Unmanned Systems, Vol. 6, No. 3 (2018) 147–164

DOI: 10.1142/S2301385018400058

Unmanned Systems
http://www.worldscientific.com/worldscinet/us

A Technical Overview of Recent Progresses on Stochastic Iterative Learning Control

Dong Shen

College of Information Science and Technology,
Beijing University of Chemical Technology,
Beijing 100029, P. R. China

This paper contributes to a technical overview of recent progresses on stochastic iterative learning control (ILC), where stochastic ILC implies the learning control for systems with various random signals and factors such as stochastic noises, random data dropouts and inherent random asynchronism. The fundamental principles of ILC are first briefed with emphasis on the system formulations and typical analysis methods. Then the recent progresses on stochastic ILC are reviewed in three parts: additive randomness case, multiplicative randomness case, and coupled randomness case, respectively. Three major approaches, i.e., expectation-based method, Kalman filtering-based method, and stochastic approximation-based method, are clarified. Promising research directions are also presented for further investigation.

Keywords: Stochastic iterative learning control; stochastic systems; additive randomness; multiplicative randomness; coupled randomness; Kalman filtering; stochastic approximation.

1. Introduction

While starting basketball shooting from a fixed position, we may fail for the first several shoots as we have insufficient information about the distance and environments. However, after each shoot, we can learn the information about the basketball shooting process and then improve our shoot angle and power. Thus, we can shoot more and more accurately until we hit the basket. The inherent principle is that we can learn from the past shoots or experiences. This learning ability helps us master almost every skill such as swimming, driving and painting. This basic cognition of learning can be applied also to the industrial systems such as robotics and batch processes. For the latter type of systems, the operation information from previous batches can be fully utilized to improve the performance. In particular, for those systems that operate in a fixed time interval, which will be called an iteration, and repeat the operations successively, the operation information including input and output as well as the tracking reference can be utilized to revise the input signal for the next iteration. As a consequence, the tracking performance is gradually improved as the iteration number increases. This type of control is called iterative learning control (ILC), motivated by the basic concept of learning, which has been an important branch of intelligent control. Clearly, ILC is a typical control strategy that mimics the learning process of human being, in which the pivotal idea is to continuously learn the inherent repetitive factors of system operation processes.

Comparing ILC with other traditional control methodologies such as adaptive control and robust control, we find that ILC has two distinct features. The first feature is that ILC requires the repetitive property of operation processes. In particular, the system should complete each iteration in a fixed time interval; that is, the iteration length is identical for all iterations. For each iteration, the system should be reset to the same initial position; that is, the initial state is identical for all iterations. Moreover, the desired reference is invariant along the iteration axis. In sum, ILC requires invariant iteration length, tracking reference and initial state,

Received 24 April 2018; Revised 13 June 2018; Accepted 14 June 2018; Published 23 August 2018. This paper was recommended for publication in its revised form by Guest Editors, Ying Tan and Qinyuan Ren.
Email Address: *dshen@ieee.org*

so that the update algorithm could learn the inherent invariant factor and then improve the tracking performance along the iteration axis. This cognition has been revealed in [1]. It is a fundamental principle for learning-based mechanism in control. ILC makes an in-depth utilization of the available information to learn the inherent invariance. The second distinct feature of ILC is that it requires little information of the system. In other words, ILC is a typical data-driven control strategy because the generation of the input for the next iteration completely depends on the input and tracking information of previous iterations. In particular, the typical structure of ILC update laws is a predefined function of the input and output/tracking information. Consequently, ILC is effective in dealing with many traditional control difficulties such as high nonlinearity, strong coupling, modeling difficulty and tracking of high precision. In essence, ILC is a kind of integral control along the iteration axis for a fixed tracking trajectory.

We should point out that both features have been deeply investigated and extended in the past decades. On the one hand, much effort has been devoted to relax the invariance limitation of ILC so that the application range can be broadened. For example, various initial state conditions were discussed and compared in [2], where the alignment condition of initial state was proposed and analyzed to remove the space resetting requirement. Some initial state learning or rectifying mechanisms were proposed in [3] and [4] to offer alternative schemes addressing the identical initial condition. The nonrepetitive system dynamics was discussed in [5, 6] to understand the essential limitation of learning ability. Moreover, recent publications [7, 8] provided an in-depth discussion on the random iteration-varying length problems, which clearly remove the invariant operation length assumption. On the other hand, although scholars have contributed many works to design suitable data-driven algorithms to facilitate various application scenarios, the involvement of system information may provide extra advantage in handling the transient performance of the learning control. For example, several papers have been published on the combination of feedback control in time domain and feedforward control in iteration domain [9–11], which can enhance the stability and improve tracking precision simultaneously.

The concept of ILC was first proposed by Uchiyama in [12], which was written in Japanese and thus not widely spread. The paper in 1984 published by Arimoto *et al.* was widely recognized as the initiation of ILC [13], where the concept of learning was applied to robot control for repetitive tasks. After that, a large amount of papers have been published on various issues of ILC. To name a few, special issues were launched by International Journal of Control [14, 15], Asian Journal of Control [16, 17], and Journal of Process Control [18]. For survey papers the readers can refer to [19–23], where different emphases are highlighted. In particular, the tutorial introduction was given in [19], a literature classification from 1998–2004 was provided in [20], the first detailed survey on stochastic ILC was given in [23], the composite energy function approach-based synthesis and design was clarified in [22] and a detailed comparison of ILC, repetitive control, run-to-run control was given in [21]. Moreover, the readers may also refer to the monographs to make an in-depth understanding of the theory issues and applications of ILC [24–33]. From these advances it is observed that many fundamental issues of ILC have been carefully explored such as the initial state condition [2–4], Lipschitz condition on nonlinear functions [25, 26], optimization of the learning gains [29], practical implementations [30, 31], repetitive requirements on the systems settings [32, 33], and robustness issues [27]. In addition, contraction mapping method, 2D system and repetitive process-based approach, and composite energy function-based method have been proposed and developed as the mainstream methods for addressing various ILC problems.

When considering the control of practical systems, it is observed that various stochastic factors exist in these systems. For example, random process disturbances and measurement noises are generally unavoidable in most systems (either bounded or unbounded), which makes the systems themselves to be stochastic systems. Moreover, in networked control structure of the practical application, where the plant and the controller are separated in different sites and communicate with each other through wired/wireless networks, the random data dropouts are very common due to limited bandwidth or data congestion. Furthermore, multi-agent systems have become a hot topic to the control community, where the communication among agents usually suffer various randomness including communication noises, linkage breaks, and fading channel. In addition, the updating process among subsystems of a large-scale system is generally randomly asynchronous rather than synchronous. All these types of randomness are generally described by random variables in probability theory. The random variables may or may not have some statistical properties. If the statistical properties such as mean and variance are known, we can utilize this information to make a primary compensation for the random signals; otherwise, we need to design the random signal independent algorithms to facilitate wide applications. This distinction makes the synthesis and analysis of stochastic ILC evidently different from the traditional ILC problem, which has become one of the important directions in the current research of ILC.

Generally, stochastic ILC indicates the part of ILC concentrating on systems with various stochastic signals or factors, where the stochastic signal or factor is described by a random variable with/without a specified probability distribution. The main feature of stochastic ILC is the introduction of random variables, which makes the conventional analysis techniques for deterministic systems inapplicable. In a previous review paper [23], the research on stochastic ILC is classified in term of the systems formulations, i.e., linear systems with system disturbances and/or measurement noises, nonlinear systems with system disturbances and/or measurement noises, and systems with other kind of stochastic signals such as data dropouts and random asynchronism. The previous results in these categories are reviewed according to their problems and formulations. Unlike [23], we will review the recent progresses on stochastic ILC according to the position of random variables in this paper. In particular, we will make the following categories: additive randomness case, multiplicative randomness case and coupled randomness case. In additive randomness case, the random signals or factors are added to the system equations and update laws as an individual part. In multiplicative randomness case, the random signals or factors are multiplied to the system variables such as state, tracking error, and input. In the coupled randomness case, we mean the random signals or factors are formulated as an inherent part of the original signals, which cannot be separated as an additive term or multiplicative coefficient. Moreover, we contribute our major effort to reveal the essential analysis techniques for each case, so that this paper may shed light on the hidden critical points behind the complex derivations. As a consequence, we expect more interesting results in the future along with the guidelines provided in this paper.

The structure of the paper is arranged as follows. In Sec. 2, the fundamental principles of ILC is proposed with basic formulations of systems and update laws. The typical methods for deterministic systems are also reviewed in this section. In Sec. 3, the main problems and critical techniques for systems with additive random signals and factors are elaborated. The multiplicative random signals and factors case is reviewed in Sec. 4. The systems with coupled randomness that cannot be separated as additive or multiplicative case are discussed in Sec. 5, where the emphasis lies on the possible solutions. Based on the review of these three aspects, the promising research directions are given in Sec. 6, which attempt to make a guide for further investigation. Concluding remarks are presented in Sec. 7.

2. Fundamental Principles of ILC

In this section, we present the fundamental principles of ILC based on the discrete-time system models. The reasons for selecting discrete-time systems are twofold: on the one hand, many systems adopt the computer-aided control framework, which is essentially discrete; on the other hand, the discrete-time model facilitates the formulation of random signals and factors. Then, we proceed to provide a review on the typical methods for deterministic systems, where the contraction mapping method and 2D system/repetitive process-based approach are addressed.

2.1. *Fundamental formulation of ILC*

The basic structure of ILC consists of a plant, a learning controller, and a memory device, as shown in Fig. 1, where the plant denotes the repetitive operation process, the learning controller generates the input based on a specified design form, and the memory device is used to store the signals of previous iterations. For the kth iteration, the input $u_k(t)$ of the whole iteration interval is fed to the plant and the corresponding output $y_k(t)$ is produced, which may have a certain degree of deviation from the desired tracking reference $y_d(t)$. Then, all these signals are utilized in the learning controller to generate the input signal $u_{k+1}(t)$ for the next iteration, which will be sent to the plant and stored in the memory device simultaneously. The synthesis objective of ILC is to propose a proper update law for the learning controller, and the analysis objective of ILC is to investigate the conditions for asymptotical convergence of the output $y_k(t)$ to the desired reference $y_d(t)$ as the iteration number k increases and study other performance indices such as transient performance, robustness and final tracking precision. Therefore, ILC differs from traditional control methodologies such as PID control, robust control and adaptive control in the major aspect that ILC concentrates on the iteration-axis-based performance improvement while traditional control methodologies pay most attention to the time-axis-based performance adjustment. In other words, ILC is a kind of 2D process.

To make a formal clarification, consider the following discrete-time linear system:

$$\begin{aligned} x_k(t+1) &= A_t x_k(t) + B_t u_k(t), \\ y_k(t) &= C_t x_k(t), \end{aligned} \tag{1}$$

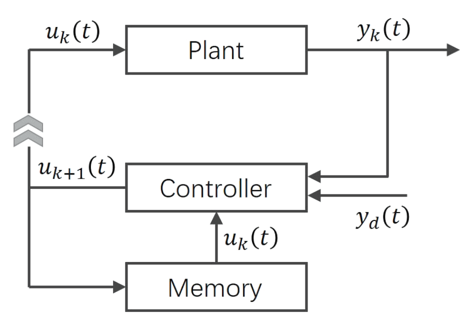

Fig. 1. Basic structure of ILC.

where k denotes the iteration index and t denotes the time index. $x_k(t) \in \mathbf{R}^n$, $u_k(t) \in \mathbf{R}^p$ and $y_k(t) \in \mathbf{R}^q$ are the state, input and output, respectively. A_t, B_t and C_t are time-varying system matrices with appropriate dimensions. Generally, we let t be valued from $\{0, 1, \ldots, N\}$ with N denoting the length of an operation iteration. For simplicity, in the following, we use $t \in [0, N]$ to denote $t \in \{0, 1, \ldots, N\}$.

The discrete-time nonlinear affine system can be expressed as follows:

$$\begin{aligned} x_k(t+1) &= h(x_k(t)) + B(x_k(t))u_k(t), \\ y_k(t) &= C_t x_k(t), \end{aligned} \tag{2}$$

where $h(\cdot)$ and $B(\cdot)$ are nonlinear functions. In general, the output equation can be formulated as $y_k(t) = g(x_k(t))$ with $f(\cdot)$ being a nonlinear function. We formulate (2) because it will be referred to later.

The desired reference to track is denoted by $y_d(t)$, $t \in [0, N]$. The general control objective of ILC is to derive some update law such that $y_k(t) \to y_d(t), \forall t$. Moreover, in ILC, it is required that the system can repeat its process from the same starting position/state, which implies that the initial state of the above dynamic evolution should be reset precisely at each iteration. This requirement is formulated as $x_k(0) = x_d(0)$, $\forall k$, where $x_d(0)$ denotes the desired initial state in accordance to $y_d(0)$, i.e., $y_d(0) = C_0 x_d(0)$.

For the kth iteration, we denote the tracking error as

$$e_k(t) = y_d(t) - y_k(t), \quad \forall t. \tag{3}$$

Generally, the update law for generating $u_{k+1}(t)$ is formulated as a function of $u_i(t)$ and $e_i(t)$ (or equivalently, $y_i(t)$ and $y_d(t)$), $1 \le i \le k$, $t \in [0, N]$,

$$u_{k+1}(t) = f(u_k(\cdot), \ldots, u_1(\cdot), e_k(\cdot), \ldots, e_1(\cdot)). \tag{4}$$

If $f(\cdot)$ is a linear function of its arguments, it is a linear update law; otherwise, it is a nonlinear update law. Moreover, if the above relationship depends only on the last iteration, it is called a first-order update law; otherwise, it is called a high-order update law [20]. In the literature, most papers adopt the first-order type for simplicity of the algorithm and it has been well revealed whether the high-order update law can surpass the first-order one [20, 34]. The general first-order update law is

$$u_{k+1}(t) = f(u_k(\cdot), e_k(\cdot)). \tag{5}$$

Further, the update law is usually linear, which is simple for both implementation and convergence analysis. In this case, if the relative degree of the system (1) is one (that is, the matrix $C_{t+1}B_t$ is not zero), a P-type update law is as follows,

$$u_{k+1}(t) = u_k(t) + L_t e_k(t+1), \tag{6}$$

where L_t is the learning gain matrix. If we replace the innovation term $L_t e_k(t+1)$ with $L_t[e_k(t+1) - e_k(t)]$, the update law is a D-type one [20].

In this paper, we present a survey on recent progresses on stochastic ILC, thus we mainly consider the case that various randomness is imposed to the above formulations. For example, the plant model may contain process disturbances and measurement noises, the communication between the plant and the controller may contain communication noises, leading to random data dropouts, and the updating process in the controller may involve random asynchronism due to data mismatch or disordering. All these random signals and factors will make the analysis and design of the corresponding algorithms much more difficult. In order to deal with these randomness, some wildly used techniques are revisited in this paper.

2.2. *Typical methods for deterministic systems*

2.2.1. *Preliminary*

The lifting technique is an important transformation for discrete-time ILC as it can fold the time-axis process dynamics by supervectors and highlight the iteration-axis evolution. In particular, we define the supervectors $\mathbf{u}_k = [u_k^T(0), u_k^T(1), \ldots, u_k^T(N-1)]^T$, $\mathbf{y}_k = [y_k^T(1), y_k^T(2), \ldots, y_k^T(N)]^T$, then from (1) we have

$$\mathbf{y}_k = H\mathbf{u}_k + \mathbf{d}_k, \tag{7}$$

where $\mathbf{d}_k = [(C_1A_0)^T, (C_2A_0^1)^T, \ldots, (C_NA_0^{N-1})^T]^T x_k(0)$,

$$H = \begin{bmatrix} C_1B_0 & 0 & \cdots & 0 \\ C_2A_1B_0 & C_2B_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_NA_1^{N-1}B_0 & C_NA_2^{N-1}B_1 & \cdots & C_NB_{N-1} \end{bmatrix}, \tag{8}$$

with $A_i^j \triangleq A_jA_{j-1}\cdots A_i$ and $A_i^j = I$ if $i > j$. By replacing the subscript k with d, we can define $\mathbf{y}_d$ similar to $\mathbf{y}_k$ and then $\mathbf{e}_k = \mathbf{y}_d - \mathbf{y}_k$. P-type update law (6) can be reformulated as

$$\mathbf{u}_{k+1} = \mathbf{u}_k + L\mathbf{e}_k, \tag{9}$$

where $L = \text{diag}\{L_0, \cdots, L_{N-1}\}$. The lifted dynamics (7) and update law (9) are adopted in many papers for facilitating the convergence analysis. As can be seen from (7), the time index t has been removed and it is an iteration-based mapping from the input $\mathbf{u}_k$ to the output $\mathbf{y}_k$. Throughout this paper, we use plain notations to denote vectors or scalars with respect to specific time instant and iteration number, while use bold notations to denote the supervectors or equivalently the stacked large-vectors of vectors.

In general, there are two different ways to show the convergence. The first one is the direct method, which shows the convergence in terms of tracking error. The second one is indirect method, by assuming the unique $\mathbf{u}_d$, which corresponds to the desired output, the convergence of control input will indicate the convergence of the output.

2.2.2. *Contraction mapping method*

There are a few methods that are widely used to show the convergence of ILC for deterministic systems. Contraction mapping (CM) method is the most popular method in the verification steps of ILC analysis. Various extensions and variations are proposed in the literature according to different environments. The inherent principle of the CM method is the well-known fixed-point principle. Note that in ILC, the main objective is to show the convergence of the output to the desired reference by retaining the system dynamics iteration-invariant, thus it is possible to apply the fixed-point principle for convergence analysis. In particular, let us revisit the lifted update law (9) and substitute $\mathbf{e}_k = \mathbf{y}_d - \mathbf{y}_k$,

$$\begin{aligned} \mathbf{u}_{k+1} &= \mathbf{u}_k + L(\mathbf{y}_d - \mathbf{y}_k) \\ &= \mathbf{u}_k + LH(\mathbf{u}_d - \mathbf{u}_k), \end{aligned} \tag{10}$$

where the identical initialization condition $x_k(0) = x_d(0)$ is applied. Then, subtracting both sides of the last equation from $\mathbf{u}_d$ and denoting $\delta\mathbf{u}_k = \mathbf{u}_d - \mathbf{u}_k$, we have

$$\delta\mathbf{u}_{k+1} = (I - LH)\delta\mathbf{u}_k. \tag{11}$$

It is evident that $\delta\mathbf{u}_k \to 0$ if we can design L such that the spectral radius of $I - LH$ is less than 1. That is, $\rho(I - LH) < 1$, where $\rho(M)$ denotes the spectral radius. Noting that I is the identity matrix, the above condition can be reformulated with respect to LH directly. Moreover, scholars can also derive convergence conditions in the norm sense. For example, $\|\delta\mathbf{u}_k\| \to 0$ if L is designed such that $\|I - LH\| < 1$, where $\|\cdot\|$ denotes compatible norms for vectors and matrices. Clearly, the existence of the learning gain matrix L heavily depends on the system matrix H. A sufficient condition to guarantee the existence of L is that the matrix H is of full-column rank [35].

Remark 2.1. For simplicity of presentation, the proof highlighted here is based on the indirect method. The similar idea can be found when direct method is used in [5, 6, 36].

For the nonlinear system (2), the CM method can be effective if the nonlinear functions are globally Lipschitz continuous (GLC), that is, there exist positive constants h_1 and b_1 such that $\|h(x) - h(y)\| \le h_1\|x - y\|$ and $\|B(x) - B(y)\| \le b_1\|x - y\|$. Note that the nonlinear system cannot be lifted similar to the linear case, it is difficult to derive the iteration-based evolution form. Therefore, the CM method can be directly applied to linear systems or nonlinear systems with GLC using the well-known Grönwall lemma, which leads to the wide applications of the α-norm of the indicated variables. The α-norm of $u_k(t)$ is defined as $\sup_{0\le t\le N}\alpha^{-\lambda t}\|u_k(t)\|$, where $\alpha > 1$ and $\lambda > 1$ are suitably selected values according to the specific systems. For details, readers may refer to [26, 69]. We remark that the GLC is required mainly due to the application of the Gronwall lemma or α-norm techniques. If we prove the convergence by mathematical induction method with respect to the time axis, the globally Lipschitz condition of nonlinear functions can be relaxed to locally Lipschitz condition [37–39].

2.2.3. 2*D theory approach*

As early as 1990s, 2D system theory has been applied to deal with ILC problems [40–42], which was then developed as a major method in the design and analysis of ILC algorithms. The inherent principle is that ILC essentially constitutes a 2D system because ILC evolves along both time and iteration axes. The 2D systems indicate those with independent evolutions along two directions simultaneously [43]. Therefore, the main procedure for the 2D system-based method is as follows: first, transform the closed-loop system with ILC algorithms into a 2D system, and then, apply the stability theory from the conventional 2D system theory to the newly transformed system. Clearly, the developments in this way depend much on the original progresses of 2D system theory. An important research direction is to apply the approach to newly emerging circumstances.

As an illustration, we apply the following D-type update law for (1):

$$u_{k+1}(t) = u_k(t) + L_t[e_k(t+1) - e_k(t)], \tag{12}$$

and define $\delta x_k(t) = x_d(t) - x_k(t)$ and $\delta u_k(t) = u_d(t) - u_k(t)$, where $x_d(t)$ and $u_d(t)$ are the desired state and input, respectively, associated with the given reference $y_d(t)$. Then, we have

$$\delta x_k(t+1) = A_t\delta x_k(t) + B_t\delta u_k(t) \tag{13}$$

and

$$\begin{aligned} \delta u_{k+1}(t) &= \delta u_k(t) - L_t[C_{t+1}\delta x_k(t+1) - C_t\delta x_k(t)] \\ &= \delta u_k(t) - L_tC_{t+1}[A_t\delta x_k(t) + B_t\delta u_k(t)] + L_tC_t\delta x_k(t) \\ &= [I - L_tC_{t+1}B_t]\delta u_k(t) + L_t[C_t - C_{t+1}A_t]\delta x_k(t). \end{aligned} \tag{14}$$

Therefore, we have derived a 2D system as follows:

$$\begin{bmatrix} \delta u_{k+1}(t) \\ \delta x_k(t+1) \end{bmatrix} = \begin{bmatrix} I - L_tC_{t+1}B_t & L_t(C_t - C_{t+1}A_t) \\ B_t & A_t \end{bmatrix} \begin{bmatrix} \delta u_k(t) \\ \delta x_k(t) \end{bmatrix}. \tag{15}$$

If one would like to apply the P-type update law (6) and involve the tracking error directly, we may define $\Delta x_k(t) = x_{k+1}(t) - x_k(t)$ and $\Delta u_k(t) = u_{k+1}(t) - u_k(t)$. Then,

$$\begin{aligned} e_{k+1}(t+1) &= e_k(t+1) - C_{t+1}A_t\Delta x_k(t) - C_{t+1}B_t\Delta u_k(t) \\ &= [I - C_{t+1}B_tL_t]e_k(t+1) - C_{t+1}A_t\Delta x_k(t) \end{aligned} \tag{16}$$

and

$$\begin{aligned} \Delta x_k(t+1) &= A_t\Delta x_k(t) + B_t\Delta u_k(t) \\ &= A_t\Delta x_k(t) + B_tL_te_k(t+1). \end{aligned} \tag{17}$$

Therefore, we have another 2D system formulation as follows:

$$\begin{bmatrix} e_{k+1}(t+1) \\ \Delta x_k(t+1) \end{bmatrix} = \begin{bmatrix} I - C_{t+1}B_tL_t & -C_{t+1}A_t \\ B_tL_t & A_t \end{bmatrix} \begin{bmatrix} e_k(t+1) \\ \Delta x_k(t) \end{bmatrix}. \tag{18}$$

At the end of this section, we note that the repetitive process has been deeply investigated in the past decades and fruitful results have been obtained, which has shown its effectiveness in design and analysis of corresponding ILC algorithms [44–46]. Novel results are expected along this direction by connecting repetitive processes with ILC formulations.

3. Additive Randomness Case

In this section, we review the major techniques for systems with additive randomness. Here, by additive randomness we mean the random signals/factors are involved into the systems as individual portions. For examples, the operation process may involve random disturbances due to various factors; the measurement of output signals may be influenced by random noises; and the data transmission of networks would introduce additional communication noises. All these signals are generally described by random variables which are additive to the original system formulas. It should noted that the additive noises will always exist, no matter whether the original signal occurs, since they are in an additive form. To facilitate the performance analysis and without loss of generality, the additive randomness is assumed to be with zero mean and finite moments.

In order to demonstrate that the additive randomness are quite common, this section starts from a few examples of additive randomness.

3.1. *Examples of additive randomness*

Example 1. (Random Initial States). Consider the lifted formulation (7), where we notice that the response to the initial state is expressed by an individual term $\mathbf{d}_k$. In many papers, the identical initialization condition is assumed, i.e., $x_k(0) = x_d(0)$, then the influence of the initial state is eliminated. However, in practical applications, the precise reset of the initial state is hard to achieve. In fact, the initial state may vary from iteration to iteration randomly in a small bound. Thus, it is reasonable to assume that $x_k(0)$ is a random variable around $x_d(0)$ with its expectation being the desired initial state $x_d(0)$. In this case, $\mathbb{E}x_k(0) = x_d(0)$ and $\sup_k\mathbb{E}[\|x_k(0) - x_d(0)\|^2] < \infty$. Clearly, $\mathbf{d}_k$ is an additive random variable in the linear formulation (7).

Example 2. (Stochastic Linear Systems). Consider the linear system (1) with random disturbances and noises,

$$\begin{aligned} x_k(t+1) &= A_tx_k(t) + B_tu_k(t) + w_k(t+1), \\ y_k(t) &= C_tx_k(t) + v_k(t), \end{aligned} \tag{19}$$

where $w_k(t)$ and $v_k(t)$ can be formulated as zero-mean white noises in most applications. This model has been studied in many papers as it denotes a general stochastic linear system. If we apply the lifting technique to this model, it follows that

$$\mathbf{y}_k = H\mathbf{u}_k + \mathbf{d}_k + \varepsilon_k, \tag{20}$$

where

$$\epsilon_k = \begin{bmatrix} v_k(1) + C_1w_k(1) \\ \vdots \\ v_k(N) + C_N\sum_{i=1}^{N} A_i^{N-1}w_k(i) \end{bmatrix}.$$

Clearly, all the process disturbances and measurement noises can be separated as additive noises and all these noises are independent with the original process.

Example 3. (Nonlinear Systems with Measurement Noises). Consider the nonlinear system (2) with measurement noises,

$$\begin{aligned} x_k(t+1) &= h(x_k(t)) + B(x_k(t))u_k(t), \\ y_k(t) &= C_tx_k(t) + v_k(t), \end{aligned} \tag{21}$$

where $v_k(t)$ denotes the measurement noise. Clearly, the measurement noise is additive. Moreover, this model can also be used to describe the case that the original system is deterministic but the output is involved with communication noise during transmission. It should be pointed out that the process disturbance is not considered in (21). Otherwise, the random disturbance would be coupled with the nonlinear dynamics $h(\cdot)$ and $B(\cdot)$, which therefore is no longer additive randomness but coupled randomness.

Example 4 (Probabilistically Quantized Error). For deterministic linear system (1) and update law (6), we present the quantized ILC problem. In particular, the output used in the update law (6) is not the original output $y_k(t)$ but the quantized measurement $\hat{y}_k(t)$. In this case, update law (6) is formulated as

$$u_{k+1}(t) = u_k(t) + L_t[y_d(t+1) - \hat{y}_k(t+1)], \tag{22}$$

where

$$\hat{y}_k(t) = \mathcal{Q}(y_k(t))$$

with $\mathcal{Q}(\cdot)$ being a probabilistic quantizer. For a real number v, the probabilistic quantizer $\mathcal{Q}(\cdot)$ is defined as

$$\mathcal{Q}(v) = \begin{cases} \lfloor v \rfloor, & \text{with probability } \lfloor v \rfloor + 1 - v \\ \lfloor v \rfloor + 1, & \text{with probability } v - \lfloor v \rfloor \end{cases}. \tag{23}$$

For a vector, the probabilistic quantizer is defined according to each entry. By simple calculations, we have that the probabilistic quantizer is unbiased, $\mathbb{E}[\mathcal{Q}(v)] = v$. Moreover, the variance for the quantization error is bounded, $\mathbb{E}[(v - \mathcal{Q}(v))^2] \leq 1/4$ when v is a number. Denote $r(v) = v - \mathcal{Q}(v)$ as the quantization error, then we can rewrite the update law (22) as follows:

$$u_{k+1}(t) = u_k(t) + L_t e_k(t+1) + L_t r(y_k(t+1)). \tag{24}$$

Clearly, the probabilistic quantization error is an additive randomness term.

It is evident that all the randomness signals and factors, including the random initial states, process disturbances, measurement noises, and quantization errors would be transformed as an additive term in the update law. Because these types of randomness cannot be predicted and eliminated, the input sequence generated by the update law with fixed step cannot converge to a stable limitation but may fluctuate in a small bound. As a consequence, the corresponding output cannot precisely track the desired reference due to the existence of random signals. Thus, when the output is coupled with random signals such as those in Examples 1–3, the tracking objective should be revised as an optimization index of the tracking error, for example, $V_t = \limsup_{n\to\infty} \frac{1}{n}\sum_{i=k}^{n} \|e_k(t)\|^2$. Moreover, to obtain a stable convergence of the input sequence, a decreasing step should be introduced to suppress the effect of random signals in the update law, which is a mature technique in stochastic control and optimization.

Next a few subsections will discuss the common techniques to deal with ILC that can handle additive randomness. In stochastic ILC, the most popular methods for addressing random signals/factors include expectation-based method, Kalman filtering-based method, and stochastic approximation-based method [23]. These methods will be reviewed in the following subsections subsequently.

3.2. *Expectation based method*

The expectation-based method has been applied in several papers [47, 48]. The main advantage of this method is the elimination of the randomness. In particular, the main procedures for the application of this method are as follows: first, an expectation is taken to both sides of the update law or other equivalent relationships; then, all variables containing randomness are transformed into deterministic ones; and then, the following procedures for addressing the control performance can be completed by the conventional analysis steps.

For example, the paper [47] presented the following lifted formulation with stochastic noises,

$$\mathbf{y}_k = H\mathbf{u}_k + \varepsilon_k, \tag{25}$$

where ϵ_k is the lifted noise vector. It is assumed to be white noise with $\mathbb{E}\varepsilon_k = 0$, $\mathbb{E}[\varepsilon_k \varepsilon_k^T] = V$, $\mathbb{E}[\varepsilon_k \varepsilon_{k+1}^T] = 0$, $i \neq 0$, where V is positive definite.

The update law is a P-type one in lifted form (9). By defining $H_e = I - HL$, it is obvious that

$$\mathbf{e}_k = H_e \mathbf{e}_{k-1} + \varepsilon_{k-1} - \varepsilon_k. \tag{26}$$

To prove the convergence, expectation is taken to both sides of Eq. (26). This treatment implies that the mathematical expectation of $\mathbf{e}_k$ converges to zero if the spectral norm of $\rho(H_e) < 1$. Besides, the variance matrix $\mathrm{Var}[\mathbf{e}_k]$ is also shown to converge to some constant matrix. From this example, it is seen that the expectation-based method is easy for eliminating the additive randomness and transforming the original relationship into a deterministic type.

However, the expectation-based method has some distinct limitations. First, the expectation of the tracking error converging to zero is not always as good as expected, because it may result in a large tracking error if the covariance limit is large. In other words, even if the expectation of the tracking error converges to zero, the actual tracking error may be quite large due to the accumulation of the random signals. Moreover, the variables in the derived equations should be independent, so that the expectation can be taken to each variable independently for product terms. Last but not least, the expectation-based method is mainly appropriate for linear systems and linear laws, but generally not applicable to nonlinear systems nor nonlinear laws, because nonlinearities may make the variables coupled together and then the expectation is hardly taken to the inherent randomness. To sum, the application range of the expectation method is narrow due to these limitations.

3.3. *Kalman filtering based method*

Kalman filtering has shown its valuable effect in eliminating stochastic noises and estimating the actual state information in the conventional control field [49]. The Kalman filter has numerous applications in practical systems and technologies such as guidance, navigation, and control of vehicles. Although extensions and generalizations of Kalman filtering have been developed much such as extended Kalman filter and unscented Kalman filter for nonlinear systems, the most favorable applications of Kalman filter is for linear systems where stochastic noises are with good statistical properties.

The conventional Kalman filtering algorithm includes two steps. The first step is called prediction, in which the current state variable is estimated on the basis of the available data; The second step is called update, in which the prior estimation is corrected with the innovation term generated from the new measurement and the optimal Kalman gain is updated. It should be emphasized that the recursive optimal Kalman gain is calculated by optimizing the covariance of the error between the predicted and measured output/state. This idea can be applied to derive the filtering algorithms for a 2D system, which further leads to the novel ILC update laws with time- and iteration-varying learning gains [50–53].

To see this point clearly, let us consider the application of the D-type update law (12) to the stochastic linear system (19), where the learning gain L_t is replaced with $L_{t,k}$ to denote the iteration-dependence [50]. In this case, similar to the derivations in Sec. 2.2, we can obtain the following 2D formulation,

$$\begin{bmatrix} \delta u_{k+1}(t) \\ \delta x_k(t+1) \end{bmatrix} = \begin{bmatrix} I - L_{t,k}C_{t+1}B_t & L_{t,k}(C_t - C_{t+1}A_t) \\ B_t & A_t \end{bmatrix} \times \begin{bmatrix} \delta u_k(t) \\ \delta x_k(t) \end{bmatrix} + \begin{bmatrix} L_{t,k}C_{t+1} & L_{t,k} \\ -I & 0 \end{bmatrix} \times \begin{bmatrix} w_k(t+1) \\ v_k(t+1) - v_k(t) \end{bmatrix}, \tag{27}$$

where we remind that $\delta x_k(t) = x_d(t) - x_k(t)$ and $\delta u_k(t) = u_d(t) - u_k(t)$, $\forall t, k$. Assume that all the random variables $\{w_k(t)\}$ and $\{v_k(t)\}$ are independent sequences of white noises with zero-mean and positive-definite covariance matrices. The initial state error and the initial input error are also assumed to be zero-mean white noises. The initial state error is uncorrelated with other random signals including initial input error, process disturbances, and measurement noises. All these assumptions are made to facilitate the application of the Kalman filtering technique.

Denote $X^+ = [(\delta u_{k+1}(t))^T\ (\delta x_k(t+1))^T]^T$. The recursive learning gain $L_{t,k}$ is calculated such that the trace of the error covariance matrix $P^+ \triangleq \mathbb{E}[X^+(X^+)^T]$ is minimized. In other words, it is calculated from the following equation,

$$\frac{d(\mathrm{trace}(P^+))}{dL_{t,k}} = 0. \tag{28}$$

Substituting the detailed expansion of P^+ (for details, please refer to [50]), we are able to derive

$$L_{t,k} = P_{u,t,k}(C_{t+1}B_t)^T[(C_{t+1}B_t)P_{u,k}(C_{t+1}B_t)^T + S_k]^{-1}, \tag{29}$$

where S_k is a positive-definite matrix associated with the state error covariance and the covariance matrices of random noises, and the input error covariance matrix $P_{u,t,k}$ is recursively defined by

$$P_{u,t,k+1} = (I - L_{t,k}C_{t+1}B_t)P_{u,t,k}. \tag{30}$$

Clearly, for any fixed time instant, the above recursions along the iteration axis are consistent with the classical expressions of Kalman filter. Later, it was proved that any positive-definite matrix selection of S_k can guarantee the mean-square convergence of the input error to zero [52]. This relaxation has greatly removed strong dependence on the system information in the derived algorithms. For the P-type update law (6), similar recursive update algorithms can be derived [52].

Indeed, the Kalman filtering-based method, which was proposed by Saab in the early 2000s, has successfully established a systematic framework for treating stochastic linear systems with good statistical properties of all involved random signals. The mean-square convergence of the proposed algorithms can be obtained, which is much stronger than the expectation-based method. Thus, it is of great significance for practical applications. Moreover, the recursive calculation of the learning gain is adaptive in both time domain and iteration domain. This will benefit the iteration-varying processes.

The main procedures for Kalman filtering-based method are as follows. First, build a 2D model with respect to the input error and state error. Next, calculate the derivative of the trace of the input error covariance matrix to generate the learning gain. Then, prove the mean-square convergence of the derived recursive algorithms and analyze the tracking performance.

Along this direction, some open problems exist. First, in order to obtain good convergence results, the initial input error is assumed to be a zero-mean white noise [50, 52], which means that the initial input $u_k(t)$ should be normally distributed around the desired input $u_d(t)$. However, it is hard to satisfy this condition when little system information is known in advance. Therefore, how to relax the requirement on the initial input is an interesting problem for practical applications. Moreover, the noise assumptions are somewhat restrictive and it is meaningful to consider the possible relaxations of this condition. Last, the application of the Kalman filtering-based method has been widely found. It is believed such method can well handle other ILC problems for stochastic linear systems. The research on this issue is also open and fruitful results are expected.

3.4. *Stochastic approximation based method*

The stochastic approximation algorithm is an effective root-seeking or extrema-seeking approach for unknown functions with noisy observations [54, 55]. The typical algorithms are Robbins–Monro [56] (RM) algorithm and Kiefer–Wolfowitz [57] (KW) algorithm. The basic principles of applying these algorithms in ILC are as follows. If there exists some desired input such that the desired reference can be realized, that is, $\mathbf{y}_d = \boldsymbol{g}(\mathbf{u}_d)$ with $\boldsymbol{g}(\cdot)$ denoting the general function, then the tracking problem can be solved as long as we can design an update law satisfying that the generated input converges to the desired input $\mathbf{u}_d$. In this case, $\mathbf{u}_d$ can be regarded as the root of the function $\mathbf{y}_d - \boldsymbol{g}(\mathbf{u})$. For this function, we can only access the noisy observations $\mathbf{e}_k = \mathbf{y}_d - \mathbf{y}_k = \mathbf{y}_d - \boldsymbol{g}(\mathbf{u}) - \varepsilon_k$, where ε_k denotes the additive noise. Then, the RM algorithm can be applied to solve this problem. Moreover, due to the existence of additive noises, it is difficult to achieve precise tracking performance, thus we may consider some optimization objective such as $\mathbb{E}\|\mathbf{e}_k\|^2$. It is evident that $\mathbf{u}_d$ can minimize this optimization objective if the noises are zero-mean, independent with the system signals, and of additive form in $\mathbf{e}_k$. In this case, the KW algorithm or its variants can be applied to solve the problem. In short, the main procedures for stochastic approximation-based method are as follows: first, transform the ILC problem into a root-seeking or extrema-seeking problem of some unknown functions with the desired input being the root or extrema-argument; then, apply the RM or KW algorithms to complete the design and analysis steps. This approach in ILC was first proposed in [58], where a KW algorithm with random differences was applied.

We first explain the RM algorithm-based approach, taking the probabilistic quantization error example in Sec. 3.1 as an illustration. Consider system (1) and update law (22) with quantized outputs. Subtracting both sides of (22) from $u_d(t)$, we have

$$\begin{aligned}\delta u_{k+1}(t) = \delta u_k(t) - a_k[&L_t C_{t+1} B_t \delta u_k(t) + C_{t+1} A_t \delta x_k(t) \\ &+ r(y_k(t+1))].\end{aligned} \tag{31}$$

Applying the lifting technique to all variables, we have

$$\delta \mathbf{u}_{k+1} = \delta \mathbf{u}_k - a_k L H \delta \mathbf{u}_k - a_k L \mathbf{r}_k, \tag{32}$$

where $\mathbf{r}_k \triangleq [r(y_k(1))^T, \ldots, r(y_k(N))^T]^T$. It is clear that $\mathbb{E}[\mathbf{r}_k] = 0$ and $\mathbb{E}[\|\mathbf{r}_k\|^2] \leq qN/4$ with q and N being the output dimension and the iteration length. Taking a careful check to (32), it is evident that 0 is the single root of the function $\boldsymbol{g}(\delta\mathbf{u}) \triangleq LH\delta\mathbf{u}$ provided that we assume $C_{t+1}B_t$ to be of full-column rank and design L_t such that all eigenvalues of $L_t C_{t+1} B_t$ are with positive real parts. Then, (32) is a typical RM algorithm and the convergence conditions for the RM algorithm can be verified [54]. As a direct corollary, we can conclude that the sequence $\{\delta\mathbf{u}_k\}$ generated by (32) converges to zero almost surely (for details, we refer to [59]).

It is seen that the design of L_t requires prior information of the system when applying the RM algorithm. This requirement can be removed if the KW algorithm is applied [54]. In particular, consider the system (19) with both process disturbances and measurement noises. The control purpose for this system is to minimize the asymptotically averaged tracking errors index,

$$\limsup_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \|e_k(t)\|^2 = \min \quad \text{a.s.}, \quad \forall 1 \leq t \leq N,$$

where "a.s." is short for "almost surely". To solve this optimization problem, a vector sequence $\{\Delta(t,k)\}$ (independent of $w(t,k)$ and $v(t,k)$) is introduced. In particular, define $\Delta(t,k) = [\Delta_1(t,k), \ldots, \Delta_p(t,k)]^T$ as a p-dimensional vector 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

$$|\Delta_j(t,k)| < m, \left|\frac{1}{\Delta_j(t,k)}\right| < n, \mathbb{E}\frac{1}{\Delta_j(t,k)} = 0, \tag{33}$$

where m and n are positive constants. We denote

$$\overline{\Delta}(t,k) = \left[\frac{1}{\Delta_1(t,k)}, \ldots, \frac{1}{\Delta_p(t,k)}\right]^T. \tag{34}$$

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

$$a_k > 0, \quad a_k \underset{k\to\infty}{\longrightarrow} 0, \quad \sum_{k=0}^{\infty} a_k = \infty, \tag{35}$$

$$c_k > 0, \quad c_k \underset{k\to\infty}{\longrightarrow} 0, \quad \sum_{k=0}^{\infty} \left(\frac{a_k}{c_k}\right)^{1+\frac{\delta}{2}} < \infty, \tag{36}$$

$$M_k > 0, \quad M_{k+1} > M_k, \quad M_k \underset{k\to\infty}{\longrightarrow} \infty, \tag{37}$$

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,

$$u(t, 2k+1) = u(t, 2k) + c_k \Delta(t,k) \tag{38}$$

and

$$\overline{u}(t,2(k+1)) = u(t,2k) - a_k \frac{\overline{\Delta}(t,k)}{c_k}(\|e(t+1,2k+1)\|^2 - \|e(t+1,2k)\|^2) \quad (39)$$

$$u(t,2(k+1)) = \overline{u}(t,2(k+1)) \cdot \mathbf{1}_{[\|\overline{u}(t,2(k+1))\| \le M_{\sigma_k(t)}]} \quad (40)$$

$$\sigma_k(t) = \sum_{l=1}^{k-1} \mathbf{1}_{[\|\overline{u}(t,2(l+1))\| > M_{\sigma_l(t)}]}, \quad \sigma_0(t) = 0, \quad (41)$$

where $\mathbf{1}_{[\cdot]}$ 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.

Evidently, no information about system matrices are involved in the above algorithms. Only the tracking error is used to generate the input signal. As a consequence, the algorithm should involve additional gradient-estimation mechanism for searching the convergence direction, c.f., (39). It was strictly proved in [54] that the input sequence generated by the update algorithms (38)–(41) would converge to the desired input almost surely, provided that the coupling matrix $C_{t+1}B_t$ is of full-column rank.

Generally, the stochastic approximation-based method is advantageous in loose convergence conditions, little information requirement on the systems (implying wide applications), and strong convergence properties. The main limitation of such method is the slow convergence speed caused by the additionally introduced decreasing sequence $\{a_k\}$. However, we should point out that the decreasing sequence is necessary for addressing the additive random noises. If we replace the decreasing sequence with a fixed gain, the almost sure convergence is still ensured with the sacrifice that the convergence limit may deviate from the desired one. The deviation mainly depends on the random noises; therefore, if the random noises are with small fluctuation range, the final deviation for algorithms with fixed gain is still acceptable for practical applications. This point can be regarded as a trade-off between the tracking performance and convergence speed.

4. Multiplicative Randomness Case

In this section, we review the major techniques for systems with multiplicative randomness. The multiplicative randomness is usually caused by the imperfect communication channels. For examples, for fading channels in communications, the multiplicative randomness is introduced to describe the fading phenomenon, and for data dropouts in the networks due to link breaks and data congestion, they are also denoted by a random variable multiplied to the original signals. Although we separate the randomness by additive, multiplicative, and coupled types, the main methods for addressing the stochastic ILC problem are consistent. Therefore, the specific derivations for the specific treatments in the following may be concise as we have detailed them in Sec. 3.

4.1. *Examples of multiplicative randomness*

Example 5. (Random Data Dropouts). Networked control structure has been widely employed in many engineering implementations because this structure has high flexibility and robustness. In the configuration, the plant and the learning controller are located at different sites and communicate with each other through wired/wireless networks. While considering the communication networks, due to data congestion, limited bandwidth, and linkage faults, the data packet may be lost during transmission. Therefore, the data transmission has two alternative states: successful transmission and loss. In this case, the data dropout is generally described by a random binary variable, say $\gamma_k(t)$ for the data packet at time instant t of the kth iteration. In particular, $\gamma_k(t)$ is equal to 1 if the corresponding data packet is successfully transmitted, and 0 otherwise. Then, to model the random data dropout, we need to impose a mathematical formulation of the random variable $\gamma_k(t)$. The most common model for $\gamma_k(t)$ is the Bernoulli variable model. In particular, the variable $\gamma_k(t)$ is independent for different time instants t and iteration number k. Moreover, $\gamma_k(t)$ obeys a Bernoulli distribution with

$$\mathbb{P}(\gamma_k(t) = 1) = \overline{\gamma}, \quad \mathbb{P}(\gamma_k(t) = 0) = 1 - \overline{\gamma}, \quad (42)$$

where $\overline{\gamma} = \mathbb{E}\gamma_k(t)$ with $0 < \overline{\gamma} < 1$.

In this example, we consider data dropout occurring at the measurement side only; that is, the network from the plant to the controller suffers random data dropouts while the network from the controller back to the plant is assumed to work well. When the data packet is lost during the transmission, we have to propose a specific data compensation mechanism for the dropped data. For simplicity, if the output data is lost during transmission, we replace it with the desired reference signal. In such formulation, the update law (6) becomes

$$u_{k+1}(t) = u_k(t) + \gamma_k(t+1)L_t e_k(t+1). \quad (43)$$

In other words, if the output is successfully transmitted, then the tracking error is available for updating the input signal; otherwise, the output is lost during transmission, it is replaced with the desired signal and thus the actual used tracking error is zero. These two scenarios are integrated in (43). Clearly, the random variable $\gamma_k(t)$ is multiplicative to the original signals. The ILC for systems with random data dropouts has been a hot topic in the past few years [60–66].

Example 6. (Iteration-Varying Lengths). In the conventional ILC, we assume that the process operation should retain the same for each iteration so that we could learn from the previous experiences. However, in many applications, the operation may end before arriving at the desired length due to safety or large deviation. For example, it was reported in [67] that the functional electrical stimulation of the peroneal nerve is applied for stroke patients, where the patients' walk steps may be cut short by suddenly putting the foot down. This observation motivates a novel random iteration-varying length problem in ILC. The mathematical formulation of this problem using random variables was first given in [7] and later developed in a series of publications [8, 68–70]. Now, we brief the problem formulation as follows [8]. Since the iteration length is not identical for all iterations, without loss of generality, there must exist a length $N_{\min}$ such that all iteration length will exceed $N_{\min}$. Then, the actual trial length N_k for the kth iteration varies between $N_{\min}$ and N randomly, i.e., $N_{\min} \leq N_k \leq N$.[a] There are $N - N_{\min} + 1$ possible iteration lengths. Denote the probability that the trial length is of $N_{\min}$, $N_{\min} + 1, \ldots, N$ be $p_1, p_2, \ldots, p_m$, respectively, where $m = N - N_{\min} + 1$. That is, $\mathbb{P}(\mathscr{A}_{N_{\min}}) = p_1, \ldots, \mathbb{P}(\mathscr{A}_N) = p_m$, where $\mathscr{A}_l$ denotes the event that the iteration length is l. Obviously, $p_i > 0$ and $p_1 + p_2 + \cdots + p_m = 1$. Then, we could define a random variable $\gamma_k(t)$ denoting the event that the operation process can continue to the time instant t in the kth iteration or not, by letting $\gamma_k(t) = 1$ and 0, respectively. It is clear that $\mathbb{P}(\gamma_k(t) = 1) = \sum_{i=t+1-N_{\min}}^{m} p_i$. Based on these preparations, we can propose the P-type update law for ILC with randomly iteration-varying lengths [8, 69],

$$u_{k+1}(t) = u_k(t) + \gamma_k(t+1)L_t e_k(t+1). \tag{44}$$

In the earlier paper [7], an iteration-average operator was introduced, $\mathbf{A}\{f_k(\cdot)\} \triangleq \frac{1}{k+1}\sum_{i=0}^{k} f_i(\cdot)$ for a sequence $f_0(\cdot)$, …, $f_k(\cdot)$. The corresponding update law with iteration-average operator is given as follows:

$$u_{k+1}(t) = \mathbf{A}\{u_k(t)\} + \frac{k+2}{k+1} L_t \sum_{i=0}^{k} \gamma_i(t+1)e_i(t+1). \tag{45}$$

Both (44) and (45) introduce multiplicative randomness.

Example 7 (One-Iteration Communication Delay). Communication delay was also considered in the existing literature to explore the limitation of networked control systems. In [71, 72], one-iteration communication delay was studied, where the communication delay indicated the iteration-axis-based delay rather than time-axis-based delay. In particular, for the kth iteration, the received output comes from either the current iteration, $y_k(t)$, or the previous iteration, $y_{k-1}(t)$, randomly subject to 0–1 Bernoulli distribution. In other words,

$$\tilde{y}_k(t) = \alpha_k(t)y_k(t) + (1 - \alpha_k(t))y_{k-1}(t), \tag{46}$$

where $\tilde{y}_k(t)$ denotes the actually received signal and $\alpha_k(t)$ takes value from $\{0, 1\}$. That is, if $\alpha_k(t) = 1$, the current output $y_k(t)$ is received; otherwise $\alpha_k(t) = 0$, the previous output $y_{k-1}(t)$ is received. In this case, the randomness is of multiplicative type.

4.2. *Expectation based method*

For the multiplicative randomness case, the expectation-based method is common in the existing literature, which is usually applied incorporated with the α-norm technique (or equivalently the Gronwall lemma), due to its simplicity in eliminating the randomness [7, 62, 64, 66, 68–72].

In [7], the expectation-based method was applied to the update law (45) for linear system (1). By direct calculations, one is able to have

$$\mathbb{E}[\mathbf{A}\{\delta u_{k+1}(t)\}] = \mathbb{E}[\mathbf{A}\{\delta u_k(t)\}] - L\mathbb{E}[\mathbf{A}\{\gamma_k(t+1)e_k(t+1)\}],$$

in which we can easily obtain

$$\mathbb{E}[\mathbf{A}\{\gamma_k(t+1)e_k(t+1)\}] = p(t+1)\mathbb{E}[\mathbf{A}\{e_k(t+1)\}]$$

by the commutative property of the operators $\mathbb{E}[\cdot]$ and $\mathbf{A}\{\cdot\}$, where $p(t)$ denotes the probability that the operation process continues up to time instant t. Therefore, the randomness in the equation has been eliminated and the proof can be completed by the conventional contraction mapping method. Consequently, the convergence condition in [7] is $\sup_t \|I - p(t)LCB\| \leq \rho < 1$.

We should remark that there are two major operators in using the expectation-based method: the expectation operator for eliminating the randomness and the norm operator for generating a contraction mapping. Generally, we should take the expectation operator first and then apply the norm to the newly derived (deterministic) equation, as is done in [7]. In this case, one can only obtain the convergence in expectation sense; that is, $\lim_{k\to\infty}\mathbb{E}[e_k(t)] = 0$ or $\lim_{k\to\infty}\mathbb{E}[\delta u_k(t)] = 0$. Because the expectation operator and the norm operator are not commutative, it is hard to obtain stronger convergence conclusions such as $\lim_{k\to\infty}\mathbb{E}[\|e_k(t)\|] = 0$ except some special cases.

As we have explained in Sec. 3.2, the convergence in expectation sense is weak. It motivates us to consider how to achieve a stronger convergence. In [70–72], the objective

[a] Note that the actual trial length may exceed the desired length N. In this case, the signals after the time instant N are redundant and useless for updating. Thus, we regard this case as the standard trial length without loss of any generality.

$\lim_{k\to\infty}\mathbb{E}[\|e_k(t)\|]=0$ is achieved by imposing strong conditions. For example, consider the ILC problem with one-iteration communication delay [71, 72]. With the received signal $\tilde{y}_k(t)$ as shown in (46), the modified tracking error $y_d(t+1)-\tilde{y}_k(t)$ is used for updating. Moreover, the transmission of the generated input to the plant also suffers random one-iteration communication delay similar to (46). In the analysis, after substituting the detailed expressions of the related signals (which are very complex and thus omitted here for saving space), an inequality can be obtained by taking norm operators to both sides of the expanded update law similar to the conventional steps of contraction mapping method. In this inequality, the randomness exists and the expectation is then taken to both sides of the inequality. As a result, the strong convergence depends on a hard-to-check condition, which we quote from [71] as follows:

$$\rho=\rho_1+\rho_2+\rho_3<1,$$

where

$$\rho_1=\|\mathbb{E}[|I-\Gamma\Lambda_k D\Omega_k|]\|_1+\|\Gamma\|_1\,\bar{\alpha}\,\bar{\omega}\,\frac{K_g\|AB\|_1}{1-K_f},$$

$$\rho_2=\|\Gamma\|_1[\bar{\alpha}(1-\bar{\omega})+(1-\bar{\alpha})\bar{\omega}]\Phi,$$

$$\rho_3=\|\Gamma\|_1[(1-\bar{\alpha})(1-\bar{\omega})\Phi,\quad \Phi=\frac{K_g\|AB\|_1}{1-K_f}+\|D\|_1,$$

where Λ_k and Ω_k denote the random matrices constituted by the random communication delay variables $\alpha_k(t)$ and $\omega_k(t)$ of the output and input sides, $\Lambda_k=\mathrm{diag}\{\alpha_k(0),\dots,\alpha_k(N-1)\}$ and $\Omega_k=\mathrm{diag}\{\omega_k(0),\dots,\omega_k(N-1)\}$, $\bar{\alpha}$ and $\bar{\omega}$ denote the expectations of the corresponding delay variables $\alpha_k(t)$ and $\omega_k(t)$, A, B, and D are stacked matrices of the system information, K_f and K_g are positive Lipschitz constants of the involved nonlinear functions, and Γ is the stacked matrix of the learning gain matrices. The specific meanings of these notations refer to [71]. It is clear that the condition of ρ_1 is difficult to verify because of the coupling of expectations and matrix norm. In other words, although a strong convergence is obtained, the proposed conditions are impractical for applications.

To further facilitate applications, [69] investigated the specific conditions such that the expectation and norm operators are commutative. In particular, the P-type update law (44) is applied for the nonlinear system (2) with $B(x_k(t))\equiv B$ and $C_t\equiv C$ under the random iteration-varying length environments. Subtracting both sides of (44) from $u_d(t)$, substituting the expression of $e_k(t+1)$, taking Euclidean norm to both sides of the newly derived equation, and then taking expectations, we arrive at

$$\begin{aligned}\mathbb{E}[\|\delta u_{k+1}(t)\|]\le{}&\mathbb{E}[\|I-\gamma_k(t+1)LCB\|\|]\mathbb{E}[\delta u_k(t)\|]\\&+h_1\mathbb{E}[\|\gamma_k(t+1)LC\|]\mathbb{E}[\|\delta x_k(t)\|].\end{aligned}$$

To access verifiable conditions, we need to exchange the computation order of expectation and norm operators. To this end, the following technical lemma was proposed in [69].

Technical Lemma. Let η be a Bernoulli binary random variable with $\mathbb{P}(\eta=1)=\overline{\eta}$ and $\mathbb{P}(\eta=0)=1-\overline{\eta}$. M is a positive matrix. Then the equality $\mathbb{E}\|I-\eta M\|=\|I-\overline{\eta}M\|$ holds if and only if one of the following conditions is satisfied: (1) $\overline{\eta}=0$; (2) $\overline{\eta}=1$; and (3) $0<\overline{\eta}<1$ and $0<M\le I$.

With the help of this lemma, the convergence condition in [69] is to design learning gain matrix L satisfying that $0<LCB<I$. We should emphasize that such condition of L is somewhat conservative compared with the conventional condition of L. However, one should note that such conservative selection of L provides us considerable advantages: the convergence property is stronger and the occurrent probability of randomly varying lengths is not required.

In short, the expectation-based method has been widely studied for the multiplicative randomness case. If the expectation is first taken, the original relationships are turned into deterministic ones and then the traditional techniques can be applied. However, the convergence is weak (in expectation sense). If the norm operator is first taken, the original relationships are turned into inequalities, where the randomness is coupled internally. In this case, the verifiable conditions for practical applications are usually difficult to access.

4.3. *Kalman filtering based method*

The results of Kalman filtering-based method are few in the multiplicative randomness case. Earlier attempts are made by Ahn *et al.* for linear time-invariant systems with random data dropouts [60, 61], where the output data suffer random loss during the transmission from the plant to the controller. A random variable subject to 0–1 Bernoulli distribution is used to denote the event of data dropout or not (see Example 5).

In [60], a time-invariant version of (1) was taken into account; that is, $A_t\equiv A$, $B_t\equiv B$, and $C_t\equiv C$. The intermittent update law was adopted due to the random data dropout at the sensor side. That is, the following update law was investigated:

$$u_{k+1}(t)=u_k(t)+L_{t,k}\gamma_k(t+1)e_k(t+1),\tag{47}$$

where $L_{t,k}$ is the learning gain matrix similar to the one defined in Sec. 3.3 and $\gamma_k(t)$ denotes the data dropout variable given in Example 5. Thus, (47) is modified from (43). Similar to the derivations in [52], the following 2D system was established:

$$\begin{bmatrix} \delta u_{k+1}(t) \\ \delta x_k(t+1) \end{bmatrix} = \begin{bmatrix} I - \gamma_k(t+1)L_{t,k}CB & -\gamma_k(t+1)L_{t,k}CA \\ B & A \end{bmatrix} \begin{bmatrix} \delta u_k(t) \\ \delta x_k(t) \end{bmatrix} + \begin{bmatrix} \gamma_k(t+1)L_{t,k}C & L_{t,k} \\ -I & 0 \end{bmatrix} \begin{bmatrix} w_k(t+1) \\ v_k(t+1) \end{bmatrix}.$$

We still use the notation $X^+ = [(\delta u_{k+1}(t))^T \ (\delta x_k(t+1))^T]^T$ and derive the recursive formula of $L_{t,k}$ by minimizing the trace of $P^+ = \mathbb{E}[X^+(X^+)^T]$. As a result, the following computation recursions are derived:

$$L_{t,k} = \overline{\gamma} V_1 P_{t,k} V_2^T (\Pi_k)^{-1}, \tag{48}$$

where $V_1 = (I, 0)$, $V_2 = (CB, CA)$, $P_{t,k} = \mathbb{E}[XX^T]$ with $X = [(\delta u_k(t))^T \ (\delta x_k(t))^T]^T$, and Π_k is a positive-definite matrix associated with the state error covariance, input error covariance, and the covariance matrices of random noises (for detailed expressions, please refer to [60]). Similarly, a recursive computation of the input error covariance was also derived,

$$P_{u,t,k+1} = (I - \overline{\gamma} L_{t,k} CB) P_{u,t,k}. \tag{49}$$

Comparing the recursive algorithms of the multiplicative randomness case with those of the additive randomness case given in Sec. 3.3, we find that the major difference is the introduction of the average successful transmission rate $\overline{\gamma}$, which clearly demonstrates the inherent effect of random data dropouts. Generally, the smaller the average rate $\overline{\gamma}$ is, the lower effect the learning gain matrix $L_{t,k}$ can exhibit, and the slower the input error covariance $P_{u,t,k}$ converges to zero. That is, the whole framework of the recursive algorithms would reduce its efficiency as the data dropout rate increases.

4.4. *Stochastic approximation based method*

The stochastic approximation-based method can behave well in addressing the multiplicative randomness. It is potential in the next phase of research. To illustrate this point, we consider Example 5 again and revisit the update law (43). We can easily rewrite (43) as follows:

$$u_{k+1}(t) = u_k(t) + \overline{\gamma} L_t e_k(t+1) + [\gamma_k(t+1) - \overline{\gamma}] L_t e_k(t+1). \tag{50}$$

From this formulation, it is found that the former part $u_k(t) + \overline{\gamma} L_t e_k(t+1)$ coincides with the traditional RM algorithm because $\overline{\gamma}$ is just a positive scalar constant. The latter part $[\gamma_k(t+1) - \overline{\gamma}] L_t e_k(t+1)$ can be viewed as a random noise term with zero mean, because $\gamma_k(t+1)$ is independent of $e_k(t+1)$ and $\mathbb{E}[\gamma_k(t+1) - \overline{\gamma}] = 0$. Therefore, the convergence conditions of the RM algorithm [54] can be verified with slight assumptions on the system model and the learning gain matrix. In other words, the convergence for this multiplicative randomness is a simple corollary of the additive randomness case as we can transform the multiplicative randomness into additive randomness. Using the above transform, we can convert most multiplicative randomness problems. Thus, we omit tedious repetitions of other similar problems. We remark that the stochastic approximation-based method is a useful analysis tool for the multiplicative randomness case.

5. Coupled Randomness Case

In this section, we proceed to brief the progresses for systems with coupled randomness. Here, the coupled randomness indicates those randomness terms which cannot be clearly separated from the original equations as individual additive and multiplicative forms. For these types of randomness, there are few efficient methods for us and thus more novel methods are desiderated. In this section, we mainly present the stochastic approximation-based method, which was reported in recent literature, as a minnow to catch a whale.

5.1. *Examples of coupled randomness*

Example 8 (Successive Update Laws). In Example 5, we have clarified that random data dropout commonly occurs for the networked control implementations. A binary variable subject to Bernoulli distribution is adopted to describe the randomness. We introduce the intermittent update scheme in Example 5 (i.e., (43)), where the algorithm updates its input if and only if the corresponding output packet is received by the learning controller; otherwise, the algorithm will just retain its previous input information and wait for the next available packet. Under this scheme, the update frequency would be very slow if the average data transmission rate is rather low. In fact, the updating frequency is equal to the successful transmission rate. Scholars are motivated to propose novel schemes in which the algorithms update the input successively no matter whether the corresponding packet is received or not [73, 74]. In this example, we consider system (1) and provide the following successive update law:

$$u_{k+1}(t) = u_k(t) + L_t e_k^*(t+1), \tag{51}$$

where L_t is the learning gain matrix for adjusting the control direction, and $e_k^*(t+1)$ denotes the latest available tracking error:

$$e_k^*(t) = \begin{cases} e_k(t), & \text{if } \gamma_k(t) = 1 \\ e_{k-1}^*(t), & \text{if } \gamma_k(t) = 0 \end{cases}. \tag{52}$$

The inherent mechanism of successive update scheme is that the algorithm keeps updating by using the latest available packet. In other words, if the output of the last iteration is received, then the algorithm will update its input using this information. If the output of the last iteration is lost, then the algorithm will update its input using the latest available output packet received previously. The algorithm (51) can be rewritten as

$$\begin{aligned} u_{k+1}(t) &= u_k(t) + \gamma_k(t+1)L_t e_k(t+1) \\ &\quad + [1-\gamma_k(t+1)]L_t e_{k-1}^*(t+1). \end{aligned} \tag{53}$$

If the measurement output of the last iteration is lost during the transmission, then the one used in (51) will be unknown because of the possibility of successive data dropouts. Thus, update information can come from any previous iteration. Therefore, we introduce stochastic stopping times $\{\tau_k^t, k = 1, 2, \ldots, 0 \le t \le N\}$ to denote the random iteration-delays of the update. In other words, (51) can be reformulated as

$$u_{k+1}(t) = u_k(t) + L_t e_{k-\tau_k^{t+1}}(t+1), \tag{54}$$

where the stopping time $\tau_k^t \le k$. The essential update mechanism is as follows: for the updating at t of the $(k+1)$th iteration, no information of $e_m(t+1)$ with $m > k - \tau_k^{t+1}$ is received but only $e_{k-\tau_k^{t+1}}(t+1)$ is available. Therefore, for the iterations $k - \tau_k^{t+1} < m \le k$, the input $u_m(t)$ is updated using the same tracking error $e_{k-\tau_k^{t+1}}(t+1)$.

Paying attention to (54), we are clear that the randomness comes from the subscript of $e_{k-\tau_k^{t+1}}(t+1)$ (or specifically, τ_k^{t+1}) and thus it is coupled with the error information. Indeed, the coupling of stochastic stopping times and the successive update scheme make the convergence analysis more complex than that of the additive and multiplicative randomness cases.

Example 9 (Random Communication Asynchronism). Large-scale systems are commonly used in many industrial applications. By large-scale systems we mean that the whole system is composed of many subsystems which are internally connected. That is, the operation of each subsystem has certain influence on other subsystems and the inner influence is generally unknown [76]. To model the inner connection, we consider a large-scale systems consisting of n subsystems, where the state of the ith subsystem is denoted by $x_i(t,k)$ at time instant t of the kth iteration. Then, the state vector of the large-scale system is denoted by $x(t,k) = [x_1^T(t,k), \ldots, x_n^T(t,k)]^T$. The influence of all subsystems on the ith subsystem can be described by a general nonlinear function $f_i(t, x(t,k))$. Due to various random factors such as communication delay and transmission congestion, the actual received state information for the ith subsystem may come from older iterations of other subsystems. In other words, at the kth iteration, the actual inner dynamics for the ith subsystem is driven by the following state vector,

$$\bar{x}_i(t,k) = [x_1^T(t, k-\tau_{1i}(k)), \ldots, x_n^T(t, k-\tau_{ni}(k))]^T, \tag{55}$$

where $\tau_{ji}(k) > 0$ denotes the random communication delay for the ith subsystem at iteration k to receive information from the jth subsystem, while each subsystem receives information from itself without any delay, i.e., $\tau_{ii} = 0$. In other words, at the kth iteration the latest information from the jth subsystem obtained by the ith subsystem is $x_j^T(t, k-\tau_{ji}(k))$, and no information from $x_j(t,m)$ with $m > k - \tau_{ji}(k)$ can reach the ith subsystem. In this case, the randomness (i.e., communication asynchronism) is involved in the formulation of the state vector, and thus, it is difficult to separate the randomness as individual variables from the system signals.

Example 10. (Hammerstein–Wiener Stochastic Systems). In [75], the following Hammaerstein-Wiener system was considered, where both system disturbances and measurement noises were included as follows:

$$\begin{aligned} v_k(t) &= f_t(u_k(t)), \\ x_k(t+1) &= A_t x_k(t) + B_t v_k(t) + \epsilon_k(t+1), \\ z_k(t) &= C_t x_k(t) + \zeta_k(t), \\ y_k(t)\ s &= g_t(z_k(t)) + \xi_k(t), \end{aligned} \tag{56}$$

where $f_t(\cdot) : \mathbf{R}^p \to \mathbf{R}^p$ and $g_t(\cdot) : \mathbf{R}^q \to \mathbf{R}^q$ are the nonlinearities at the input (Hammerstein part) and output (Wiener part) sides, respectively. $\epsilon_k(t)$, $\zeta_k(t)$, and $\xi_k(t)$ are random noises. Clearly, due to the existence of the nonlinearities, the internal noises $\epsilon_k(t)$ and $\zeta_k(t)$ are coupled with the nonlinear functions. That is, these random noises cannot be separated from the system variables. It has been proved in [75] that the optimal input for a given reference according to the index $V_t = \limsup_{n\to\infty} \frac{1}{n} \times \sum_{i=k}^{n} \|y_d(t) - y_k(t)\|^2$ is not identical to the one computed from the same system without any noise. This result demonstrated the effect of the random noises in stochastic nonlinear systems. For this kind of systems, the approaches in the previous sections are no longer applicable.

5.2. *Stochastic approximation-based method*

There are few effective methods for addressing the coupled randomness. The expectation-based method fails to solve this problem because the mathematical expectation cannot be taken to the randomness directly, by noting that the involvement of the randomness in the system signals is complex and unknown. The Kalman filtering-based method has not been proved effective for this problem because the distribution assumptions on randomness are generally invalid. Unlike these methods, stochastic approximation only requires little information of the system structure and relaxed conditions on randomness, thus it would be a promising approach in the future.

To demonstrate the application of this method, we review the techniques in addressing the successive update scheme [73, 74]. Reconsidering the update law (54) and noting there are random noises in system (1), we add the decreasing step-size to (54):

$$u_{k+1}(t) = u_k(t) + a_k L_t e_{k-\tau_k^{t+1}}(t+1), \tag{57}$$

where $a_k > 0$, $\sum_{k=1}^{\infty} a_k = \infty$, and $\sum_{k=1}^{\infty} a_k^2 < \infty$. We observe that the main difficulty lies in the inner randomness τ_k^{t+1}, which cannot be transformed into individual form. To solve this difficulty, we make a qualitative estimation on the influence of this random variable first.

Note the assumption that the data dropout is subject to a generic Bernoulli distribution and thus the successive iteration number of data dropouts (i.e., τ_k^{t+1}) obeys the geometric distribution. To make the notations concise, let τ denote a random variable satisfying the same geometric distribution, i.e., $\tau \sim G(\overline{\gamma})$. Then, by simple calculations, we have $\mathbb{E}[\tau] = 1/\overline{\gamma}$ and $\mathrm{Var}(\tau) = (1-\overline{\gamma})/\overline{\gamma}^2$. It follows that $\mathbb{E}[\tau^2] = (\mathbb{E}[\tau])^2 + \mathrm{Var}(\tau) = (2-\overline{\gamma})/\overline{\gamma}^2$. Using direct calculations, we have

$$\begin{aligned}\sum_{n=1}^{\infty} \mathbb{P}(\tau \geq n^{1/2}) &= \sum_{n=1}^{\infty} \mathbb{P}(\tau^2 \geq n) \\ &= \sum_{j=1}^{\infty} j\mathbb{P}(j \leq \tau^2 < j+1) \leq \mathbb{E}\tau^2 < \infty.\end{aligned}$$

Incorporating with Borel–Cantelli lemma, we derive that $\mathbb{P}(\tau > n^{1/2},\ \text{i.o.}) = 0$ and consequently, $\tau_k^t/k \to 0$ almost surely as k increases to infinity. Essentially, the result indicates that the influence of the successive data dropouts along the iteration axis is asymptotically negligible as the iteration number increases.

On the basis of the above estimation and conclusion, the convergence analysis can be done by two steps. First, show the convergence of the following update law:

$$u_{k+1}(t) = u_k(t) + a_k L_t e_k(t+1), \tag{58}$$

using the basic stochastic approximation techniques. Second, complete the proof by verifying that the difference between (57) and (58), i.e., $e_k(t+1) - e_{k-\tau_k^{t+1}}(t+1)$, satisfies the noise conditions in the conventional RM algorithms. The details for this verification can be found in [73, 74].

6. Possible Future Directions

Stochastic ILC has gained more and more attention from both scholars and engineers, where suitable treatment of the unknown random variables is emphatically concerned. The convergence analysis of update laws with random signals is much different from the traditional ILC analysis approach. In particular, the analysis of stochastic ILC would involve much knowledge of probability theory and stochastic process. Moreover, the convergence should be expressed in certain probability senses such as expectation, mean-square, and almost sure senses. Further, the control objective for systems with various types of randomness may also be different from the conventional ILC problems. In sum, the investigation of stochastic ILC has its own distinction and requires novel techniques.

Currently, we are at the starting stage of stochastic ILC as we mainly obtain the primary results on the classical models. Even for the classical models, the integrated framework of synthesis and analysis of update laws are blank for most issues. Therefore, there are many open issues and topics in stochastic ILC. In consideration of recent progresses, we would like to emphasize the following research directions, which have shown their promising significance for further developments.

- The stochastic counterparts of various classical ILC topics are expected for contributions. For example, point-to-point control has become an important direction of ILC owing to its additional freedom of the tracking reference [77, 78]. In point-to-point control, only some desired points may be required to realize accurate tracking, while the others are not considered. This topic has been heavily studied for deterministic systems; however, few papers are found for systems with randomness such as stochastic noises [79, 80]. Thus, it is of importance to consider the stochastic point-to-point control problem. Similarly, we have contributed significant works on ILC with iteration-varying tracking references, ILC for multi-agent systems, decentralized/distributed ILC algorithms, and data-driven ILC algorithm design and analysis; however, the corresponding discussions with additional random signals and factors are seldom reported.
- The major analysis approaches for systems with randomness are still insufficient. As can be seen from the above overview, the expectation-based method, Kalman filtering-based method, and stochastic approximation-based method have been deeply explored. However, the expectation-based method aims to transform the relationships into deterministic type so that the conventional techniques can be applied. In this case, the random characteristic is neglected and thus cannot well describe the specific operation process. The Kalman filtering-based method mainly requires the system to be linear and with

Gaussian random signals. The stochastic approximation-based method generally exhibits a slow convergence speed due to the introduction of decreasing step-size sequence. Therefore, we believe novel synthesis and analysis approaches are of great value for promoting the developments of stochastic ILC.

- The comprehensive framework for solving any special problem is welcome. We have listed some typical examples of systems with additive, multiplicative, and coupled randomness. However, most of the mentioned examples have not been well resolved. In comparison with other examples, the random data dropout problem has gained much attention in recent years [60–64]. Various techniques have been provided from different perspectives. A systematic design and analysis framework for three data dropout models was reported in [65] based on the stochastic approximation techniques. For the other examples, systematic frameworks are still open.
- It is seen that both additive and multiplicative randomness cases have been deeply investigated and all three methods have been showing effectiveness in addressing different problems. However, for the coupled randomness case, the progresses are very limited. The additive randomness generally indicates the additional noises and disturbances. The multiplicative randomness generally indicates the network failure and system process failure. Both of them can be separated from the system signals themselves. For the coupled randomness, the random variable is included as part of the system signals and thus the conventional techniques fail to eliminate or transform the random factors or signals. Consequently, more novel and effective approaches are fairly expected for this case.
- In the literature, most results in stochastic ILC concentrate on the theoretical research and few works on the practical implementations are found. Indeed, various types of randomness exist in the practical systems, while most practical experiments adopt the conventional techniques for deterministic systems. Therefore, it is of great interest to examine the performance of stochastic ILC algorithms in applications and compare it with the deterministic learning algorithms. In consideration of practical implementations, more randomness may be involved such as sampling and quantization. We believe stochastic ILC can exhibit distinct performance and property when unpredictable signals are involved in the system operation.

Here, we only list part of points for stochastic ILC based on our vision, which may neglect some important directions unintentionally. We should remark that stochastic ILC is a broad topic in ILC as it includes various randomness models, various specific problems, and various treatment techniques. We expect that more attention can be paid to stochastic ILC from both scholars and engineers.

7. Conclusion

In this paper, we present a technical overview of the recent progresses on stochastic ILC. Unlike the existing surveys, we focus on the principles and applications of effective approaches for addressing the stochastic ILC problem. In particular, we first demonstrate the basic problem formulation of ILC to clarify the fundamental principles and then specify two major methods for deterministic systems: contraction mapping method and 2D system-based method. Next, we proceed to classify the possible stochastic ILC into three categories according to the position of random variables: additive randomness, multiplicative randomness, and coupled randomness. For the additive randomness, the kernel idea is to eliminate the random variables since they are added to the system signals, where the expectation-based method, Kalman filtering-based method, and stochastic approximation-based method have shown their distinct advantages in different angles. For the multiplicative randomness, the kernel idea is to transform the original form into a randomness-free formulation or additive randomness formulation. All three methods are analyzed in sequence with emphasis on the comparisons with additive randomness case. For the coupled randomness, limited results have been reported and we mainly highlight the stochastic approximation-based method. Last, we have presented promising directions for the future research. It should be mentioned that this paper tries to present a technical tutorial for the reader to quickly understand the common problems of stochastic ILC and widely-applied techniques for the problems, thus we have not tried to seek as many related papers as possible and we may have missed some important papers. We expect more publications on this attractive subject will be realized in the future.

Acknowledgments

This paper is dedicated to the late Professor Jian-Xin Xu, an IEEE Fellow, a leading expert in various fields of systems and control, and an excellent model with his passion and dedication to science and engineering for us. The author has discussed deeply with Prof. Xu on various topics of systems and control and greatly inspired by his insightful directions during the academic visiting from February 2016 to February 2017. His professionalism will be forever remembered.

This work is supported by National Natural Science Foundation of China (61673045).

References

[1] J.-X. Xu and J. Xu, 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.

[2] J.-X. Xu and R. Yan, On initial conditions in iterative learning control, *IEEE Trans. Autom. Control* **50**(9) (2005) 1349–1354.

[3] M. Sun and D. Wang, Iterative learningcontrol with initial rectifying action, *Automatica* **38**(7) (2002) 1277–1282.

[4] M. Sun and D. Wang, Initial shift issues on discrete-time iterative learning control with system relative degree, *IEEE Trans. Autom. Control* **48**(1) (2003) 144–148.

[5] D. Meng and K. L. Moore, Robust iterative learning control for non-repetitive uncertain systems, *IEEE Trans. Autom. Control* **62**(2) (2017) 907–913.

[6] D. Meng and K. L. Moore, Convergence of iterative learning control for SISO nonrepetitive systems subject to iteration-dependent uncertainties, *Automatica* **79** (2017) 167–177.

[7] X. Li, J.-X. Xu and D. Huang, An iterative learning control approach for linear systems with randomly varying trial lengths, *IEEE Trans. Autom. Control* **59**(7) (2014) 1954–1960.

[8] D. Shen, W. Zhang, Y. Wang and C.-J. Chien, On almost sure and mean square convergence of P-type ILC under randomly varying iteration lengths, *Automatica* **63**(1) (2016) 359–365.

[9] Z.-B. Wei, Q. Quan and K.-Y. Cai, Output feedback ILC for a class of nonminimum phase nonlinear systems with input saturation: An additive-state-decomposition-based method, *IEEE Trans. Autom. Control* **62**(1) (2017) 502–508.

[10] Z. Hou, J. Yan, J.-X. Xu and Z. Li, Modified iterative-learning-control-based ramp metering strategies for freeway traffic control with iteration-dependent factors, *IEEE Trans. Intell. Transp. Syst.* **13**(2) (2012) 606–618.

[11] Z. Li, Y. Hu and D. Li, Robust design of feedback feed-forward iterative learning control based on 2D system theory for linear uncertain systems, *Int. J. Syst. Sci.* **47**(11) (2016) 2620–2631.

[12] M. Uchiyama, Formulation of high-speed motion pattern of a mechanical arm by trial, *Trans. Soc. Instrum. Control Eng.* **14**(6) (1978) 706–712.

[13] S. Arimoto, S. Kawamura and F. Miyazaki, Bettering operation of robots by learning, *J. Robotic Syst.* **1**(2) (1984) 123–140.

[14] K. L. Moore and J.-X. Xu (Guest Editors), Special issue on iterative learning control, *Int. J. Control* **73**(10) (2000) 819–999.

[15] C. T. Freeman and Y. Tan (Guest Editors), Special issue on iterative learning control and repetitive control, *Int. J. Control* **84**(7) (2011) 1193–1294.

[16] Special issue on iterative learning control, *Asian J. Control* **4**(1) (2002) 1–118.

[17] H.-S. Ahn and K. L. Moore (Guest Editors), Special issue on iterative learning control, *Asian J. Control* **13**(1) (2011) 1–212.

[18] Y. Wang (Guest Editor), Special issue on latest updates of iterative learning control and their applications, *J. Proc. Control* **24**(12) (2014) 62–124.

[19] D. A. Bristow, M. Tharayil and A. G. Alleyne, A survey of iterative learning control: A learning-based method for high-performance tracking control, *IEEE Control Syst. Mag.* **26**(3) (2006) 96–114.

[20] H.-S. Ahn, Y. Q. Chen and K. L. Moore, Iterative learning control: Survey and categorization from 1998 to 2004, *IEEE Trans. Syst. Man Cybern. Part C* **37**(6) (2007) 1099–1121.

[21] Y. Wang, F. Gao and F. J. Doyle, III, Survey on iterative learning control, repetitive control and run-to-run control, *J. Proc. Control* **19** (10) (2009) 1589–1600.

[22] J.-X. Xu, A survey on iterative learning control for nonlinear systems, *Int. J. Control* **84**(7) (2011) 1275–1294.

[23] D. Shen and Y. Wang, Survey on stochastic iterative learning control, *J. Proc. Control* **24**(12) (2014) 64–77.

[24] K. L. Moore, *Iterative Learning Control Control for Deterministic Systems* (Advances in Industrial Control, Springer-Verlag, 1993).

[25] Y. Q. Chen and C. Wen, *Iterative Learning Control: Convergence, Robustness and Applications* (LNCIS, Springer-Verlag, London, 1999).

[26] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control* (LNCIS, Springer, New York, 2003).

[27] H.-S. Ahn, K. L. Moore and Y. Q. Chen, *Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems* (Communications and Control Engineering Series, Springer Verlag, 2007).

[28] C. T. Freeman, E. Rogers, J. H. Burridge, A.-M. Hughes and K. L. Meadmore, *Iterative Learning Control for Electrical Stimulation and Stroke Rehabilitation* (Springer-Verlag, London, 2015).

[29] D. H. Owens, *Iterative Learning Control: An Optimization Paradigm* (Advances in Industrial Control, Springer-Verlag, 2016).

[30] J.-X. Xu, S. K. Panda and T. H. Lee, *Real-time Iterative Learning Control: Design and Applications* (Advances in Industrial Control, Springer-Verlag London, 2009).

[31] D. Wang, Y. Ye and B. Zhang, *Practical Iterative Learning Control with Frequency Domain Design and Sampled Data Implementation*, Advances in Industrial Control (Springer, Singapore, 2014).

[32] S. Yang, J.-X. Xu, X. Li and D. Shen, *Iterative Learning Control for Multi-Agent Systems Coordination* (Wiley, 2017).

[33] D. Shen, *Iterative Learning Control with Passive Incomplete Information: Algorithm Design and Convergence Analysis* (Springer, Singapore, 2018).

[34] S. S. Saab, Optimality of first-order ILC among higher order ILC, *IEEE Trans. Autom. Control* **51**(8) (2006) 1332–1336.

[35] S. N. Huang, K. K. Tan and T. H. Lee, Necessary and sufficient condition for convergence of iterative learning algorithm, *Automatica* **38**(7) (2002) 1257–1260.

[36] K. K. Tan, S. N. Huang, T. H. Lee and S. Y. Lim, A discrete-time iterative learning algorithm for linear time-varying systems, *Engineering Applications of Artificial Intelligence* **16** (2003) 185–190.

[37] D. Shen and Y. Wang, Iterative learning control for networked stochastic systems with random packet losses, *Int. J. Control* **88**(5) (2015) 959–968.

[38] D. Shen and Y. Wang, ILC for networked nonlinear systems with unknown control direction through random lossy channel, *Syst. Control Lett.* **77** (2015) 30–39.

[39] D. Shen, C. Zhang and Y. Xu, Intermittent and successive ILC for stochastic nonlinear systems with random data dropouts. *Asian J. Control* (2018) doi: 10.1002/asjc.1480.

[40] Z. Geng and M. Jamshidi, Learning control system analysis and design based on 2D system theory, *J. Intell. Robot. Syst.* **3**(1) (1990) 17–26.

[41] J. E. Kurek and M. B. Zaremba, Iterative learning control synthesis based on 2-D system theory, *IEEE Trans. Autom. Control* **38**(1) (1993) 121–125.

[42] S. S. Saab, A discrete-time learning control algorithm for a class of linear time-invariant systems, *IEEE Trans. Autom. Control* **40**(6) (1995) 1138–1141.

[43] T. Kaczerek, *Two-Dimensional Linear Systems* (Springer-Verlag, Germany, 1985).

[44] E. Rogers, K. Galkowski and D. H. Owens, *Control Systems Theory and Applications for Linear Repetitive Processes* (Springer-Verlag, Berlin, Heidelberg, 2007).

[45] B. Altin and K. Barton, Exponential stability of nonlinear differential repetitive processes with applications to iterative learning control, *Automatica* **81** (2017) 369–376.

[46] J. Bolder and T. Oomen, Inferential iterative learning control: A 2D-system approach, *Automatica* **71** (2016) 247–253.

[47] D. Meng, Y. Jia, J. Du and F. Yu, Robust learning controller design for MIMO stochastic discrete-time systems: An H_∞-based approach, *Int. J. Adapt. Control Signal Proc.* **25**(7) (2011) 653–670.

[48] M. Butcher, A. Karimi and R. Longchamp, A statistical analysis of certain iter-ative learning control algorithms, *Int. J. Control* **81**(1) (2008) 156–166.

[49] R. E. Kalman, A new approach to linear filtering and prediction problems, *J. Basic Eng.* **82**(1) (1960) 35–45.

[50] S. S. Saab, A discrete-time stochastic learning control algorithm, *IEEE Trans. Autom. Control* **46**(6) (2001) 877–887.
[51] S. S. Saab, On a discrete-time stochastic learning control algorithm, *IEEE Trans. Autom. Control* **46**(8) (2001) 1333–1336.
[52] S. S. Saab, Stochastic P-type/D-type iterative learning control algorithms. *Int. J. Control* **76**(2) (2003) 139–148.
[53] S. S. Saab, A stochastic iterative learning control algorithm with application to an induction motor, *Int. J. Control* **77**(2) (2004) 144–163.
[54] H. F. Chen, *Stochastic Approximation and Its Applications* (Dordrecht, The Netherlands: Kluwer, 2002).
[55] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control* (Wiley, 2003).
[56] H. Robbins and S. Monro, A stochastic approximation method, *Ann. Math. Stat.* **22**(3) (1951) 400–407.
[57] J. Kiefer and J. Wolfowitz, Stochastic estimation of the maximum of a regression function, *Ann. Math. Stat.* **23**(3) (1952) 462–466.
[58] H. F. Chen, Almost sure convergence of iterative learning control for stochastic systems, *Sci. China (Series F)* **46**(1) (2003) 67–79.
[59] D. Shen and J.-X. Xu, Zero-error tracking of iterative learning control using probabilistically quantized measurements, in *Proc. the 2017 Asian Control Conf.* (Gold Coast, Australia, December 17–20, 2017), pp. 1029–1034.
[60] H.-S. Ahn, Y. Q. Chen and K. L. Moore, Intermittent iterative learning control, in *Proc. 2006 IEEE Int. Symp. Intelligent Control* (Munich, Germany, October 4–6, 2006), pp. 832–837.
[61] H.-S. Ahn, K. L. Moore and Y. Q. Chen, Discrete-time intermittent iterative learning controller with independent data dropouts, in *Proc. 2008 IFAC World Congress* (Seoul, Korea, July 6–11, 2008), pp. 12442–12447.
[62] X. Bu, F. Yu, Z. Hou and F. Wang, Iterative learning control for a class of nonlinear systems with random packet losses, *Nonlinear Anal. Real World Appl.* **14**(1) (2013) 567–580.
[63] X. Bu, Z. Hou, S. Jin and R. Chi, An iterative learning control design approach for networked control systems with data dropouts, *Int. J. Robust Nonlinear Control* **26** (2016) 91–109.
[64] J. Liu and X. Ruan, Networked iterative learning control design for nonlinear systems with stochastic output packet dropouts, *Asian J. Control* **20**(3) (2018) 1077–1087.
[65] D. Shen and J.-X. Xu, A framework of iterative learning control under random data dropouts: Mean square and almost sure convergence, *Int. J. Adapt. Control Signal Process.* **31**(12) (2017) 1825–1852.
[66] D. Shen, Iterative learning control with incomplete information: A survey, *IEEE/CAA J. Autom. Sin.* **5**(5) (2018) 885–901.
[67] T. Seel, C. Werner, J. Raisch and T. Schauer, Iterative learning control of a drop foot neuroprosthesis — Generating physiological foot motion in paretic gait by automatic feedback control, *Control Eng. Pract.* **48** (2016) 87–97.
[68] X. Li, J.-X. Xu and D. Huang, Iterative learning control for nonlinear dynamic systems with randomly varying trial lengths, *Int. J. Adapt. Control Signal Process.* **29**(11) (2015) 1341–1353.
[69] D. Shen, W. Zhang and J.-X. Xu, Iterative learning control for discrete nonlinear systems with randomly iteration varying lengths, *Syst. Control Lett.* **96** (2016) 81–87.
[70] L. Wang, X. Li and D. Shen, Sampled-data iterative learning control for continuous-time nonlinear systems with iteration-varying lengths, *Int. J. Robust Nonlinear Control* **28**(8) (2018) 3073–3091.
[71] J. Liu and X. Ruan, Networked iterative learning control approach for nonlinear systems with random communication delay, *Int. J. Syst. Sci.* **47**(16) (2016) 3960–3969.
[72] J. Liu and X. Ruan, Networked iterative learning control design for discrete-time systems with stochastic communication delay in input and output channels, *Int. J. Syst. Sci.* **48**(9) (2017) 1844–1855.
[73] D. Shen, C. Zhang and Y. Xu, Two compensation schemes of iterative learning control for networked control systems with random data dropouts, *Inf. Sci.* **381** (2017) 352–370.
[74] D. Shen, C. Zhang and Y. Xu, Intermittent and successive ILC for stochastic nonlinear systems with random data dropouts, *Asian J. Control* **20**(3) (2018) 1102–1114.
[75] D. Shen and H.-F. Chen, A Kiefer-Wolfowitz algorithm based iterative learning control for Hammerstein-Wiener systems, *Asian J. Control* **14**(4) (2012) 1070–1083.
[76] D. Shen and H.-F. Chen, Iterative learning control for large scale nonlinear systems with observation noise, *Automatica* **48**(3) (2012) 577–582.
[77] C. T. Freeman, Z. Cai, E. Rogers and P. L. Lewin, Iterative learning control for mul-tiple point-to-point tracking application, *IEEE Trans. Control Syst. Technol.* **19**(3) (2011) 590–600.
[78] C. T. Freeman and Y. Tan, Iterative learning control with mixed constraintsfor point-to-point tracking, *IEEE Trans. Control Syst. Technol.* **21**(3) (2013) 604–616.
[79] Y. Xu, D. Shen and X.-D. Zhang, Stochastic point-to-point iterative learning control based on stochastic approximation, *Asian J. Control* **19**(5) (2017) 1748–1755.
[80] D. Shen, J. Han and Y. Wang, Stochastic point-to-point iterative learning tracking without prior information on system matrices, *IEEE Trans. Autom. Sci. Eng.* **14**(1) (2017) 376–382.

Dong Shen received the B.S. degree in mathematics from School of Mathematics, Shandong University, Jinan, China, in 2005. He received the Ph.D. degree in mathematics from the Key Laboratory of Systems and Control, Institute of Systems Science, Academy of Mathematics and System Science, Chinese Academy of Sciences (CAS), Beijing, China, in 2010.

From 2010 to 2012, he was a Post-Doctoral Fellow with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, CAS. From 2016.02 to 2017.02, he was a visiting scholar at National University of Singapore (NUS), Singapore. Since 2012, he has been with the College of Information Science and Technology, Beijing University of Chemical Technology (BUCT), Beijing, China, where he now is a Professor.

His current research interests include iterative learning control, stochastic control and optimization. He has published more than 70 refereed journal and conference papers. He is author of *Stochastic Iterative Learning Control* (Science Press, 2016, in Chinese) and *Iterative Learning Control with Passive Incomplete Information: Algorithm Design and Convergence Analysis* (Springer, 2018), co-author of *Iterative Learning Control for Multi-Agent Systems Coordination* (Wiley, 2017), and co-editor of *Service Science, Management and Engineering: Theory and Applications* (Academic Press and Zhejiang University Press, 2012). Dr. Shen received IEEE CSS Beijing Chapter Young Author Prize in 2014 and Wentsun Wu Artificial Intelligence Science and Technology Progress Award in 2012.