it is required that a given tracking task should be repeated, the initial state needs to be reset at each iteration. The following is a basic reset condition, which has been used in many publications.

$$x(0,k) = x_0, \quad \forall k \tag{2}$$

The reference trajectory is denoted by $y(t, d)$, $t \in [0, N]$. With regard to the reset condition, it is usually required that $y(0, d) = y_0 \triangleq Cx_0$. The control purpose of ILC is to design a proper update law for the input $u(t, k)$, so that the corresponding output $y(t, k)$ can track $y(t, d)$ as closely as possible. To this end, for any t in $[0, N]$, we define the tracking error as

$$e(t,k) = y(t,d) - y(t,k) \tag{3}$$

Then the update law is a function of $u(t, k)$ and $e(t, k)$ to generate $u(t, k+1)$, whose general form is as follows

$$u(t,k+1) = h(u(\cdot,k), \ldots, u(\cdot,0), e(\cdot,k), \ldots, e(\cdot,0)) \tag{4}$$

When the above relationship depends only on the last iteration, it is called a first-order ILC update law; otherwise, it is called a high-order ILC update law. Generally, considering the simplicity of the algorithm, most update laws are first-order laws, i.e.,

$$u(t,k+1) = h(u(\cdot,k), e(\cdot,k)) \tag{5}$$

Additionally, the update law is usually linear. The simplest update law is as follows

$$u(t,k+1) = u(t,k) + Ke(t+1,k) \tag{6}$$

where K is the learning gain matrix, which is also the designed parameter. In (6), $u(t, k)$ is the input of the current iteration, while $Ke(t+1, k)$ is the innovation term. The update law (6) is called a P-type ILC update law. If the innovation term is replaced by $K(e(t+1, k) - e(t, k))$, the update law is a D-type one.

For system (1) and update law (6), a basic convergence result is that K satisfies

$$\|I - CBK\| < 1 \tag{7}$$

Then, one has $\|e(t,k)\| \underset{k\to\infty}{\to} 0$, where $\|\cdot\|$ denotes the operator norm.

From this result, one can deduce that the design of K needs no information regarding the system matrix A, but for the coupling matrix CB. This illustrates the advantage of ILC from the perspective where ILC has little dependence on the system information. Thus, ILC can handle tracking problems that have more uncertainties.

Remark 1. From the formulation of ILC, one can see that the model takes the classic features of a 2D system. Many researchers have made contributions from this point of view, and developed a 2D system-based approach, which is one of the principal techniques for ILC design and analysis.

Note that the operation length is limited by N, and is then repeated multiple times. Thus, one could use the so-called lifting technique, which implies lifting all of the inputs and outputs as supervectors,

$$U_k = [u^T(0,k), u^T(1,k), \ldots, u^T(N-1,k)]^T \tag{8}$$

$$Y_k = [y^T(1,k), u^T(2,k), \ldots, y^T(N,k)]^T \tag{9}$$

Denote

$$G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \cdots & \cdots & CB \end{bmatrix} \tag{10}$$

then one has

$$Y_k = GU_k + \mathbf{d} \tag{11}$$

where

$$\mathbf{d} = [(Cx_0)^T, (CAx_0)^T, \ldots, (CA^N x_0)^T]^T \tag{12}$$

Similar to (8) and (9), define

$$\begin{aligned} Y_d &= [y^T(1,d), u^T(2,d), \ldots, y^T(N,d)]^T \\ E_k &= (e^T(1,k), e^T(2,k), \ldots, e^T(N,k))^T \end{aligned}$$

then it leads to

$$U_{k+1} = U_k + \mathbf{K}E_k \tag{13}$$

where $\mathbf{K} = diag\{K, K, \ldots, K\}$. By simple calculation, one has

$$\begin{aligned} E_{k+1} &= Y_d - Y_{k+1} = Y_d - GU_{k+1} - \mathbf{d} = Y_d - GU_k - G\mathbf{K}E_k - \mathbf{d} \\ &= E_k - G\mathbf{K}E_k = (I - G\mathbf{K})E_k \end{aligned}$$

Therefore, we obtain a condition that is sufficient to guarantee the convergence of ILC (7). Actually, the lifting technique not only helps us to obtain the convergence condition, but it also provides us with an intrinsic understanding of ILC. In the lifted model (11), the evolutionary process of an operation iteration has been integrated into G, where the relationship between adjacent iterations is highlighted. That is, the lifted model (11) is only along the k-axis, while the t-axis has no more influence.

Remark 2. Note that the focus of ILC is how to improve the tracking performance iteratively along the iteration index, as one can see from the design of the update law and lifted model (11). Therefore, it would not cause additional difficulties when the system is extended from the linear time-invariant case to a linear time-varying case. This is because for any fixed time, the updating along the iteration index is a time-invariant case.

It is usually assumed that the reference trajectory $y(t, d)$ is realizable. That is, there exist an appropriate initial state x_0 and input $u(t, d)$ such that the expression (1) still holds with k replaced by d. In other words, $Y_d = GU_d + \mathbf{d}$, where U_d is defined in a manner similar to (8). Then, the discussions that the system output converges to the reference trajectory, $\lim_{k\to\infty} Y_k = Y_d$, becomes one where the system input converges to the objective input, $\lim_{k\to\infty} U_k = U_d$. For the system with stochastic noises, this approach is more convenient.

Remark 3. One may be interested in the case where the reference trajectory is not realizable, i.e., there is no control input producing the reference trajectory; thus, entirely accurate tracking is impossible. Then, the design objective of the ILC algorithm is no longer to guarantee asymptotically accurate tracking, but to converge the nearest trajectory to the given reference. Consequently, the tracking problem has become an optimization problem. On the other hand, from the point of view of practical applications, the reference trajectory is usually realizable; thus, the assumption is not rigorous.

All the above discusses the formulation of ILC for linear systems. On the other hand, ILC for nonlinear systems also gains much attention [26–28]. Due to the inherent nonlinear property, it is more difficult to handle the related ILC problem for nonlinear systems than the linear ones. Most reported studies assume that the nonlinear functions satisfy global Lipschitz condition (GLC), which is an essential restriction for their applications. Thus how to relax this condition is a quite interesting direction in nonlinear ILC study.

The available literature covers numerous topics, including the update law design, relaxing the initial reset condition, robustness, optimality, transient performance, tracking iteration-varying references, and practical applications. Here, we will not explain them in detail. For reference, [11] presented detailed categorizations of publications from 1998 to 2004, which were separated into two parts, i.e., theoretical developments and applications. In [25], the authors surveyed related literatures on norm optimization.

1.2. Stochastic ILC

In this subsection, the definition of stochastic iterative learning control (SILC) will be given. Throughout the paper, SILC is used to describe ILC for the system containing some stochastic signals, where the stochastic signal is described by random variables in probability theory. For example, system noises in the operation process, measurement noises, and random packet losses are all included. The key features of a stochastic signal are the uncertainty, randomness, and usually have no upper bound.

Specifically, the related literature on SILC may be separated into the following categories:

- linear systems with system disturbances and/or measurement noises;
- nonlinear systems with system disturbances and/or measurement noises;
- systems with other kind of stochastic signals, such as random packet losses and random asynchronism.

Currently, thorough explanations of ILC for deterministic systems have been contributed by the control theorists and engineers. Many significant results have been published and applied to actual systems. However, the interferences and noises are inevitable during practical operations. Thus, disturbance rejection is an important issue for ILC research. In most studies, it is assumed that the disturbances are bounded by a specific limit. Then, it was shown that the algorithms would converge into a small range formulated as a function of this limit. More specifically, we denote the disturbance by w_k and assume $\|w_k\| \le \epsilon$, where ϵ is a proper constant. It is generally shown that ILC algorithms would guarantee the tracking error e_k satisfying $\|e_k\| \le L(\epsilon)$, $L(\epsilon) \underset{\epsilon\to 0}{\to} 0$ [29,30]. It is obvious that this approach fails in the SILC case.

Therefore, more attention should be given to dealing with random variables when considering SILC. To this end, some novel methods have been proposed, which will be surveyed below. However, this is only the first step to SILC, and more work needs to be done for this ongoing topic. Moreover, although many papers have been published on SILC, none of them have yet provided an in-depth survey.

The rest of the paper is arranged as follows. Section 2 discusses the case of SILC for linear stochastic systems, while the case of SILC for nonlinear stochastic systems is reviewed in Section 3. Both sections are arranged based on key approaches. Then, comments on SILC for systems with other stochastic signals are presented in Section 4. In Section 5, the promising directions and outlooks are provided. Concluding remarks are given in Section 6.

2. SILC for linear stochastic systems

2.1. Kalman filtering based approach

Consider the following linear stochastic systems

$$\begin{aligned} x(t+1, k) &= A(t)x(t, k) + B(t)u(t, k) + w(t, k) \\ y(t, k) &= C(t)x(t, k) + v(t, k) \end{aligned} \tag{14}$$

where $A(t)$, $B(t)$, and $C(t)$ are time-varying system matrices, $w(t, k)$ and $v(t, k)$ are the system noise and measurement noise, respectively. Note that the major difference between (14) and (1) is the introduction of the additive noise terms. Thus the traditional norm contraction based analysis technique and condition (7) are no longer suitable.

Saab contributed a series of papers on SILC for a system (14) [31–34]. In [31], the author proposed the first SILC algorithm for a discrete-time linear system, and progressed in subsequent publications [32–34]. The update law designed in [31] is

$$u(t, k+1) = u(t, k) + K(t, k)[e(t+1, k) - e(t, k)] \tag{15}$$

where $K(t, k)$ is the learning gain matrix.

Remark 4. Actually, the system formulation in [31] has a slight difference from (14) in that there is an additional time-independent noise term $v_b(k)$ in the measurement equation. One can observe that by using the D-type learning update law, the noise term $v_b(k)$ is directly canceled. Thus, this term has no influence on the design and analysis, which is why we omit it here.

To analyze the convergence property of the algorithm (15), some hypotheses are required. It is first assumed that $y(t, d)$ is realizable, the reason for which has been explained in Section 1.1 and Remark 3. That is, there exist input $u(t, d)$ and initial state $x(0, d) = x_0$ such that

$$\begin{aligned} x(t+1, d) &= A(t)x(t, d) + B(t)u(t, d) \\ y(t, d) &= C(t)x(t, d) \end{aligned} \tag{16}$$

Besides, the input/output coupling matrix $C(t+1)B(t)$ is assumed to be full column rank. To keep the notations concise without causing confusion, it is simplified as $C^+B = C(t+1)B(t)$ in the rest of the paper. The pre-fix notation δ denotes the distance between the objective value and the actual value, i.e., $\delta x(t,k) \triangleq x(t,d) - x(t,k)$, $\delta u(t,k) = u(t,d) - u(t,k)$. We denote $\mathbb{E}$ as a mathematical expectation. Then, the conditions on noises, initial states, and initial input are specified as follows [31].

Both $w(t,k)$ and $v(t,k)$ are assumed as Gaussian white noise with zero mean, where the covariance matrix $Q_t = \mathbb{E}[w(t,k)w(t,k)^T]$ is positive semidefinite while $R_t = \mathbb{E}[v(t,k)v(t,k)^T]$ is positive definite. Moreover, $w(t,k)$ is uncorrelated with $v(s,l)$, $\forall t, s \in [0,N]$, $k, l \in \mathbb{R}$. The initial state is assumed to be a random variable, which is also a Gaussian white noise with zero mean and covariance $P_{x,0} = \mathbb{E}[\delta x(0,k)x(0,k)^T]$ being positive semidefinite. Additionally, $\delta x(0,k)$ is uncorrelated with any noise. Moreover, the initial input satisfies the requirement that the initial input error $\delta u(t,0)$ is zero-mean white noise with symmetric positive definite covariance $\mathbb{E}[\delta u(t,0)\delta u(t,0)^T] = P_{u,0}$.

Remark 5. The above conditions on noises, initial states and initial input are mainly required by the Kalman filtering approach. Here, we make more comments on the assumption of the initial input. As we observe, it actually means that the initial input $u(t,0)$ should be normally distributed around the objective input $u(t,d)$. The author also pointed out that a simple case satisfying the above condition is that $\delta u(t,0) = 0, \forall t$ [31]. However, as a matter of fact, it is quite hard to satisfy this condition when little system information is known in advance. The essential idea of ILC is to achieve sufficiently good input signals by iterative learning for any initial input, which contradicts this assumption. On the other hand, if we can guarantee that $\delta u(t,0) = 0, \forall t$, it equivalently declares that the initial input is exactly the objective input $u(t,d)$. Then, there is no obvious need to update the input. The subsequent contributions [32–35,49] have not removed this limitation, and it remains an open issue.

According to system (14) and update law (15) with the above assumptions, [31] obtained the computational algorithm for $K(t,k)$ by minimizing the trace of the input error covariance matrix, and further proved the convergence in the mean square. To show this, the following 2D-Roessor model was first derived as

$$X^+ = \Phi X + \Gamma Z \tag{17}$$

where

$$X^+ = \begin{bmatrix} \delta u(t,k+1) \\ \delta x(t+1,k) \end{bmatrix} \quad X = \begin{bmatrix} \delta u(t,k) \\ \delta x(t,k) \end{bmatrix}$$
$$Z = \begin{bmatrix} w(t,k) \\ v(t+1,k) - v(t,k) \end{bmatrix}$$
$$\Phi = \begin{bmatrix} I - K(t,k)C^+B & K(t,k)[C(t) - C(t+1)A(t)] \\ B(t) & A(t) \end{bmatrix}$$
$$\Gamma = \begin{bmatrix} K(t,k)C(t+1) & K(t,k) \\ -I & 0 \end{bmatrix}$$

Then, set the derivative of the trace of the covariance matrix $P^+ \triangleq \mathbb{E}(X^+X^{+^T})$ with respect to $K(t,k)$ to zero. The recursive algorithm for the computation of $K(t,k)$ is thus obtained as follows

$$K(t,k) = P_{u,k}\Xi^T(\Xi P_{u,k}\Xi^T + \Lambda_{D,k})^{-1} \tag{18}$$

$$P_{u,k+1} = (I - K(t,k)\Xi)P_{u,k} \tag{19}$$

where $\Xi \triangleq C(t+1)B(t)$, $\Lambda_{D,k} \triangleq (C(t) - C(t+1)A(t))P_{x,k}(C(t) - C(t+1)A(t))^T + C(t+1)Q_tC(t+1)^T + R_t + R_{t+1}$, $P_{x,t} = \mathbb{E}[\delta x(t,k)\delta x(t,k)^T]$, $P_{u,k} = \mathbb{E}[\delta u(t,k)\delta u(t,k)^T]$.

Remark 6. In [31], it was noted that the learning gain matrix $K(t,k)$ is used to update the input $u(t,k+1)$ at the $(k+1)$th iteration; thus, $K(t,k)$ has no impact on the state $x(t+1,k)$. Consequently, the derivative of the trace of P^+ with respect to $K(t,k)$ is equivalent to the derivative of the trace of $P_{u,k+1}$ with respect to $K(t,k)$. The latter idea is more concise and intuitive, and was thus adopted in the subsequent results [33,34,49,35,36].

The following theorem is the major convergence theorem of [31].

Theorem 1. [31] *For system (14) and update law (15), (18) and (19), if $C(t+1)B(t)$ is of full column rank, then $\forall k$, t, there exist a consistent norm $\|\cdot\|$ such that $\|I - K(t,h)C(t+1)B(t)\| < 1$. Consequently, $\|P_{u,k+1}\| < \|P_{u,k}\|$. Furthermore, $P_{u,k} \to 0$, and $K(t,k) \to 0$ uniformly on $[0,N]$ as $k \to \infty$.*

Now, we briefly summarize the Kalman filtering based approach, which is also used in [32–34,49]. First, build a 2D-Roessor model with input error $\delta u(t,k)$ and state error $\delta x(t,k)$. Next, calculate the derivative of the optimization objective, which is usually the input error covariance $P_{u,k}$, with respect to the learning gain matrix $K(t,k)$, and set it to zero to obtain the recursive algorithm. Finally, prove the mean square convergence based on the irrelevance of noises and (semi)-positive definiteness of covariance matrices.

Theorem 1 has established the framework of a Kalman filtering based SILC approach. However, a lot of information about the system is required to ensure the correct running of algorithms (15), (18) and (19). It is seen that system matrices $A(t)$, $B(t)$, $C(t)$, noise covariances Q_t, R_t, and state error covariance $P_{x,t}$ are all required (see the expression of $\Lambda_{D,k}$). Some scenarios in which this requirement is relaxed are presented in [32–34].

In [32], the term $P_{x,t}$ in (18) was removed; then, the algorithms become

$$\widetilde{K}(t,k) = \widetilde{P}_{u,k}\Xi^T(\Xi\widetilde{P}_{u,k}\Xi^T + \widetilde{\Lambda}_D)^{-1} \tag{20}$$

$$\widetilde{P}_{u,k+1} = (I - \widetilde{K}(t,k)\Xi)\widetilde{P}_{u,k} \tag{21}$$

where $\widetilde{\Lambda}_D = C(t+1)Q_tC(t+1)^T + R_t + R_{t+1}$. In [32], it was proven that the learning gain matrix $\widetilde{K}(t,h)$ given by (20) and (21) for update law (15) still worked. That is, a theorem similar to Theorem 1 holds if $C(t+1)B(t)$ is of full column rank.

It should be especially noted that $\widetilde{P}_{u,k}$ is constructed only for the recursion of the algorithms, and not the real input error covariance. We denote the real input error covariance based on algorithms (15), (20) and (21) as $\overline{P}_{u,k}$. An interesting result is that the convergence of $\widetilde{P}_{u,k}$ and $\overline{P}_{u,k}$ is equivalent, i.e., $\widetilde{P}_{u,k} \to 0 \Leftrightarrow \overline{P}_{u,k} \to 0$. Besides, the rate of convergence is inversely proportional to the iteration number k [32].

So far, it is believed that the removal of the state error covariance does not affect the convergence. However, the input of the modified algorithms is not optimal; thus, (18) and (19) are called optimal algorithms and (20) and (21) are called suboptimal algorithms. Note that only D-type algorithms are considered in [31,32]. Then, it is natural to ask whether similar results hold for P-type algorithms. A positive answer is provided by [33].

Consider the following P-type update law

$$u(t,k+1) = u(t,k) + K(t,k)e(t+1,k) \tag{22}$$

Then, by performing exactly the same steps as in [31,32], the recursive computational algorithms are derived. The optimal algorithms are

$$K(t,k) = P_{u,k}\Xi^T(\Xi P_{u,k}\Xi^T + \Lambda_{P,k})^{-1} \tag{23}$$

$$P_{u,k+1} = (I - K(t,k)\Xi)P_{u,k} \tag{24}$$

where $\Lambda_{P,k} = C(t+1)A(t)P_{x,k}(C(t+1)A(t))^T + C(t+1)Q_tC(t+1)^T + R_{t+1}$, while the suboptimal algorithms are

$$\widetilde{K}(t,k) = \widetilde{P}_{u,k}\Xi^T(\Xi\widetilde{P}_{u,k}\Xi^T + \widetilde{\Lambda}_P)^{-1} \tag{25}$$

$$\widetilde{P}_{u,k+1} = (I - \widetilde{K}(t,k)\Xi)\widetilde{P}_{u,k} \quad (26)$$

where $\widetilde{\Lambda}_P = C(t+1)Q_t C(t+1)^T + R_{t+1}$. It has been shown in [33] that the results of Theorem 1 still hold.

Because there are both P-type and D-type algorithms, another natural question is whether there is a relationship between these two kinds of algorithms. As a matter of fact, note that the essential distinction among algorithms (18)–(26) is the selection of the bounded positive matrix Λ. This matrix should be $\Lambda_{D,k}$, $\widetilde{\Lambda}_D$, $\Lambda_{P,k}$, or $\widetilde{\Lambda}_P$, respectively. As long as this matrix is bounded, symmetrical, and positive, all of the above algorithms have the same convergence properties when $C(t+1)B(t)$ is of full column rank. In other words, regardless of the algorithm that is selected, the input error covariance matrix will converge to 0, with the rate being inversely proportional to the iteration number.

At the end of this subsection, some related results are listed. The first one is that a preliminary attempt was made for the case where $C(t+1)B(t)$ is of full row rank and the contribution is very limited [34]. Saab [35] showed that no forgetting factor is needed for output tracking of system (14) based on the P-type update law. Saab [36] showed that the first-order update law is the best one among high-order update laws in the sense of minimizing the trace of the input error covariance.

2.2. Stochastic approximation based approach

As in Section 2.1, fruitful achievements have been realized in the Kalman filtering based approach, but all of the convergence results are in the mean square sense. Chen first published the almost sure convergence result for SILC in [37], where the system model is also given by (14). The purpose of the control is to minimize the following asymptotically average output tracking errors

$$\limsup_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} \|y(t+1,k) - y(t+1,d)\|^2 = \min \quad a.s. \quad (27)$$

To make a comparison, the conditions of noises and initial states in [37] are first introduced.

The noises $w(t,k)$ and $v(t,k)$ are independent zero-mean random variables with finite moments of order $2+\delta$, where $\delta>0$. These covariance matrices are defined as before, but are unknown. The initial state is mutually independent of all noises $w(t+1,k)$, $v(t,k)$, and $\mathbb{E}x(0,k) = x_0$, $\mathbb{E}\|x(0,k)\|^{2+\delta} < \infty$ with unknown covariance matrix.

It is easy to see that the requirements on covariances are removed in [37] by making comparisons with [31].

The reference trajectory $y(t,d)$ is realizable, i.e., Eq. (16) holds. When $C(t+1)B(t)$ is of full column rank, the objective input $u(t,d)$ is solved $\forall i = 0, 1, \ldots, N-1$

$$u(i,d) = [(C(i+1)B(i))^T (C(i+1)B(i))]^{-1} (C(i+1)B(i))^T \times (y(i+1,d) - C(i+1)A(i)x(i,d)) \quad (28)$$

It was proven that any input sequence $\{u(t,k)\}$ converging to $u(t,d)$ is optimal according to (27). That is, (27) is minimized [37].

The system matrices $A(t)$, $B(t)$, $C(t)$, as well as noise covariances Q_t, R_t are all assumed to be unknown, thus one has to estimate the correction direction when constructing the control algorithm, which may further generate an input sequence $\{u(t,k)\}$ converging to $u(t,d)$ in [37]. Then a novel SILC algorithm is provided based on stochastic approximation. Here we rewrite the algorithm in the following form. To this end, a vector sequence $\{\Delta(t,k)\}$ that is independent of $w(t,k)$ and $v(t,k)$ is needed, where $\Delta(t,k) = [\Delta_1(t,k), \ldots, \Delta_p(t,k)]^T$ is a p-dimensional and all components $\Delta_j(t,k)$ are mutually independent identically distributed (i.i.d.) random variables, $\forall k = 1, 2, \ldots, t \in [0, N-1], j = 1, \ldots, p$, such that

$$|\Delta_j(t,k)| < m, \ \left|\frac{1}{\Delta_j(t,k)}\right| < n, \ \mathbb{E}\frac{1}{\Delta_j(t,k)} = 0 \quad (29)$$

where m and n are positive constants. We denote

$$\overline{\Delta}(t,k) = \left[\frac{1}{\Delta_1(t,k)}, \ldots, \frac{1}{\Delta_p(t,k)}\right]^T \quad (30)$$

Let $\{a_k\}$, $\{c_k\}$, $\{M_k\}$ be sequences of real numbers satisfying the following conditions

$$a_k > 0, \quad a_k \underset{k\to\infty}{\to} 0, \quad \sum_{k=0}^{\infty} a_k = \infty \quad (31)$$

$$c_k > 0, \quad c_k \underset{k\to\infty}{\to} 0, \quad \sum_{k=0}^{\infty} \left(\frac{a_k}{c_k}\right)^{1+(\delta/2)} < \infty \quad (32)$$

$$M_k > 0, \quad M_{k+1} > M_k, \quad M_k \underset{k\to\infty}{\to} \infty \quad (33)$$

where δ is defined in the noise assumptions. The initial input $u(t,0)$, $t \in [0, N]$ is arbitrarily given. The algorithm is given according to the odd iteration number and even iteration number, respectively. Specifically,

$$u(t, 2k+1) = u(t, 2k) + c_k \Delta(t,k) \quad (34)$$

and

$$\overline{u}(t, 2(k+1)) = u(t, 2k) - a_k \frac{\overline{\Delta}(t,k)}{c_k} (\|e(t+1, 2k+1)\|^2 - \|e(t+1, 2k)\|^2) \quad (35)$$

$$u(t, 2(k+1)) = \overline{u}(t, 2(k+1)) \cdot I_{[\|\overline{u}(t,2(k+1))\| \le M_{\sigma_k(t)}]} \quad (36)$$

$$\sigma_k(t) = \sum_{l=1}^{k-1} I_{[\|\overline{u}(t,2(l+1))\| > M_{\sigma_l(t)}]}, \quad \sigma_0(t) = 0 \quad (37)$$

where $I_{[\cdot]}$ is an indicator function meaning that it equals 1 if the condition indicated in the bracket is fulfilled, and 0 if the condition does not hold.

Note that the algorithm (34)–(37) is actually a stochastic approximation algorithm with expanding truncations [38]. The main idea of (34)–(37) is that in the odd iteration, a small stochastic disturbance is added to the input. Then, in the subsequent even iteration, the update gradient is estimated by using the two tracking errors from the two previous adjacent iterations. Here, a_k is the update step length. Unlike the Kalman filtering based approach, where the learning gain matrix $K(t,k)$ is directly calculated, the stochastic approximation based approach first gives an estimation of the gradient, then ensures the convergence with the help of an iteration-varying step size. The main convergence result is as follows [37].

Theorem 2. [37] *For system (14) and update law (34)–(37), if $C(t+1)B(t)$ is of full column rank, then the input sequence $\{u(t,k)\}$ converges to the objective input $u(t,d)$ with probability 1, and is thus optimal according to index (27).*

The significance of Theorem 2 lies in bringing a stochastic approximation based approach for the construction of SILC. The approach requires no information on system matrices and noise covariances, which to an extent agrees with the basic advantage of ILC. Because little information is known, the algorithm has to estimate its updating gradient by the stochastic difference method, which has been extensively studied in [38–41].

Table 1
Comparison of Kalman filtering based approach and stochastic approximation based approach.

	Kalman filtering based approach	Stochastic approximation based approach
Index	Input error covariance	Asymptotically averaged quadratic tracking error
System information	Known prior	Unknown prior
Stochastic noises	Gaussian white noise with zero-mean	Independent zero-mean random variables with finite moments
Learning gain matrix estimation method	Taking a derivation of the index and setting it to zero	Fixing the steps size and estimating updating gradient based stochastic difference
Convergence	Mean square sense convergence	Almost sure convergence

The differences between the Kalman filtering based approach [31] and stochastic approximation based approach [37] are given in Table 1. From Table 1, we could see that one major difference between these two indices in [31,37]: the former is an index of expectations of random variables and while the latter is an index of actual random variables. According to the law of large numbers, the limit of asymptotically averaged quadratic tracking error is the covariance, which links these two indices. Another major difference is the estimation method of learning gain matrix, where [31] carries out it through directly computing the error covariance, which thus leads to mean square convergence, and while [37] proposes a gradient-estimation-based approach.

In addition, [42] also considered system (14) and index (27), and constructed an SILC algorithm based on stochastic approximation. It differs from [37] in that [42] is based on Robbins–Monro (RM) algorithm, while [37] is based on Kiefer–Wolfowitz (KW) algorithm. The KW algorithm uses the stochastic difference to estimate the gradient, while the RM algorithm removes this term [38]. Thus, without prior information of the system, it is required to estimate the control direction.

For clarification, consider the SISO case of (14) [42]. Then, the input/output coupling matrix $C(t+1)B(t)$ is actually a real number; thus, the control direction is the sign of $C(t+1)B(t)$. To obtain the correct control direction, [42] provided the following algorithm

$$u_{k+1}(t) = u_k(t) + a_k S(p_k(t+1)) e_k(t+1), \tag{38}$$

$$q_{k+1}(t+1) = q_k(t+1) + \frac{1}{k+1}(e_{k+1}^2(t+1) - q_k(t+1)), \tag{39}$$

$$p_{k+1}(t+1) = \max\{p_k(t+1), q_{k+1}(t+1)\}, \tag{40}$$

where $q_0(t) = 0$, $a_0 = 1$ and $a_k = \frac{1}{k}$, $k \geq 1$. $S(\cdot)$ is the direction switch function with a value of either +1 or −1, for which $p_k(t+1)$ is the argument. The latter reveals the effect of the tracking performance on direction switching. For this algorithm, it has been shown that the input sequence $\{u(t, k)\}$ converges to $u(t, d)$ with probability 1.

Remark 7. Comparing (34)–(37) and (38)–(40), one could find that the former includes the indicator function, which means that the algorithm may be pulled back to 0 and restarted, while the latter does not include any indicator function, which means that the input is continuously updated. In practical applications, resetting the input to 0 may negatively affect the production. However, algorithms (34)–(37) can deal with more general systems, as explained in Section 3.

2.3. Other approaches

There are other approaches that deal with linear stochastic systems other than the Kalman filtering based approach and stochastic approximation based approach. The first one is the statistics based approach given by [43–46], which in this case means that the research focus is on the mathematical expectation of random variables and their variances. In the paper [43], the authors established the following super-vector model

$$Y_k = GU_k + \varepsilon_k \tag{41}$$

where ε_k is the lifted noise vector defined as (8) and (9). The assumptions regarding the noise are as follows.

The noise vector ε_k is white noise with $\mathbb{E}\varepsilon_k = 0$, $\mathbb{E}[\varepsilon_k \varepsilon_k^T] = V$, $\mathbb{E}[\varepsilon_k \varepsilon_{k+i}^T] = 0$, $i \neq 0$, where V is positive definite.

The update law is

$$U_{k+1} = U_k + LE_k \tag{42}$$

where L is the learning gain matrix. By defining $G_e = I - GL$, it is obvious that

$$E_k = G_e E_{k-1} + \varepsilon_{k-1} - \varepsilon_k \tag{43}$$

When proving the convergence, mathematical expectations are first taken of both sides of Eq. (43). Then, it implies that the mathematical expectation of E_k converges to zero if the spectral norm of G_e satisfies $\rho(G_e) < 1$. Besides, $\mathrm{Var}[E_k]$ is also shown to converge to some constant matrix. However, by using expectations, this approach first removes the basic difficulty of stochastic systems. This is because as a mathematical expectation is taken, the model becomes deterministic. On the other hand, the expectation that the tracking error converges to zero is not always as good as expected, since it may result in a large tracking error if the covariance limit is large.

This approach was also used in [44], where an SISO linear discrete time system following the formation of (41) was taken into account, and the noises are modeled as zero mean weakly stationary sequences. Both the mathematical expectation and variance of the tracking error were formulated there according to general ILC update law. Then, in-depth analyses were given for the update law with forgetting factor, update law with decreasing learning gain, and update law with filter.

The disturbance rejection problem was considered in [45,46], where the stochastic noise was assumed to be a white stationary process. In [45], the author showed that iteration variant learning filters could asymptotically give the controlled signal zero error and zero variance, while in [46], the authors formulated an error equation of the covariance matrix of the controlled signal error.

The frequency analysis based approach was also introduced for stochastic systems. Bristow [47] was a pioneer publication on this topic, where the system was modeled as an SISO linear time invariant system with stationary noises. The tradeoff between the convergence rate and converged error spectrum was proposed based on a closed loop 2D model and frequency domain analysis. In the simulation results, colored noises were also tested.

The stochastic adaptive control based approach is also believed to be an effective approach, although no related papers limited to stochastic systems have been found by the authors. However, stochastic adaptive control has been studied extensively [48]. The stochastic adaptive control based approach means that the system parameters are first iteratively identified, and then the control signals are generated. This approach would lead to important achievements for SILC. However, note that there still exists a gap between stochastic adaptive control and SILC, since the former focuses on the time domain while the latter focuses on the iteration domain.

3. SILC for nonlinear stochastic systems

Unlike SILC for linear stochastic systems which has multi-aspect discussions, the research on SILC for nonlinear stochastic systems is limited. One possible reason is that there are limited powerful techniques when stochastic noise and nonlinear function are combined. It is hoped that additional efforts will be made to further develop this interesting issue.

3.1. Kalman filtering based approach

The Kalman filtering based approach is adopted in [49] for a class of affine nonlinear system modeled as

$$\begin{aligned} x(t+1,k) &= f(x(t,k)) + B(x(t,k))u(t,k) \\ y(t,k) &= C(t)x(t,k) + v(t,k) \end{aligned} \tag{44}$$

where $f(\cdot)$ is a vector function defined on $\mathbb{R}^n$. It can be seen that the system equation is time-invariant, that is, both $f(\cdot)$ and $B(\cdot)$ are independent of the time instance t. Note that for affine nonlinear systems, the relationship between the output and input is linear.

Basic assumptions: The coupling matrix $G(x)=C(t+1)B(x)$ is assumed to be of either full column rank or full row rank. The reference trajectory $y(t, d)$ is realizable, i.e., for appropriate $x(0, d)$, there exists suitable $u(t, d)$ such that the noiseless system can exactly generate $y(t, d)$. The nonlinear functions $f(\cdot)$ are allowed to grow as fast as a polynomial with an arbitrary degree, but both $B(\cdot)$ and $G(\cdot)$ should be bounded. Assumptions regarding the noise, initial state, and initial input are similar to [31].

Remark 8. In [49], the nonlinear functions $f(\cdot)$ are allowed to grow as fast as a polynomial. This condition relaxes the requirements of nonlinear systems. As a common condition in ILC for nonlinear systems, the global Lipschitz condition is usually required. However, if the nonlinear function grows as fast as a polynomial with arbitrary degree, then only the local Lipschitz condition is satisfied.

In [49], $y(t, k)$ is called the measurement output, which is only used for the update law, while the tracking output is $C(t)x(t,k)$. Thus, $e(t, k)=y(t, d)-y(t, k)$ denotes the measured tracking error in [49], while $\delta\psi(t,k)\triangleq C(t)[x(t, d)-x(t, k)]$ is the tracking error. Therefore, the control purpose of [49] is to produce an input sequence such that the tracking error converges to zero and all trajectories are bounded.

A P-type update law was taken, where the learning gain matrix $K(t, k)$ was derived by a similar but more detailed derivation of [33,34]. The algorithm was shown to guarantee the input error covariance matrix converging to zero with a rate that is inversely proportional to the iteration number.

However, the major contribution of [49] is not providing a single SILC algorithm, but proposing a framework for the design of an update law with necessary and sufficient conditions for the boundedness of trajectories and output tracking. Here, as an example, we consider only the case of full column rank. We denote $G_k\triangleq G(x(t, k))$, $K_k\triangleq K(t, k)$, and $\Phi_k\triangleq I-K_kG_k$. The following theorem is for the boundedness of trajectories.

Theorem 3. [49] *For system* (44) *and update law* (22), *∀k, t, the boundedness of all trajectories is guaranteed if and only if* $\exists c_\Sigma>0$, *such that*

$$\left\| \sum_{i=0}^{k-1}\left[\prod_{j=1}^{k-1-i}\Phi_{k-j}\right]K_iK_i^T\left[\prod_{j=1}^{k-1-i}\Phi_{k-j}\right]^T\right\| \le c_\Sigma \tag{45}$$

Based on this theorem, to make trajectories bounded, the learning gain matrix may be designed to satisfy $\exists\, c_5>0$ and $c_6>0$, such that $\left\|\prod_{i=0}^{k-1}\Phi_{k-i-1}\right\|\le c_5$ and $\|K_k\|\le(c_6/k)$.

Remark 9. It should be noted that the meaning of boundedness in [49] is from the perspective of the mathematical expectation. Specifically, we say that $x(t, k)$ is bounded if $\mathbb{E}[\delta x(t,k)\delta x(t,k)^T]$ is bounded. Strictly, the boundedness of the covariance does not imply that the random variable is bounded.

The following theorem is for output tracking.

Theorem 4. [49] *For system* (44) *and update law* (22), *if* $\lim_{k\to\infty}\delta x(0,k)=0$, *then the state error, input error, and output tracking error all converge to zero in the mean-square sense as* $k\to\infty$, *if and only if*

$$\lim_{k\to\infty}K_k=0,\quad \lim_{k\to\infty}\left[\prod_{i=0}^{k-1}\Phi_{k-i-1}\right]=0$$
$$\lim_{k\to\infty}\sum_{i=0}^{k-1}\left[\prod_{j=1}^{k-1-i}\Phi_{k-j}\right]K_iK_i^T\left[\prod_{j=1}^{k-1-i}\Phi_{k-j}\right]^T=0$$

Based on this theorem, to ensure that the input error converges to zero in the mean square sense with a rate that is inversely proportional to the iteration number, the learning gain matrix should be designed to satisfy $\exists c_7>0$ and $c_8>0$ such that $\|K_k\|\le(c_7/k)$ and $\|\prod_{j=1}^{k-1-i}\Phi_{k-j}\|\le c_8(i/k)$. Furthermore, to ensure that the state error and tracking error converges, we require that $\|\delta x(0, k)\|^2\le(c_9/k)$, $c_9>0$.

Finally, it should be noted that because of the introduction of nonlinear functions $f(\cdot)$ and $B(\cdot)$, the system noise term $w(t, k)$ is removed, as seen from (44) and (14). Further attempts can be made on this topic. Besides, a stricter condition is required where the initial state error converges to zero asymptotically to guarantee the zero convergence of the output tracking error. In other words, the initial state should be asymptotically reset to the required position as the iteration number increases. In one sense, since the initial state cannot be influenced by the input signal, the output therefore cannot exactly track the reference if the initial state is not exactly reset. It remains an open issue to determine whether the initial state can be asymptotically reset accurately using some learning strategy when random noises exist.

3.2. Stochastic approximation based approach

An affine nonlinear system was also considered in [50], where the stochastic approximation based approach is applied. The model is expressed as follows

$$\begin{aligned} x(t+1,k) &= f(t,x(t,k)) + B(t,x(t,k))u(t,k) + w(t+1,k) \\ y(t,k) &= C(t)x(t,k) + v(t,k) \end{aligned} \tag{46}$$

where $f(t, x)$ and $B(t, x)$ are time-varying functions. However, as explained in Remark 2, the characteristics of time variation do not play an important role. The purpose of the control is to minimize the index (27).

Basic assumptions: The functions $f(t, x)$ and $B(t, x)$ are continuous in x, and are bounded by some polynomial function. In other words, there exist c_{10}, c_{11}, l such that $\|f(t, x)\|+\|B(t, x)\|\le c_{10}\|x\|^l+c_{11}$. The coupling matrix $C(t+1)B(t, x)$ is of full column rank, $\forall x\in\mathbb{R}^n$. Assumptions regarding the noises are similar to those in [37]. Besides, it is further required that $w(t, k)$ and $x(0, k)$ satisfy $\mathbb{E}\|w(t, k)\|^r<\infty$, $\mathbb{E}\|x(0, k)\|^r<\infty$, $\forall r\in\mathbb{Z}^+$. All random variables are i.i.d. along the iteration index k, ∀t.

Denote $P(t,x)\triangleq B^T(t,x)C^T(t+1)C(t+1)B(t,x)$. Then, $P(t,x)$ is positive definite. For the reference $y(t, d)$, let us first give the optimal

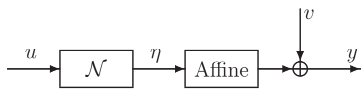

Fig. 2. Block diagram of system with hard nonlinearities at the input.

Fig. 3. Block diagram of Hammerstein–Wiener system.

input $u^0(t)$ minimizing (27) inductively. Let $x^0(0,k) \equiv x(0,k)$, then $u^0(t)$ and $x^0(t,k)$ are well-defined consecutively as follows

$$\begin{aligned} u^0(t) = &-[\mathbb{E}P(t, x^0(t,k))]^{-1} \\ &\times\{\mathbb{E}[B^T(t, x^0(t,k))C^T(t+1)f(t, x^0(t,k))] \\ &-\mathbb{E}[B^T(t, x^0(t,k))]C^T(t+1)y(t+1,d)\} \end{aligned} \tag{47}$$

$$\begin{aligned} x^0(t+1,k) = &f(t, x^0(t,k)) + B(t, x^0(t,k))u^0(t) \\ &+w(t+1,k) \end{aligned} \tag{48}$$

It was first proven that if the input sequence $\{u(t,k)\}$ satisfies $u^0(t) - u(t,k) \underset{k\to\infty}{\to} 0$, $\forall t$, then $\{u(t,k)\}$ is also optimal, i.e., the index (27) is minimized. The SILC update law was also defined by (34)–(37), where all parameters were given in (29)–(33). The generated input sequence was shown to converge to $u^0(t)$ with probability 1.

There is a linear relationship between the output and input in both [49,50]. This further motivates us to discuss the case where the relationship is nonlinear. A class of such kind of SISO system was studied in [51], as shown in Fig. 2. Here, the input did not enter the system directly, but first passed static hard nonlinear functions such as dead-zone, saturation, and pre-load, which are described as follows

$$\begin{aligned} &x(t+1,k) = f(t, x(t,k)) + b(t, x(t,k))\eta(t,k) \\ &\eta(t,k) = \mathcal{N}(u(t,k)) \\ &y(t,k) = c(t)x(t,k) + v(t,k) \end{aligned} \tag{49}$$

where $\eta(t,k)$ is an unknown intermediate signal denoting what the static nonlinear function generates when the input enters. $\mathcal{N}$ denotes the nonlinear function including dead-zone, saturation, and pre-load. These three kinds of hard nonlinearities are quite common in practical engineering applications. The existence of such nonlinearities causes the output to depend on the input in a nonlinear way. The purpose of the control is also to minimize the index (27).

Basic assumptions: The reference trajectory $y(t,d)$ is realizable. The coupling matrix $c(t+1)b(t,x)$ is unknown but has a known nonzero sign, which is denoted by $\text{sgn}(c^+b_k(t))$. The nonlinear functions $f(t,x)$ and $b(t,x)$ with respect to x are bounded by some polynomial. The measurement noise $v(t,k)$ is mutually independent along the iteration k, and has zero mean and finite second-order moment. The initial state is asymptotically reset exactly, i.e., $\delta x(0,k) \underset{k\to\infty}{\to} 0$.

Note that all of the nonlinearities mentioned are nonsmoothing; that is, their derivative could not be obtained at some points and may be discontinuous. Thus, the stochastic difference based gradient estimation method could not be used in this case. The SILC algorithm proposed in [51] is

$$\begin{aligned} u(t,k+1) = &[u(t,k) + a_k \text{sgn}(c^+b_k(t))e(t+1,k)] \\ &\times I_{[|u(t,k)+a_k\,\text{sgn}(c^+b_k(t))e(t+1,k)| \le M_{\sigma_k(t)}]} \end{aligned} \tag{50}$$

$$\sigma_k(t) = \sum_{i=1}^{k-1} I_{[|u(t,i)+a_i\,\text{sgn}(c^+b_i(t))e(t+1,i)| > M_{\sigma_i(t)}]} \tag{51}$$

$$\sigma_0(t) = 0 \tag{52}$$

where a_k and M_k are given in (31) and (33), respectively. This algorithm is actually the RM algorithm with expanding truncations [38]. It is apparent that the above algorithm does not include the hard nonlinearity $\mathcal{N}(\cdot)$. In other words, the paper [51] constructed a unified algorithm for three kinds of nonlinearities. The input sequence was also proved to be bounded and optimal.

In this study, all of the parameters of three kinds of hard nonlinearities, as well as the system parameters are unknown. However, $\text{sgn}(c^+b_k(t))$, which is equivalent to the control direction for an SISO system, needs to be known. Besides, only the measurement noise is considered, which means that there is no coupling between the stochastic noise and nonlinear function. Thus, the RM algorithm can generate a suitable input sequence.

Then, the authors extended the above three special nonlinearities to a general nonlinear function in [52], where the SILC problem for the Hammerstein–Wiener system was handled. The so-called Hammerstein–Wiener system is a cascading system consisting of a static nonlinearity followed by a linear stochastic system and then a static nonlinearity, as shown in Fig. 3, where SLS denotes the stochastic linear system. Thus, the output depends on the input in a nonlinear way. The specific expression is

$$\begin{aligned} &\eta(t,k) = f(t, u(t,k)) \\ &x(t+1,k) = A(t)x(t,k) + B(t)\eta(t,k) + \varepsilon(t+1,k) \\ &z(t,k) = C(t)x(t,k) + \epsilon(t,k) \\ &y(t,k) = g(t, z(t,k)) + v(t,k) \end{aligned} \tag{53}$$

where $\eta(t,k)$ and $z(t,k)$ are unknown intermediate signals. Nonlinear functions $f(t,\cdot): \mathbb{R}^p \to \mathbb{R}^p$, $g(t,\cdot): \mathbb{R}^q \to \mathbb{R}^q$, $\forall t \in [0,T]$, denote static nonlinearities at the input side and output side, respectively. $\varepsilon(t,k)$ and $\epsilon(t,k)$ are system noises, while $v(t,k)$ is the measurement noise. The purpose of the control is to minimize the index (27). The following are needed.

Basic assumptions: System noises $\varepsilon(t,k)$, $\epsilon(t,k)$ are with zero mean and finite moments of any integer order. The measurement noise $v(t,k)$ has zero mean and finite second order moment. The initial state is independent of all the noise signals with $\mathbb{E}x(0,k) = x_0$ and $\mathbb{E}\|x(0,k)\|^m < \infty, \forall m$. All of the random variables are i.i.d. along iteration k. For any $t \in [0,T]$, both $f(t,\cdot)$ and $g(t,\cdot)$ are continuous differential. Moreover, $\|g(t,x)\|$ is bounded by some polynomial with respect to x.

Before focusing on the convergence result, we first make more explanations of the existence of the optimal input. We denote the objective input as $u(t,d)$, which is the inverse solution of $y(t,d)$ in the noiseless case. However, when stochastic noises exist, the index (27) cannot be minimized even when the input is $u(t,d)$ at each iteration. The reason for the difference in the optimal input in the noiseless case as opposed to the noise case is that system noises are coupled with the nonlinearities, which makes the expectation of the output move further from the reference. Thus, according to the index (27), the optimal input should be expressed based on mathematical expectation. The existence condition on the optimal input was given in [52].

Specifically, note that the noises $\varepsilon(t,k)$, $\epsilon(t,k)$ in the linear subsystem of (53) are additive. Denote the noiseless subsystem as

$$\begin{aligned} &x'(t+1,k) = A(t)x'(t,k) + B(t)\eta(t,k) \\ &z'(t,k) = C(t)x'(t,k) \end{aligned} \tag{54}$$

Table 2
Literature on SILC for linear and nonlinear systems.

	Linear case	Nonlinear case
KF	[31–36]	[49]
SA	[37,42]	[50–52]
St	[43–46]	
Fr	[47]	

Notations: KF, Kalman filtering based approach; SA, stochastic approximation based approach; St, statistics based approach; Fr, frequency analysis based approach.

where $x'(0, k) = x(0, d)$; therefore, the gap between $z(t, k)$ and $z'(t, k)$ is a random variable denoted by $\omega(t, k)$. In other words,

$$y(t,k) = g(t, z'(t,k) + \omega(t,k)) + v(t,k) \tag{55}$$

Let $P_t(x) \triangleq \mathbb{E}\|y(t,d) - g(t, x+\omega_t)\|^2, \forall t$, where w_t is i.i.d. with $\omega(t, k)$. Thus, the intermediate signal $z'(t)$ minimizing (27) is the argument minimizing $P_t(x)$. It was shown in [52] that the input sequence produced by (34)-(37) converges to an optimal input with probability 1.

3.3. Other approaches

Because of the basic difficulty of stochastic nonlinear systems, there are few publications regarding SILC related to it. Here, we note only that the stochastic adaptive control based method may be a potential approach for parameterized stochastic nonlinear systems. One specific case is that the nonlinear system is linearly parameterized, and then the major idea is updating the parameter estimation and generating control signals. On the other hand, a general nonlinear system can be approximated by neural networks, fuzzy functions, wavelet functions, etc. Therefore, the stochastic adaptive control based approach is also believed to be helpful for general stochastic nonlinear systems. At the end of this section, the publications on SILC for linear and nonlinear stochastic systems are classified in Table 2.

4. SILC for systems with other stochastic signals

4.1. Random packet losses

In addition to developments in the field of network and communication technology, the application of networked control systems has recently become more widespread. In this kind of control system the sensors, actuators, and controllers are connected by a network, which enhances the flexibility and reliability. In the meantime, there is random packet loss because of network congestion, linkage interrupts, transmission errors, and/or other factors, therefore reducing the system performance. There has been preliminary exploration of SILC for systems with random packet losses. Fig. 4 is a simple illustration of a networked system with a packet loss channel.

Here, we first give a model of random packet losses problem. Considering system (1), the update law is (6) if there is no packet loss. Now, assume that the packet loss occurs at the output side, i.e., output signal $y(t, k)$, is lost when transmitting back to the controller. We denote the new tracking error by $\gamma e(t, k)$, where $\gamma \in \{0, 1\}$ is a Bernoulli distributed random variable representing packet loses or not. In other words, if $\gamma = 0$, the packet loss occurs; otherwise, if $\gamma = 1$ the packet is successfully transmitted. We let $\overline{\gamma} = \mathbb{E}\gamma$.

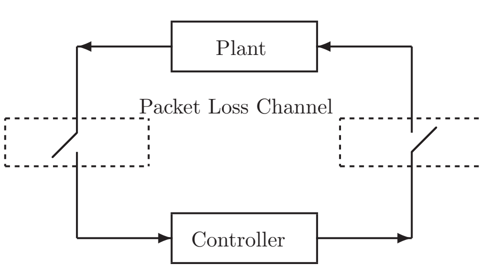

Fig. 4. Block diagram of networked system with packet loss channel.

Ahn et al. considered a linear discrete-time system with random packet losses at the output and/or input sides in [53–55] based on the Kalman filtering approach. Therefore, all convergence results are in the mean-square sense.

In [53], the system was a deterministic one (1), and the update law took the same form as follows

$$u(t,k+1) = u(t,k) + K(t,k)\gamma e(t+1,k) \tag{56}$$

By performing similar steps to those in the derivation of [31,33,34], we obtained the computational algorithm for $K(t, k)$. We proved that the input error would converge to 0 in the mean-square sense as long as $\overline{\gamma} \neq 0$. That is, the convergence was still guaranteed by applying (56) provided that the packets were not 100% lost.

Note that in [53], the packet loss is modeled according to the whole vector output, i.e. the output is either completely lost or completely transmitted. However, in many applications, only some data of the output vector is lost. The case of partially lost data was discussed in [54], where the system was a time-variant case of (14), i.e., $A(t) \equiv A$, $B(t) \equiv B$, $C(t) \equiv C$. For the q-dimensional output $y(t, k) \in \mathbb{R}^q$, we may denote it as $y(t,k) = [y^1(t,k), y^2(t,k), \ldots, y^q(t,k)]^T$. Then the update law (56) was rewritten as:

$$u(t,k+1) = u(t,k) + K(t,k)\Gamma e(t+1,k) \tag{57}$$

where Γ is the diagonal matrix

$$\Gamma = diag[\gamma_i] = \begin{pmatrix} \gamma_1 & 0 & \cdots & 0 \\ 0 & \gamma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_q \end{pmatrix} \tag{58}$$

with $\gamma_i \in \{0, 1\}$, $i = 1, \ldots, p$ being mutual independent Bernoulli distributed random variables. We denote $\overline{\gamma_i} = \mathbb{E}\gamma_i$, $i = 1, \ldots, p$. In this case, the input error covariance was shown to converge to zero if $\overline{\Gamma}CB$ is of full rank, where the latter implies that $\overline{\Gamma}$ is nonsingular, or $\overline{\gamma_i} \neq 0, \forall i$.

In both [53,54], the packet loss is assumed to occur at the output side, while it may also occur at the input side. This was studied in [55], where the system was modeled in supervector form

$$\begin{aligned} U_{k+1} &= U_k + \Upsilon \mathcal{N}_k E_k \\ Y_k &= G\mathcal{M}_k U_k \end{aligned} \tag{59}$$

where $\mathcal{N}_k$ and $\mathcal{M}_k$ denote the packet loss factor at the output side and input side, respectively. The diagonal elements of $\mathcal{N}_k$ and $\mathcal{M}_k$ are Bernoulli distributed random variables while other elements are zero. Υ is the learning gain matrix, while G is the same as in (10). Then the authors provided a sufficient condition to guarantee the mean-square convergence of the tracking error based on system matrices and expectations of $\mathcal{N}_k$ and $\mathcal{M}_k$.

Bu et al. studied the problem from another point of view, i.e., a statistics based approach, in [56–58]. In [56], the system was the time-variant SISO case of supervector form (11) with $\mathbf{d} = 0$. Accordingly, the update law was the scale case of (56). By taking the mathematical expectation of the iterative error equation, similar to [43], a traditional stability condition was given. The corresponding nonlinear system was dealt with in [57,58], where the system was modeled as follows

$$x(t+1,k) = f(x(t,k)) + b(x(t,k))u(t,k)$$
$$y(t,k) = g(x(t,k)) + d(x(t,k))u(t,k) \tag{60}$$

It was assumed that all functions satisfy the global Lipschitz condition and the reference $y(t, d)$ is realizable. Then, based on the traditional contract mapping method, a stability condition that is similar to [56] was given on the basis of $d(x(t, k))$. This is because there is a control term in the measurement equation, which plays a major role in the generation of a suitable output. Two comments should be noted. The first is that when the system is nonlinear, the lifting technique fails to construct the supervector form, which implies that the techniques of [56] could not be applied to [57,58]. The second is that as the contract mapping method is introduced, mathematical expectations are taken of the stochastic iterative inequalities rather than stochastic iterative equations, which may restrict the conditions. From our observations, the main idea of Bu's work is to transform a stochastic process into a deterministic process by taking mathematical expectations, and then giving conditions based on the latter one. Moreover, [59] provided an H_∞ iterative learning controller for a class of discrete-time systems with data dropouts. With help of the super-vector formulation, the original system is reformulated as a linear discrete-time stochastic system in the iteration domain and hence the H_∞ performance problem in the iteration domain is defined and discussed.

In[60], the author presented an averaging technique to handle packet losses and the communication delay problem. The system was modeled as (1) with a disturbance term in the system equation. The packet loss was modeled by a Bernoulli distributed random variable similar to [53], while the communication delay was modeled as

$$y_c(t,k) = \pi y(t,k) + (1-\pi)y(t-1,k) \tag{61}$$

where π was also a Bernoulli distributed random variable valued from $\{0, 1\}$. Please note that this communication delay will be referred to in Section 4.3 as it is a random delay. Another thing to note here is that the packet loss is not independent of the communication delay since communication only happens when packet loss does not occur. However, it is easy to prove $\mathbb{E}[\gamma\pi] = \overline{\gamma\pi}$. With a multidimensional case, the notations are $\Gamma = diag(\gamma_i)$ and $\Pi = diag(\pi_i)$.

To compensate for packet losses and communication delays, the averaging technique along the iteration axis was used in [60]. The key idea is to reduce disturbances using historical data. To this end, an iteration-average operator $\mathbb{A}$ was defined for a sequence $f_0(\cdot)$, $f_1(\cdot), \ldots, f_i(\cdot)$,

$$\mathbb{A}\{f_i(\cdot)\} = \frac{1}{i+1}\sum_{j=0}^{i} f_j(\cdot) \tag{62}$$

Then, the update law was given based on this operator

$$u(t,k+1) = \mathbb{A}[u(t,k)] + \frac{k+2}{k+1}K\Gamma\sum_{j=0}^{k}\widetilde{e}(t+1,j) \tag{63}$$

Here, $\widetilde{e}(t+1,j) = y(t+1,d) - y_c(t+1,j) = \Pi e(t+1,j) + (I-\Pi)e(t,j) + (I-\Pi)\delta(t)$, $\delta(t) = y(t+1, d) - y(t, d)$. In addition, K is the learning gain matrix. Using the contract mapping method, it was shown that the expectation of the averaged tracking error $\mathbb{E}[\mathbb{A}[e(t,k)]]$ converges into a bound if $\|I - K\overline{\Gamma\Pi}CB\| \le \rho < 1$ is satisfied for the above update law. Roughly speaking, with the help of averaging historical data, we may obtain a better performance. This requires that further steps be taken.

Although research into ILC research for random packet losses has just commenced, there have already been fruitful results in the field of networked control systems, where the packet losses problem is an important issue. However, there are differences between ILC and traditional networked control. For example, it was found that there is a critical value that ensures the stability of networked systems with packet losses[61], while for ILC, this critical value may not be needed [53]. Thus, we believe that many interesting topics remaining to be found and resolved.

4.2. Random asynchronism

In practical engineering, many systems are typical large scale systems that consist of many interconnected subsystems, such as power systems and petrochemistry processes. In such kinds of systems, it is hard for each subsystem to learn all information of the whole system when generating control signals; hence decentralized control is more suitable. Due to potentially different working efficiency values among subsystems, the control action may not be updated for all subsystems at an iteration step. This was the motivation for the research on ILC with the random asynchronism problem. A recent publication attempted to address this topic [62].

Let the large scale system be composed of n subsystems, of which the ith one is described by

$$x_i(t+1,k) = f_i(t, x(t,k)) + b_i(t, x(t,k))u_i(t,k)$$
$$y_i(t,k) = c_i(t)x_i(t,k) + v_i(t,k) \tag{64}$$

where $u_i(t,k) \in \mathbb{R}$, $y_i(t,k) \in \mathbb{R}$, $x_i(t,k) \in \mathbb{R}^{n_i}$. The reference for each subsystem is denoted by $y_i(t, d)$, $i = 1, \ldots, n$. The purpose of the control is to minimize the index (27) for all subsystems.

The assumptions are similar to [51], but more attention should be given to asynchronous updates among subsystems. The asynchronism here is assumed to be random and unpredictable, and is described by a random set S_k. Specifically, by using $S_k \subset \{1, \ldots, n\}$, we refer to the set of those subsystems that are updated at the kth iteration, and denote by $\tau(i, k)$ the number of control updates that occurred up-to and including the kth iteration in subsystem i: $\tau(i,k) \triangleq \sum_{j=1}^{k} I_{[i\in S_j]}$. The update law for the ith subsystem is

$$u_i(t,k+1) = u_i(t,k) + a(\tau(i,k))I_{[i\in S_k]}e_i(t+1,k) \tag{65}$$

where $a(k)$ is the learning step that satisfies the following conditions

$$a(k) > 0, \quad \sum_{k=0}^{\infty} a(k) = \infty, \quad \sum_{k=0}^{\infty} a(k)^2 < \infty \tag{66}$$

$$a(j) = a(k)(1 + O(a(k))) \tag{67}$$

Obviously, $a(k) = (1/(k+1))$ is a suitable choice.

Remark 10. The set S_k is random and characterizes the asynchronous nature of updating. During the period, where the whole system has been updated k times, the ith subsystem has actually been updated only $\tau(i, k) \le k$ times.

The algorithm (65) is based on asynchronous stochastic approximation. The almost surely convergence result of [51] is based on the following condition on random asynchronism.

Random asynchronism assumption: There is a number L such that $\forall k, i$

$$\tau(i,k+L) - \tau(i,k) > 0 \tag{68}$$

Generally speaking, this condition means that control in any subsystem should be updated at least once during L successive updates of the whole system. Thus it is a frequency requirement related to updates, which requires only the existence of L without knowing its value in advance. Work remains to be done to determine how to relax this condition.

4.3. Random delays

Time delays may reduce the performance of systems, as reported in numerous studies such as [63,64]. As explained in the introduction section, the essence of ILC is to adjust the input using the input and output information of previous iterations; thus, the repetitive information along the iteration axis would not significantly affect ILC. One of the main advantages of ILC is its reduced dependence on system information; thus, unknown but fixed time delays would have no impact on control performance since it could be regarded as a part of the information of the system. This intuition is verified in [65], where a class of affine nonlinear systems with time delays are studied. Under unknown but fixed time delays, a simple ILC algorithm (50) can ensure the almost surely convergence of the input sequence.

Generally, the proposed ILC algorithm may track the desired output without any effects caused by state delays. This may indicate that the state delays do not significantly affect ILC. However, further explorations should be made regarding the basic influence of time delays on ILC performance.

On the other hand, we aim to determine the random time delay that would make a significant effect on system performance. There is yet no complete or explicit reply to this question. It is should be noted that the communication delay of [60] may be viewed as a kind of random delay, see (61). Besides, [55] also provided a preliminary attempt for random delays. Specifically, in model (59), if diagonal elements of $\mathcal{N}_k$ are binary random variables and other elements are all 0, then it denotes the random packet losses problem. If $\mathcal{N}_k$ is a general stochastic matrix, then it also indicates the random time delays. The conclusions of [55] are made based on the general stochastic matrix, where only the mean square convergence is shown.

As observed, a lot of questions remain unanswered. For example, the effect by which the random time delays affect the control performance and how to eliminate or reduce this influence by designing appropriate algorithms.

5. Promising directions and outlooks

In this section, three promising directions are surveyed, which are not fully revealed yet even for standard ILC. However, it should be pointed out that most of problems for standard ILC may also be considered for SILC, such as initial shifting conditions, iteration-varying uncertainties, coordination control, and so on.

5.1. Point-to-point SILC

The standard ILC usually requires the system output to track a desired objective over the whole time interval. However, in more practical applications, only some desired points may be required to realize accurate tracking, while the others are not considered. For example, a basketball player repeatedly shoots from a fixed position. As we could see, the focus is on whether the basketball hits the target, rather than whether the basketball follows some predetermined trajectory. This kind of iterative tracking problem is called point-to-point ILC. It is obvious that this problem can be settled if one simply tracks any set of full trajectories, including the desired outputs, but it may waste the control energy and lose the freedom of the control design.

If only the terminal point is considered for tracking, it is called terminal iterative learning control (TILC), which is a special case of point-to-point ILC. In [66], TILC was applied to rapid thermal processing chemical vapor deposition in the wafer fabrication industry. The authors considered a discrete-time linear system and parameterized the control input as a linear combination of properly chosen basis functions. Thus, the TILC, was designed according to the updating of combination coefficients. This idea was also given in [67,68], where the latter considered a continuous-time system and specified shifted Legendre orthogonal polynomials as the basis functions. Another application study was provided in [69,70], where the TILC problem regarding plastic sheet surface temperature control in thermoforming machine was presented. The corresponding high-order case was given in [71]. The recent paper [72], where the initial state learning for the final state control was presented for a general motion system, proposed a novel geometric method in the convergence proof of four simple algorithms. Then based on [72], the initial learning control was applied to a train station stop in [73], where the terminal stop position error was used to correct the input profile. Moreover, [74] constructed estimation and control algorithms based on input/output data for a linear discrete time-varying MIMO system. The algorithms there were obtained by optimizing a quadratic index, which was also used in [75]. It should be noted that [75] proposed optimal update rules for a constant input, while [74] considered continuous inputs.

As a general case, point-to-point motion control was also discussed in many papers. The general point-to-point ILC may be divided into two types according to the required input and output positions. One type is that the whole operation range could be separated into two adjacent intervals, namely the actuation interval and observation interval. The other one is that input and output positions are interspersed with each other. The residual vibration suppression problem is a typical case of the former type point-to-point problem studied in [76–78], where the standard ILC technique was used. Thus, they may fail to utilize the extra freedom of the unrequested pass points for better performance. For the latter type, [79,80] solved the multiple point-to-point tracking problem by iteratively updating the reference between trials instead of the input profile, which made good use of the freedom of trajectory, and showed a novel way for resolving the point-to-point control problem. The difference between [79] and [80] is that the former uses the discrete Fourier transform technique in the frequency domain, while the latter uses an interpolation technique in the time domain. Another promising method used to deal with the point-to-point problem is to directly update the control signal on the basis of specified tracking data and performance index [75,80]. In [75], the authors proposed ILC algorithms for the multiple pass points tracking problem, where the performance index is a kind of quadratic form of only pass points and inputs rather than the whole trajectory. In [80], the control input was linearly parameterized, where the basis functions were constructed by system information matrices. The performance index there is a general cost function of the error, control efforts, and a variation of the control amount. In addition, a recent paper [81] published a norm-optimal ILC solution to the continuous-time point-to-point problem with a comparison between the experimental performance and theoretical results.

For a MIMO system, the required pass point in the above literature is the whole output vector for a given time, while in practice we may only claim some components of the output vector to satisfy the constraints. For example, let the output be a three dimensional vector denoting a spatial location. Then, for a given time, we may only give a constraint for the altitude while letting the other two remain free. This kind of point-to-point tracking problem was studied in [82] for linear systems and in [83] for nonlinear systems, respectively. Freeman and Tan [82] also provided extensive analysis on the gradient descent-based ILC and Newton method-based ILC with various mixed constraints. Readers may also refer the paper [84] for some concise results and experimental verification.

However, no stochastic noise was considered in the above publications. The SILC algorithm for stochastic point-to-point tracking systems was first addressed in [85]. A linear stochastic system with both system noise and measurement noise were taken into account and the stochastic point-to-point tracking problem was formulated through a variant expression of [82]. The P-type ILC update law with decreasing gain was given, along with the almost sure convergence results directly according to the modified tracking error. As can be seen, SILC for stochastic point-to-point control is just the first step; thus more effort is required to further develop this promising topic.

5.2. SILC with iteration-varying references

It is a basic premise in traditional ILC that the reference trajectory is invariant along the iteration, as repetition is an inherent requirement for the learning strategy. However, this may limit the applications of ILC, since the ILC algorithm has to learn from scratch if the reference varies, which results in a waste of the previously learnt experiences. This was the motivation for research on the iteration-varying references tracking problem, but little progress has so far been made. The reason may be a lack of thorough understanding of how to use existing knowledge when the reference trajectory is changed.

For the iteration-varying references tracking problem, [86] is an earlier study that considered ILC with "slowly" varying trajectories for a continuous-time nonlinear system. Here, "slowly varying" means that the difference of reference trajectories between previous and current iterations is bounded by a small deviation. The update laws with a forgetting factor were proposed and shown to be robust and convergent into a desired tolerance bound. Xu et al. proposed a direct learning control approach for a class of iteration-varying reference tracking problems [87–89], where the references have the same form but different magnitude scales [87,89] or time scales [88]. The same form resulted in the benefit of enabling successful tracking by directly updating the control law. Besides, decentralized ILC algorithms, as well as the strict analysis in the sense of L_p norm, were given in [90,91] for a large-scale industrial system with non-repetitive references. All of the above works focus on the direct ILC.

Another path for the iteration-varying references tracking problem is iteratively learning the invariants, such as system parameters during operations instead of directly updating control. That is, it follows the indirect ILC method. The paper [92] considered a class of parameterized nonlinear system and proposed a parameterized control law using the equivalence principle, where the parameters were iteratively updated following the ILC idea. A composite energy function (CEF) was constructed to show the efficiency of the proposed algorithm. Chi et al. also discussed a class of parameterized high-order system [93], and adopted adaptive ILC to design the parameter estimation and control algorithms. The system parameters were also extended to the iteration-varying case, where the parameter was subject to a second-order internal model [94]. The convergence of the proposed learning control method was shown by using the CEF approach. For the non-parameterized nonlinear system, an intuitive idea is the introduction of a universal approximator to approximate the nonlinear function. This idea was used in [95], where a fuzzy system was used as the approximator to compensate for the plant nonlinearity, on which the adaptive ILC was then formulated. In general, the idea of learning invariants during the process instead of directly updating the control law may play a more important role in iteration-varying references tracking problems.

However, as we have seen, no SILC results have been reported for the iteration-varying references tracking problem. We therefore have to consider how to simultaneously handle iteration-varying references and stochastic noises using the learning algorithm. We believe the SILC approaches for linear/nonlinear stochastic systems should be modified for such an open problem.

5.3. Decentralized/distributed coordinated SILC

The control and tracking of multi-systems that are composed by multiple subsystems according to fixed or varying topological relationships, has been a key issue in the control field. Networked control systems, cruise satellites, large industrial systems, and multi-agent systems are examples of typical multi-systems. It is at the preliminary stage that ILC for the coordinated tracking for multi-objectives [96–104].

The state estimation problem was considered in [96,97]. In [96], the performances of joint estimate and independent estimate were compared for multi-agent system, and showed that joint estimate was superior to independent estimate only under certain conditions. The performance of joint estimate depended on the similarity of multi-agents. The corresponding sensitivity problem of joint estimate was addressed in [97].

The finite-time consensus problem was formulated in [98]. There, each agent was described by a linear system and controlled by a distributed coordinated algorithm based on the terminal ILC. The authors proved that all agents would achieve finite-time consensus as the iteration number going to infinity. Then, the result was extended from two perspectives. One is that the agent model was extended to a class of nonlinear systems [99]. The other one is that the reference was extended to a general trajectory [100].

The formation control problem for multi-agent systems was discussed in [101]. The tracking objective for each agent was not required to be identical. A distributed coordinate algorithm was proposed and proved to be convergent using a contract mapping method. Ahn et al. [102–104] also addressed the formation control problem. In [102] each agent was described as a continuous-time affine nonlinear system and the reference was the relative distance signal among agents. In [104], each agent was described by a single-integrator model and the reference was an absolute relative Euclid distance among agents. In [103], the formation control problem of satellites was discussed, where the reference for each satellite was a given trajectory.

Note that most of the above publications are based on continuous-time deterministic model; thus, it leaves a lot of interesting questions open. As is commonly known, it is required for ILC that the operation should be finished within a limited time interval, which makes our problem different from the traditional consensus problem of multi-agent systems. Therefore, we do not yet know how to properly define decentralized/distributed coordinate SILC in theory and for practical applications. Moreover, the transmissions among subsystems may lead to interesting issues.

6. Concluding remarks

In this paper, research into SILC is surveyed and analyzed. Specifically, SILC for linear stochastic systems and nonlinear systems are first discussed according to the key approaches. As stated earlier, the Kalman filtering based approach and stochastic approximation based approaches are the two major approaches for SILC. Then, SILC for systems with other stochastic signals including packet losses, asynchronism, and time delays are discussed, which, as one could see, is still at the first stage. Moreover, some promising directions are briefly reviewed, namely SILC for point-to-point control, SILC for iteration-varying references tracking, and decentralized/distributed coordinated SILC, respectively. We believe that there is scope for further development in this area.

It is worth noting that studies into the application of SILC are very few. Most papers give numerical simulation results such as [34], which considers two different models of an inductor for angular speed tracking control. However, practical applications are yet to be identified. As is well known, random disturbances/noises are inevitable in practical systems, and more effort is therefore welcomed in this regard.

However, it should be mentioned that although we have try our best to seek as many related SILC papers as possible, we may have missed some important papers. Besides, more attention has been given to SILC by academics and engineers, thus many related papers may be currently in press awaiting publication. Nevertheless, we believe that this paper may help readers to understand the history, current status, and trend of SILC, and it is hoped more publications on this attractive subject will be realized in the future.

Classification of references

- Basic ILC papers [1–4,25–30]
- ILC monographs [5–9]
- ILC surveys [10–12]
- ILC special issues [13–16]
- ILC application papers [17–24]
- SILC for linear stochastic systems [31–37,42–47]
- SILC for nonlinear stochastic systems [49–52]
- SILC with random packet loss [53–61]
- SILC with random asynchronism [62]
- ILC with delays [63–65,60]
- Point-to-point ILC [66–85]
- ILC with iteration-varying references [86–95]
- Decentralized/distributed ILC [96–104]
- Auxiliary materials [38–41,48]

Acknowledgements

This work was supported by National Natural Science Foundation of China (61304085 and 61374099), the Fundamental Research Funds for the Central Universities (ZY1318), and Beijing Higher Education Young Elite Teacher Project (YETP0505).

References

[1] M. Uchiyama, Formulation of high-speed motion pattern of a mechanical arm by trial, Trans. SICE (Soc. Instrum. Contr. Eng.) 14 (6) (1978) 706–712 (in Japanese).
[2] S. Arimoto, S. Kawamura, F. Miyazaki, Bettering operation of robots by learning, J. Robot. Syst. 1 (2) (1984) 123–140.
[3] G. Casalino, G. Bartolini, A learning procedure for the control of movements of robotic manipulators, in: Proc. IASTED Symp. Robot. Automat., 1984, pp. 108–111.
[4] J.J. Craig, Adaptive control of manipulators through repeated trials, in: Proc. Am. Contr. Conf., 1984, pp. 1566–1573.
[5] K.L. Moore, Iterative Learning Control for Deterministic Systems, Advances in Industrial Control, Springer-Verlag, London, 1993.
[6] Z. Bien, J.-X. Xu, Iterative Learning Control – Analysis, Design, Integration and Applications, Kluwer Academic Publishers, Boston, 1998.
[7] Y.Q. Chen, C. Wen, Iterative learning control: convergence, robustness and applications, in: Lecture Notes Series on Control and Information Science, vol. LNCIS-248, Springer-Verlag, London, 1999.
[8] J.-X. Xu, Y. Tan, Linear and Nonlinear Iterative Learning Control. Lecture Notes in Control and Information Sciences, Springer, New York, 2003.
[9] H.-S. Ahn, K.L. Moore, Y.Q. Chen, Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems, Communications and Control Engineering Series, Springer-Verlag, London, 2007.
[10] D.A. Bristow, M. Tharayil, A.G. Alleyne, A survey of iterative learning control: a learning-based method for high-performance tracking control, IEEE Control Syst. Magaz. 26 (3) (2006) 96–114.
[11] H.-S. Ahn, Y.Q. Chen, K.L. Moore, Iterative learning control: survey and categorization from 1998 to 2004, IEEE Trans. System Man Cybern. C 37 (6) (2007) 1099–1121.
[12] Y. Wang, F. Gao, F.J. Doyle III, Survey on iterative learning control repetitive control and run-to-run control, J. Process Control 19 (10) (2009) 1589–1600.
[13] K.L. Moore, J.-X. Xu (Guest Editors), Special issue on iterative learning control, Int. J. Control 73(10) (2000) 819–999.
[14] in: Special issue on iterative learning control, Asian J. Control 4 (1) (2002) 1–118.
[15] H.-S. Ahn, K.L. Moore (Guest Editors), Special issue on iterative learning control, Asian J. Control 13(1) (2011) 1–212.
[16] C.T. Freeman, Y. Tan (Guest Editors), Special issue on iterative learning control and repetitive control, Int. J. Control 84(7) (2011) 1193–1294.
[17] A. Tayebi, S. Abdul, M.B. Zaremba, Y. Ye, in: Robust iterative learning control design: application to a robot manipulator, IEEE/ASME Trans. Mech. 13 (5) (2008) 608–613.
[18] C. Freeman, P. Lewin, E. Rogers, J. Ratcliffe, in: Iterative learning control applied to a gantry robot and conveyor system, Trans. Inst. Meas. Control 32 (3) (2010) 251–264.
[19] K. Inaba, Iterative learning control for industrial robots with end effector sensing (PhD dissertation), University of California, Berkeley, 2008.
[20] D.J. Hoelzle, A.G. Alleyne, A.J.W. Johnson, Iterative learning control for robotic deposition using machine vision, in: American Control Conference, Washington, USA, 2008, pp. 4541–4547.
[21] Y.Q. Chen, K.L. Moore, J. Yu, T. Zhang, in: Iterative learning control and repetitive control in hard disk drive industry – a tutorial, Int. J. Adapt. Control Signal Process. 22 (4) (2008) 325–343.
[22] S.-C. Wu, M. Tomizuka, in: An iterative learning control design for self-servowriting in hard disk drives, Mechatronics 20 (1) (2010) 53–58.
[23] T. Liu, F. Gao, in: IMC-based iterative learning control for batch processes with time delay variation, J. Process Control 20 (2) (2010) 173–180.
[24] T. Liu, F. Gao, in: Robust two-dimensional iterative learning control for batch processes with state delay and time-varying uncertainties, Chem. Eng. Sci. 65 (23) (2010) 6134–6144.
[25] D.H. Owens, J. Hätönen, in: Iterative learning control – an optimization paradigm, Annu. Rev. Control 29 (1) (2005) 57–70.
[26] J.X. Xu, in: A survey on iterative learning control for nonlinear systems, Int. J. Control 84 (7) (2011) 1275–1294.
[27] Y. Tan, H.H. Dai, D. Huang, J.X. Xu, in: Unified iterative learning control schemes for nonlinear dynamic systems with nonlinear input uncertainties, Automatica 48 (12) (2012) 3173–3182.
[28] Y.-C. Wang, C.-J. Chien, in: Design and analysis of fuzzy-neural discrete adaptive iterative learning control for nonlinear plants, Int. J. Fuzzy Syst. 15 (2) (2013) 149–158.
[29] B. Zhang, G. Tang, S. Zheng, in: PD-type iterative learning control for nonlinear time-delay system with external disturbance, J. Syst. Eng. Electron. 17 (3) (2006) 600–605.
[30] C.-J. Chen, C.-Y. Yao, in: An output-based adaptive iterative learning controller for high relative degree uncertain linear systems, Automatica 40 (1) (2004) 145–153.
[31] S.S. Saab, in: A discrete-time stochastic learning control algorithm, IEEE Trans. Autom. Control 46 (6) (2001) 877–887.
[32] S.S. Saab, in: On a discrete-time stochastic learning control algorithm, IEEE Trans. Autom. Control 46 (8) (2001) 1333–1336.
[33] S.S. Saab, in: Stochastic P-type/D-type iterative learning control algorithms, Int. J. Control 76 (2) (2003) 139–148.
[34] S.S. Saab, in: A stochastic iterative learning control algorithm with application to an induction motor, Int. J. Control 77 (2) (2004) 144–163.
[35] S.S. Saab, in: Optimal selection of the forgetting matrix into an iterative learning control algorithm, IEEE Trans. Autom. Control 50 (12) (2005) 2039–2043.
[36] S.S. Saab, in: Optimality of first-order ILC among higher order ILC, IEEE Trans. Autom. Control 51 (8) (2006) 1332–1336.
[37] H.F. Chen, in: Almost sure convergence of iterative learning control for stochastic systems, Sci. Chin. (Ser. F) 46 (1) (2003) 67–79.
[38] H.F. Chen, Stochastic Approximation and Its Applications, Kluwer, Dordrecht, The Netherlands, 2002.
[39] J.C. Spall, in: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, IEEE Trans. Autom. Control 37 (2) (1992) 332–341.
[40] H.F. Chen, T.E. Duncan, B. Pasik-Duncan, in: A Kiefer–Wolfowitz algorithm with randomized differences, IEEE Trans. Autom. Control 44 (3) (1999) 442–453.
[41] H.J. Kushner, G. Yin, Stochastic Approximation Algorithms and Applications, Springer-Verlag, New York, 1997.
[42] D. Shen, Z.S. Hou, in: Iterative learning control with unknown control direction: a novel data-based approach, IEEE Trans. Neural Netw. 22 (12) (2011) 2237–2249.
[43] D. Meng, Y. Jia, J. Du, F. Yu, in: Robust learning controller design for MIMO stochastic discrete-time systems: an H_∞-based approach, Int. J. Adapt. Control Signal Process. 25 (7) (2011) 653–670.
[44] M. Butcher, A. Karimi, R. Longchamp, in: A statistical analysis of certain iterative learning control algorithms, Int. J. Control 81 (1) (2008) 156–166.
[45] M. Norrlóf, in: Disturbance rejection using an ILC algorithm with iteration varying filters, Asian J. Control 6 (3) (2004) 432–438.
[46] S. Gunnarrson, M. Norrlöf, in: On the disturbance properties of high order iterative learning control algorithms, Automatica 42 (11) (2006) 2031–2034.
[47] D.A. Bristow, Frequency domain analysis and design of iterative learning control for systems with stochastic disturbances, in: 2008 Proc. American Control Conference, Seattle, Washington, USA, June 11–13, 2008, pp. 3901–3907.
[48] H.F. Chen, L. Guo, Identification and Stochastic Adaptive Control, Birkhäuser, Boston, 1991.
[49] S.S. Saab, in: Selection of the learning gain matrix of an iterative learning control algorithm in presence of measurement noise, IEEE Trans. Autom. Control 50 (11) (2005) 1761–1774.
[50] H.F. Chen, H.T. Fang, in: Output tracking for nonlinear stochastic systems by iterative learning control, IEEE Trans. Autom. Control 49 (4) (2004) 583–588.

[51] D. Shen, H.F. Chen, in: Iterative learning control for a class of nonlinear systems, J. Syst. Sci. Math. Sci. 28 (9) (2008) 1053–1064 (in Chinese).
[52] D. Shen, H.F. Chen, in: A Kiefer–Wolfowitz algorithm based iterative learning control for Hammerstein–Wiener systems, Asian J. Control 14 (4) (2012) 1070–1083.
[53] H.S. Ahn, Y.Q. Chen, K.L. Moore, Intermittent iterative learning control, in: Proc. the 2006 IEEE Int. Symposium on Intelligent Control, Munich, Germany, October 4–6, 2006, pp. 832–837.
[54] H.S. Ahn, K.L. Moore, Y.Q. Chen, Discrete-time intermittent iterative learning controller with independent data dropouts, in: Proc. the 2008 IFAC World Congress, Seoul, Korea, July, 2008, pp. 12442–12447.
[55] H.S. Ahn, K.L. Moore, Y.Q. Chen, Stability of discrete-time iterative learning control with random data dropouts and delayed controlled signals in networked control systems, in: Proc. the IEEE Int. Conf. Control Automation, Robotics, and Vision, December, 2008.
[56] X. Bu, Z.-S. Hou, F. Yu, in: Stability of first and high order iterative learning control with data dropouts, Int. J. Control Autom. Syst. 9 (5) (2011) 843–849.
[57] X. Bu, F. Yu, Z.-S. Hou, F. Wang, in: Iterative learning control for a class of nonlinear systems with measurement dropouts, Control Theory Appl. 29 (11) (2012) 1458–1464.
[58] X. Bu, F. Yu, Z.-S. Hou, F. Wang, in: Iterative learning control for a class of nonlinear systems with random packet losses, Nonlinear Anal. Real World Appl. 14 (1) (2013) 567–580.
[59] X. Bu, Z.-S. Hou, F.S. Yu, F. Wang, in: $H\infty$ iterative learning controller design for a class of discrete-time systems with data dropouts, Int. J. Syst. Sci. (2013), http://dx.doi.org/10.1080/00207721.2012.757815.
[60] C. Liu, J.-X. Xu, J. Wu, Iterative learning control for network systems with communication delay or data dropout, in: Proc. Joint 48th IEEE Conf. Decision and Control and 28th Chinese's Control Conference, Shanghai, PR China, December 16–18, 2009, pp. 4858–4863.
[61] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, S.S. Sastry, in: Kalman filtering with intermittent observations, IEEE Trans. Autom. Control 49 (9) (2004) 1453–1464.
[62] D. Shen, H.F. Chen, in: Iterative learning control for large scale nonlinear systems with observation noise, Automatica 48 (3) (2012) 577–582.
[63] X.D. Li, T.W.S. Chow, J.K.L. Ho, in: 2D system theory based iterative learning control for linear continuous systems with time delays, IEEE Trans. Circuits Syst. 52 (7) (2005) 1421–1430.
[64] D. Meng, Y. Jia, J. Du, F. Yu, in: Robust iterative learning control design for uncertain time-delay systems based on a performance index, IET Control Theory Appl. 4 (5) (2010) 759–772.
[65] D. Shen, Y. Mu, G. Xiong, in: Iterative learning control for non-linear systems with deadzone input and time delay in presence of measurement noise, IET Control Theory Appl. 5 (12) (2011) 1418–1425.
[66] J.-X. Xu, Y. Chen, T.H. Lee, S. Yamamoto, in: Terminal iterative learning control with an application to RTPCVD thickness control, Automatica 35 (9) (1999) 1535–1542.
[67] Y. Chen, J.-X. Xu, C. Wen, A high-order terminal iterative learning control scheme, in: Proc. the 36th Conference on Decision and Control, 1997, pp. 3771–3772.
[68] L.-P. Zhang, F.-W. Yang, in: Study on the application of iterative learning control to terminal control of linear time-varying systems, Acta Autom. Sin. 31 (2) (2005) 309–313.
[69] G. Gauthier, B. Boulet, Terminal iterative learning control design with singular value decomposition decoupling for thermoforming ovens, in: Proc. American Control Conference, 2009, pp. 1640–1645.
[70] G. Gauthier, B. Boulet, Robust design of terminal ILC with an internal model control using – analysis and a genetic algorithm approach, in: Proc. American Control Conference, 2010, pp. 2069–2075.
[71] S. Boudria, G. Gauthier, High order robust terminal iterative learning control design using genetic algorithm, in: Proc. 38th Annual Conference on IEEE Industrial Electronics Society, 2012, pp. 2313–2318.
[72] J.-X. Xu, D. Huang, in: Initial state iterative learning for final state control in motion systems, Automatica 44 (12) (2008) 3162–3169.
[73] Z. Hou, Y. Wang, C. Yin, T. Tang, in: Terminal iterative learning control based station stop control of a train, Int. J. Control 84 (7) (2011) 1263–1274.
[74] R. Chi, D. Wang, Z. Hou, S. Jin, in: Data-driven optimal terminal iterative learning control, J. Process Control 22 (10) (2012) 2026–2037.
[75] T.D. Son, H.-S. Ahn, Terminal iterative learning control with multiple intermediate pass points, in: Proc. American Control Conference, 2011, pp. 3651–3656.
[76] J. Park, P.-H. Chang, H.-S. Park, E. Lee, in: Design of learning input shaping techniques for residual vibration suppression in an industrial robot, IEEE/ASME Trans. Mech. 11 (1) (2006) 55–65.
[77] J. van de Wijdeven, O. Bosgra, in: Residual vibration suppression using Hankel iterative learning control, Int. J. Robust Nonlinear Control 18 (10) (2008) 1034–1051.
[78] H. Ding, J. Wu, in: Point-to-point control for a high-acceleration positioning table via cascaded learning schemes, IEEE Trans. Ind. Electron. 54 (5) (2007) 2735–2744.
[79] C.T. Freeman, Z. Cai, E. Rogers, P.L. Lewin, in: Iterative learning control for multiple point-to-point tracking application, IEEE Trans. Control Syst. Technol. 19 (3) (2011) 590–600.
[80] T.D. Son, H.-S. Ahn, K.L. Moore, in: Iterative learning control in optimal tracking problems with specific data points, Automatica 49 (5) (2013) 1465–1472.
[81] D.H. Owens, C.T. Freeman, T.V. Dinh, in: Norm-optimal iterative learning control with intermediate point weighting: theory algorithms and experimental evaluation, IEEE Trans. Control Syst. Technol. 21 (3) (2013) 999–1007.
[82] C.T. Freeman, Y. Tan, in: Iterative learning control with mixed constraints for point-to-point tracking, IEEE Trans. Control Syst. Technol. 21 (3) (2013) 604–616.
[83] C.T. Freeman, Y. Tan, Point-to-point iterative learning control with mixed constraints, in: Proc. American Control Conference, 2011, pp. 3657–3662.
[84] C.T. Freeman, in: Constrained point-to-point iterative learning control with experimental verification, Control Eng. Pract. 20 (5) (2012) 489–498.
[85] D. Shen, Y. Wang, Iterative learning control for stochastic point-to-point tracking system, in: Proc. the 12th International Conference on Control, Automation, Robotics and Vision (ICARCV 2012), Guangzhou, China, December 5–7, 2012, pp. 480–485.
[86] S.S. Saab, W.G. Vogt, M.H. Mickle, in: Learning control algorithms for tracking "slowly" varying trajectories, IEEE Trans. Syst. Man Cybern. B 27 (4) (1997) 657–670.
[87] J.-X. Xu, in: Direct learning of control efforts for trajectories with different magnitude scales, Automatica 33 (12) (1997) 2191–2195.
[88] J.-X. Xu, in: Direct learning of control efforts for trajectories with different time scales, IEEE Trans. Autom. Control 43 (7) (1998) 1027–1030.
[89] J.-X. Xu, J. Xu, B. Viswanathan, in: Recursive direct learning of control efforts for trajectories with different magnitude scales, Asian J. Control 4 (1) (2002) 49–59.
[90] X. Ruan, F. Chen, B. Wan, in: Decentralized iterative learning controllers for nonlinear large-scale systems to track trajectories with different magnitudes, Acta Autom. Sin. 34 (4) (2008) 426–432.
[91] X. Ruan, H. Wu, N. Li, B. Wan, in: Convergence analysis in sense of Lebesgue-p norm of decentralized non-repetitive iterative learning control for linear large-scale systems, J. Syst. Sci. Complex. 22 (3) (2009) 422–434.
[92] J.-X. Xu, J. Xu, in: On iterative learning from different tracking tasks in the presence of time-varying uncertainties, IEEE Trans. Syst. Man Cybern. B 34 (1) (2004) 589–597.
[93] R. Chi, Z. Hou, J.-X. Xu, in: Adaptive ILC for a class of discrete-time systems with iteration-varying trajectory and random initial condition, Automatica 44 (8) (2008) 2207–2213.
[94] C. Yin, J.-X. Xu, Z. Hou, in: An ILC scheme for a class of nonlinear continuous-time systems with time-iteration-varying parameters subject to second-order internal model, Asian J. Control 13 (1) (2011) 126–135.
[95] C.-J. Chien, in: A combined adaptive law for fuzzy iterative learning control of nonlinear systems with varying control tasks, IEEE Trans. Fuzzy Syst. 16 (1) (2008) 40–51.
[96] A. Schoellig, J. Alonso-Mora, R. D'Andrea, in: Limited benefit of joint estimation in multi-agent iterative learning, Asian J. Control 14 (3) (2012) 613–623.
[97] A. Schoellig, R. D'Andrea, Sensitivity of joint estimation in multi-agent iterative learning control, in: Proc. the 18th IFAC World Congress, Milano, Italy, 28 August–2 September, 2011, pp. 1204–1212.
[98] D. Meng, Y. Jia, in: Finite-time consensus for multi-agent systems via terminal feedback iterative learning, IET Control Theory Appl. 5 (18) (2011) 2098–2110.
[99] D. Meng, Y. Jia, in: Iterative learning approaches to design finite-time consensus protocols for multi-agent systems, Syst. Control Lett. 61 (1) (2012) 187–194.
[100] D. Meng, Y. Jia, J. Du, F. Yu, in: Tracking control over a finite interval for multi-agent systems with a time-varying reference trajectory, Syst. Control Lett. 61 (7) (2012) 807–818.
[101] Y. Liu, Y. Jia, in: An iterative learning approach to formation control of multi-agent systems, Syst. Control Lett. 61 (1) (2012) 148–154.
[102] H.-S. Ahn, Y.Q. Chen, Multi-agent coordination by iterative learning control: centralized and decentralized strategies, in: ICROS-SICE International Joint Conference 2009, Fukuoka, Japan, 18–21 August, 2009, pp. 3111–3116.
[103] H.-S. Ahn, K.L. Moore, Y.Q. Chen, in: Trajectory-keeping in satellite formation flying via robust periodic learning control, Int. J. Robust Nonlinear Control 20 (14) (2010) 1655–1666.
[104] K.-K. Oh, H.-S. Ahn, in: Formation control of mobile agents based on inter-agent distance dynamics, Automatica 47 (10) (2011) 2306–2312.