Discrete-time Stochastic Iterative Learning Control: A Brief Survey*

Dong Shen and Gang Xiong
State Key Laboratory of Management and Control for Complex Systems
Beijing Engineering Research Center of Intelligent Systems and Technology
Institute of Automation, CAS, Beijing 100190, P.R. China
{*dong.shen, gang.xiong*}@*ia.ac.cn*

***Abstract*— This note gives a brief survey on discrete-time stochastic iterative learning control (SILC) from three aspects, namely, SILC for linear system, nonlinear system and system with other stochastic signal. Two major approaches, stochastic Kalman filtering approach and stochastic approximation approach, for SILC are proposed. Some open questions are also included.**

***Index Terms*— Discrete-time Stochastic System, Iterative Learning Control, Linear System, Nonlinear System.**

I. Introduction

In our lives, it appears that if we work on some task repeatedly, it would perform better and better. This basic cognition motivates the research on iterative learning control (ILC). ILC is normally designed for those systems that could complete some task over a fixed time interval and perform them repeatedly. In such systems, the input and output information of past cycles, as well as the tracking objective, are used to formulate the input signal for the next cycle, so that the tracking performance could be improved as the number of cycles increases to infinity.

Taking a comparison between ILC and real life experience, we find that the input/output information of past cycles is equivalent to the experience of life. As is known, we usually make a new policy when facing the same task based on the past experience. This new policy is equivalent to the input signal applied to the next cycle. As past experience could make us handle affairs better, the operating information of past cycles may be able to improve the control performance in ILC.

Since ILC could learn from past cycles, a rather simply updated control algorithm may achieve a high quality of tracking. As a matter of fact, ILC has benefits that it requires little system information, but the algorithm is still effective [1]–[3].

In this note, we briefly review the recent issues on stochastic iterative learning control (SILC). Here by SILC we mean ILC for those systems containing stochastic signals, such as system noise and measurement noise, which are described random variable of probability theory. These signals are with both uncertainty and randomness, usually unbounded. According to different system models and stochastic signal types, the SILC issues reviewed later consists of three parts,

- SILC for linear system with system noise and/or measurement noise;
- SILC for nonlinear system with system noise and/or measurement noise;
- SILC for system with other stochastic signals such as networked control system with random data dropout and large-scale system with asynchronous update.

The rest of the note is arranged as follows. Section 2 discusses the linear system case, while Section 3 is for the nonlinear system case. The SILC for other types of stochastic signals is addressed in Section 4. Some concluding remarks are given in Section 5.

II. SILC for Linear System

Consider the following discrete-time linear system model

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

where k denotes different cycles, $k = 1, 2, \cdots$ and t denotes an arbitrary time in an operation cycle, $t = 0, 1, \cdots, N$. For notation convenience, by $t \in [0, N]$ we mean t value from 0 to N. $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $y \in \mathbb{R}^q$ are state, input and output, respectively. $A(t)$, $B(t)$, and $C(t)$ are time-varying system matrices with appropriate dimensions. $w(t,k)$ and $v(t,k)$ denote system noise and measurement noise, respectively.

Saab first discusses the SILC for discrete-time linear system (1) in [4], and improves the analysis techniques in the following research [5]–[7]. The update law in [4] is

$$u(t,k+1) = u(t,k) + K(t,k)[e(t+1,k) - e(t,k)] \tag{2}$$

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

In order to give recursive algorithm for the computation of $K(t,k)$, some assumptions are needed. First, the tracking objective $y(t,d)$ is realizable, i.e. for appropriate initial state $x(0,d)$, there exists a unique input $u(t,d)$ generating the

*This work is supported in part by NSFC #61174172, #70890084, #60921061, #90920305; CAS #2F09N05, #2F09N06, #2F10E08, #2F10E10; MOST: #2011AA110502.

trajectory for nominal plant. In other words, the following equations hold,

$$\begin{aligned} x(t+1,d) &= A(t)x(t,d) + B(t)u(t,d) \\ y(t,d) &= C(t)x(t,d) \end{aligned} \tag{3}$$

Besides, the input/output coupling matrix $C(t+1)B(t)$ is assumed full-column rank. In the following, a pre-notation δ is used to denote the difference of desired value and actual value, i.e. $\delta x(t,k) \triangleq x(t,d) - x(t,k)$, $\delta u(t,k) = u(t,d) - u(t,k)$. $\mathbb{E}(\cdot)$ denotes mathematical expectation.

The assumptions on noise and initial state given in [4] are as follows.

Noise and Initial State Assumptions of [4]: $w(t,k)$ and $v(t,k)$ are assumed to be zero-mean white Gaussian noise such that $Q_t = \mathbb{E}[w(t,k)w(t,k)^T]$ is positive-semidefinite matrix, $R_t = \mathbb{E}[v(t,k)v(t,k)^T]$ is positive-definite matrix for all k. Besides, $w(t,k)$ is uncorrelated with $v(s,l)$, $\forall t,s \in [0,N], k,l \in \mathbb{R}$. The initial state satisfies that $\delta x(0,k)$ is also zero-mean white noise whose covariance matrix $P_{x,0} = \mathbb{E}[\delta x(0,k)x(0,k)^T]$ is positive-semidefinite. Moreover, $\delta x(0,k)$ is uncorrelated with all noises.

In addition, [4] also gives conditions on initial input $u(t,0)$. The initial input error $\delta u(t,0)$ is zero-mean white noise, and $\mathbb{E}[\delta u(t,0)\delta u(t,0)^T] = P_{u,0}$ is a symmetrical positiveCdefinite matrix.

Remark 1: The author points out that one simple scenario for the above conditions is to set $\delta u(t,0) \equiv 0$, $\forall t$. However, it is quite difficult to meet this scenario when the system information is completely unknown. In fact, the essence idea of ILC is to achieve good input signal by repeated learning from the past cycles for arbitrary initial input. On the other hand, $\delta u(t,0) \equiv 0$ means that the initial input has been chosen to be the desired input $u(t,d)$. If so, there is no need to update the input anymore, because $u(t,d)$ is actually good enough from some point of view. The latter publications [5]–[7], [9] of Saab also use the same assumptions.

Under these conditions, the recursive algorithm for $K(t,k)$ is derived by minimizing input error covariance matrix $P_{u,k+1}$ in Saab's approach. That is, the derivation of the trace of $P_{u,k+1}$ according to $K(t,k)$ is set to zero, i.e. $\mathrm{d}(\mathrm{trace}(P_{u,k+1}))/\mathrm{d}K(t,k) \equiv 0$. This leads to the following

$$K(t,k) = P_{u,k}\Xi^T(\Xi P_{u,k}\Xi^T + \Lambda_{D,k})^{-1} \tag{4}$$

$$P_{u,k+1} = (I - K(t,k)\Xi)P_{u,k} \tag{5}$$

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

Theorem 1 ([4]): For system (1) and control update laws (2) (4) (5), if $C(t+1)B(t)$ is full-column rank, then $\forall k,t$, there exists suitable norm $\|\cdot\|$ such that $\|I - K(t,h)C(t+1)B(t)\| < 1$. Consequently, $\|P_{u,k+1}\| < \|P_{u,k}\|$. Furthermore, $P_{u,k} \to 0$, $K(t,h) \to 0$ uniformly in $[0,N]$ as $k \to \infty$.

Let us call this approach as Kalman filtering approach. From Theorem 1 it is seen the algorithm of learning gain matrix $K(t,k)$ could guarantee the convergence of input sequence in the mean square sense. However, a lot of information is required by the algorithm, including system matrices $A(t)$, $B(t)$, $C(t)$, noise covariance Q_t, R_t, and state error covariance $P_{x,t}$ (see the formulation of $\Lambda_{D,k}$).

In order to remove the requirement on $P_{x,t}$, [5] proposed a revised algorithm,

$$\widetilde{K}(t,k) = \widetilde{P}_{u,k}\Xi^T(\Xi\widetilde{P}_{u,k}\Xi^T + \widetilde{\Lambda}_D)^{-1} \tag{6}$$

$$\widetilde{P}_{u,k+1} = (I - \widetilde{K}(t,k)\Xi)\widetilde{P}_{u,k} \tag{7}$$

where $\widetilde{\Lambda}_D = C(t+1)Q_tC(t+1)^T + R_t + R_{t+1}$. Note that $P_{x,t}$ is removed from the above algorithm now. It is proved in [5] that for the update law (2) with learning gain matrix $K(t,k)$ given by (6) (7), similar convergence results of Theorem 1 still hold when $C(t+1)B(t)$ is full-column rank.

It should be pointed out that $\widetilde{P}_{u,k}$ is an constructed matrix for the algorithm and it is not the real input error covariance matrix. Denote the real input error covariance generated by (2) (6) (7) as $\overline{P}_{u,k}$. It is shown that the convergence characteristics of $\widetilde{P}_{u,k}$ are equivalent to $\overline{P}_{u,k}$. That is, $\widetilde{P}_{u,k} \to 0 \Leftrightarrow \overline{P}_{u,k} \to 0$, and the convergence rates of $\widetilde{P}_{u,k}$ and $\overline{P}_{u,k}$ are both inversely proportional to the iteration index k [5].

Thus the convergence proposition is still valid although the state error covariance matrix is removed from the algorithm. However, in some sense, the input generated by the revised algorithm is not optimal. Thus for expression convenience, the update law (2) with (4) (5) containing the state error covariance matrix is called optimal learning algorithm, while the one with (6) (7) is called suboptimal learning algorithm.

Noticing that both [4] and [5] consider the D-type update law (2), a natural question arises: does these results hold for P-type update law? The answer is yes [6].

Consider the following P-type ILC law

$$u(t,k+1) = u(t,k) + K(t,k)e(t+1,k) \tag{8}$$

Completely similar derivation to [4], [5], it is easy to obtain the optimal learning algorithm for $K(t,k)$,

$$K(t,k) = P_{u,k}\Xi^T(\Xi P_{u,k}\Xi^T + \Lambda_{P,k})^{-1} \tag{9}$$

$$P_{u,k+1} = (I - K(t,k)\Xi)P_{u,k} \tag{10}$$

where $\Lambda_{P,k} = C(t+1)A(t)P_{x,k}(C(t+1)A(t))^T + C(t+1)Q_tC(t+1)^T + R_{t+1}$. While the suboptimal one is

$$\widetilde{K}(t,k) = \widetilde{P}_{u,k}\Xi^T(\Xi\widetilde{P}_{u,k}\Xi^T + \widetilde{\Lambda}_P)^{-1} \tag{11}$$

$$\widetilde{P}_{u,k+1} = (I - \widetilde{K}(t,k)\Xi)\widetilde{P}_{u,k} \tag{12}$$

where $\Lambda_P = C(t+1)Q_tC(t+1)^T + R_{t+1}$. It is proved in [6] that the above P-type optimal and suboptimal algorithms could ensure the same convergence characteristics to [4], [5] under suitable conditions.

So far, we already have four kinds of algorithms classified into P-type and D-type. Someone may ask: Is there any relationship of these P-type and D-type algorithms? In fact, by comparing (4)(5), (6)(7), (9)(10) and (11)(12) it is found that the essence difference among these algorithms is the selection of the bounded positive-definite matrix Λ, which is selected as $\Lambda_{D,k}$, $\widetilde{\Lambda}_D$, $\Lambda_{P,k}$ and $\widetilde{\Lambda}_P$, respectively. The algorithm would guarantee the same convergence properties as long as the matrix Λ is bound, symmetric and positive-definite [6]. This reveals the equivalence of the former four kinds of Saab-type algorithms.

Notice that the input/output coupling matrix $C(t+1)B(t)$ is all required to be full-column rank in [4]–[6]. To meet the requirement, the dimension of input should be no larger than the dimension of output. What is the case when the dimension of input is larger? [7] presents a first step for the following system

$$\begin{aligned} x(t+1,k) &= A(t)x(t,k) + Bu(t,k) \\ y(t,k) &= C(t)x(t,k) + v(t,k) \end{aligned} \tag{13}$$

There are two differences between (13) and (1), one of which is there is no system noise $w(t+1,k)$ in (13), while the other one is that $B(t) = B$, $\forall t \in [0, N]$. Besides, the system is assumed accurately re-initialized, i.e. $\delta x(0,k) = 0$, $\forall k$. The convergence results are included in the following theorem.

Theorem 2 ([7]): For system (13) and P-type optimal learning algorithm (8) (9) (10) or P-type suboptimal learning algorithm (8) (11) (12), if $C(t+1)B$ is full-row rank and $\delta x(0,k) = 0$, then there exists some suitable constant c such that $\|\mathbb{E}[z(t,k)z(t,k)^T]\| < \frac{c}{k}$, $\lim_{k\to\infty}\mathbb{E}[z(t,k)z(t,k)^T] = 0$, where $z(t,k) = C(t+1)\delta x(t,k)$.

To recap, Saab proposes optimal and suboptimal learning algorithms based on the idea minimizing the trace of input error covariance matrix, and proves the convergence in the mean square sense in [4]–[7]. However, it is required in all the algorithms to have the information of covariance of noises and coupling matrix $C(t+1)B(t)$ as prior knowledge. This may limit the application of these algorithms when the system information is unknown.

Chen proposes a learning algorithm with probability 1 convergence property in [8] for the stochastic system (1), where the system information is wiped off. Different from [4]–[7], the control objective is to minimize the averaged tracking error:

$$\limsup_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} \|y(t+1,k) - y(t+1,d)\|^2 = \min \quad a.s. \tag{14}$$

In order to take a comparison, here we list the assumptions on noise and initial state first.

Noise and Initial State Assumptions of [8]: The noises $w(t,k)$ and $v(t,k)$ are mutually independent with zero mean and finite but unknown moments of order $2+\delta$, where $\delta > 0$. Q_t and R_t are defined as previously mentioned but unknown here. The initial state $x(0,k)$ is mutually independent of both $w(t+1,k)$ and $v(t,k)$ with $\mathbb{E}x(0,k) = x_0$, $\mathbb{E}\|x(0,k)\|^{2+\delta} < \infty$. The covariance matrix is also assumed unknown.

The author of [8] provides a novel ILC algorithm based on stochastic approximation [13]. Because all system matrices $A(t)$, $B(t)$, $C(t)$, as well as noise covariance matrices Q_t, R_t, are unknown, it demands an estimation of the update direction. For this purpose, we construct vector sequences $\{\Delta(t,k)\}$ which is independent of noises $w(t,k)$, $v(t,k)$. $\Delta(t,k) = [\Delta_1(t,k), \cdots, \Delta_p(t,k)]^T$ is a p-dimension vector, whose components $\Delta_j(t,k)$ are mutually independent and identically distributed random variables satisfying that $\forall k = 1, 2, \cdots$, $t \in [0, N-1]$, $j = 1, \cdots, p$

$$|\Delta_j(t,k)| < m, \ |\frac{1}{\Delta_j(t,k)}| < n, \ \mathbb{E}\frac{1}{\Delta_j(t,k)} = 0 \tag{15}$$

where m, n are positive constants. Denote

$$\overline{\Delta}(t,k) = \left[\frac{1}{\Delta_1(t,k)}, \cdots, \frac{1}{\Delta_p(t,k)}\right]^T \tag{16}$$

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

$$a_k > 0, \quad a_k \xrightarrow[k\to\infty]{} 0, \quad \sum_{k=0}^{\infty} a_k = \infty \tag{17}$$

$$c_k > 0, \quad c_k \xrightarrow[k\to\infty]{} 0, \quad \sum_{k=0}^{\infty}\left(\frac{a_k}{c_k}\right)^{1+\frac{\delta}{2}} < \infty \tag{18}$$

$$M_k > 0, \quad M_{k+1} > M_k, \quad M_k \xrightarrow[k\to\infty]{} \infty \tag{19}$$

where δ is defined in noise assumptions. The initial input $u(t,0)$, $t \in [0, N]$ could be arbitrarily given. Then the update laws are defined according even iterations and odd iterations, respectively. For the odd cycle of operation the input is defined as follows

$$u(t, 2k+1) = u(t, 2k) + c_k\Delta(t,k) \tag{20}$$

while for the even cycle

$$\begin{aligned} \overline{u}(t, 2(k+1)) =& u(t, 2k) - a_k\frac{\overline{\Delta}(t,k)}{c_k}(\|e(t+1, 2k+1)\|^2 \\ & - \|e(t+1, 2k)\|^2) \end{aligned} \tag{21}$$

$$u(t, 2(k+1)) = \overline{u}(t, 2(k+1)) \cdot I_{[\|\overline{u}(t,2(k+1))\| \le M_{\sigma_k(t)}]} \tag{22}$$

$$\sigma_k(t) = \sum_{l=1}^{k-1} I_{[\|\overline{u}(t,2(l+1))\| > M_{\sigma_l(t)}]}, \quad \sigma_0(t) = 0 \tag{23}$$

where $I_{[\cdot]}$ denotes an indicator function meaning that it equals 1 if the condition indicated in the bracket is fulfilled, and 0 otherwise.

It is worthy to point out that the algorithm (20)-(23) is the Kiefer-Wolfowitz (KW) algorithm with expanding truncations [13]. KW algorithm is a basic type of stochastic approximation algorithm. In this algorithm, stochastic difference of adjacent cycles is used to estimate the update direction, while a_k is the update step.

Theorem 3 ([8]): For system (1) and algorithm (20)-(23), if $C(t+1)B(t)$ is full-column rank, then the input sequence $\{u(t,k)\}$ converges to the desired input $u(t,d)$ and minimizes the index (14).

The significance of [8] is providing an SILC approach based on stochastic approximation, while the latter requires much less information about the system matrices and noise covariance matrices. This coincides with the major advantage of ILC that it requires little information about system but improves performance by learning in some sense.

III. SILC for Nonlinear System

A. Affine Nonlinear System

As a kind of special nonlinear system, affine nonlinear system usually is the starting point. It contains the nonlinear attribute, but the output depends on the input, in essence, in a linear way. Actually, some good SILC results have been obtained according to affine nonlinear system [9], [10].

The following affine system model is used in [9],

$$\begin{aligned} x(t+1,k) &= f(x(t,k)) + B(x(t,k))u(t,k) \\ y(t,k) &= C(t)x(t,k) + v(t,k) \end{aligned} \tag{24}$$

where $f(\cdot)$ denotes a vector function defined on $\mathbb{R}^n$. It is obvious that the system equation is time-invariant, i.e. $f(\cdot)$, $B(\cdot)$ only depends on the state vector. The relative degree is set 1 here for notation clear and expression convenience.

Comparing the conditions with [4], [7], the tracking objective $y(t,d)$ is also assumed to be realizable and coupling matrix $C(t+1)B(x)$ is full-column rank or full-row rank. The assumptions on noise, initial state, and initial input are the same with [4]. The major difference lie in the nonlinearities $f(\cdot)$ and $B(\cdot)$, which are allowed to grow as fast as any polynomial with arbitrary order in [9]. Thus the usual requirement of global Lipschitz condition is relaxed.

Besides, $y(t,k)$ is called measurement output in [9], while the actual output signal that tracks $y(t,d)$ is $C(t)x(t,k)$. Thus $e(t,k) = y(t,d) - y(t,k)$ is called output measurement error, and $\delta\psi(t,k) \triangleq C(t)[x(t,d) - x(t,k)]$ denotes the output tracking error. The control objective in [9] is generating input sequence such that the output tracking error converges to zero in the mean square sense and trajectories are bounded.

The SILC update law is designed in P-type (8). By similar but more delicate derivation to [6], [7], the computational algorithm for learning gain matrix $K(t,k)$ could be formulated like (11) (12). The algorithm guarantees asymptotic zero-convergence of input error covariance matrix, whose rate is inversely proportional to the number of iteration index k.

However, the major contribution of [9] is not proposing an ILC algorithm with its convergence analysis, but presenting the necessary and sufficient conditions for boundedness of trajectories and for zero convergence of output tracking error under the measurement noise. These conditions could be regarded as kind of guidance for the selection of suitable learning gain matrix $K(t,k)$. All such conditions depends only on the learning gain $K(t,k)$ and coupling matrix $C(t+1)B(x)$.

Remark 2: It would be specifically mentioned that the boundedness of [9] is defined on basis of mathematical expectation. In particular, by boundedness of $x(t,k)$ the author actually means $\mathbb{E}[\delta x(t,k)\delta x(t,k)^T]$ is bounded. However, strictly speaking, boundedness of covariance matrix may just hint the second moment of some random variable is bounded, while the random variable itself may be unbounded.

Chen and Fang have also considered the SILC for affine nonlinear system [10], which is a further investigation of [8]. The system model contains both system noise and measurement noise, described as follows:

$$\begin{aligned} x(t+1,k) &= f(t,x(t,k)) + B(t,x(t,k))u(t,k) \\ &\qquad + w(t+1,k) \\ y(t,k) &= C(t)x(t,k) + v(t,k) \end{aligned} \tag{25}$$

where $f(t,x)$ and $B(t,x)$ are time-varying functions. The control objective also is to minimize the index (14).

The assumptions are given here. For any $t \in [0,N]$, $f(t,x)$ and $B(t,x)$ are continuous in x, and bounded by some polynomial function. That is, there exist real numbers r, s and l such that $\|f(t,x)\| + \|B(t,x)\| \leq r\|x\|^l + s$. The input is no more than the output, i.e. $p \leq q$, and the coupling matrix $C(t+1)B(t,x)$ is full-column rank, $\forall x \in \mathbb{R}^n$. The conditions on noises and initial states are the same as those in [8] with more additional moment requirements, $\mathbb{E}\|w(t,k)\|^r < \infty$, $\mathbb{E}\|x(0,k)\|^r < \infty$, $\forall r \in \mathbb{Z}^+$. Moreover, all random variables are assumed to be i.i.d. along the iteration index, $\forall t$.

Denote $P(t,x) \triangleq B^T(t,x)C^T(t+1)C(t+1)B(t,x)$, then $P(t,x)$ is positive-definite. For the tracking target $y(t,d)$, the optimal input $u^0(t)$ is first inductively formulated in [10] which minimizes the index (14). Let $x^0(0,k) \equiv x(0,k)$, then define $u^0(t)$ and $x^0(t,k)$ in turn along the time index as follows,

$$\begin{aligned} u^0(t) &= -[\mathbb{E}P(t,x^0(t,k))]^{-1} \\ &\qquad \times \{\mathbb{E}[B^T(t,x^0(t,k))C^T(t+1)f(t,x^0(t,k))] \\ &\qquad - \mathbb{E}[B^T(t,x^0(t,k))]C^T(t+1)y(t+1,d)\} \end{aligned} \tag{26}$$

$$\begin{aligned} x^0(t+1,k) &= f(t,x^0(t,k)) + B(t,x^0(t,k))u^0(t) \\ &\qquad + w(t+1,k) \end{aligned} \tag{27}$$

It is proved that $u^0(t)$ is optimal in the sense minimizing (14), and further any input sequence $\{u(t,k)\}$ satisfying $u^0(t) - u(t,k) \xrightarrow[k\to\infty]{} 0$, $\forall t$, is also optimal. Thus the actual objective now is to construct an algorithm generating input sequence that converges to $u^0(t)$.

ILC update algorithm is designed as (20)-(23), where the parameters are given by (15)-(19). The authors prove that the input sequence define by this algorithm would converge to $u^0(t)$ with probability one, and thus is optimal. However, due to limited space, only sketch of the proof is provided.

Note that the output depends on the input in a linear way either in [9] or [10]. What is the case where the relationship is nonlinear in essence? This will be given in the next subsection.

B. Non-affine Nonlinear System

One of the difficulties dealing with non-affine nonlinear system with stochastic noise may lie in the strong coupling of random variables and nonlinear functions. Shen and Chen make an attempt on this topic in [11], where the nonlinear is simplified as typical nonlinearities, such as deadzone, preload and saturation.

The SISO system is described as

$$\begin{aligned} &x(t+1,k) = f(t,x(t,k)) + b(t,x(t,k))\eta(t,k)\\ &\eta(t,k) = \mathcal{N}(u(t,k))\\ &y(t,k) = c(t)x(t,k) + v(t,k) \end{aligned} \tag{28}$$

where $\eta(t,k)$ is unknown intermediate signal. $\mathcal{N}$ denotes nonlinear function, including deadzone, preload and saturation. These three nonlinearities are common in industrial systems and make the relationship of the input and the output be nonlinear in essence.

The tracking target is a realizable signal $y(t,d)$ and the control objective is to minimize the index (14). The nonlinear functions $f(t,x)$ and $b(t,x)$ are allowed to grow up not faster than a polynomial as the state x diverges. The measurement noise $v(t,k)$ is mutually along the iteration index with zero mean and finite second moment. The initial state is assumed asymptotically accurate, i.e. $\delta x(0,k) \xrightarrow[k\to\infty]{} 0$. Besides, the symbol of $c^+b_k(t)$ is required prior known, and denote it by $\text{sgn}(c^+b_k(t))$.

A unified algorithm is used for three different input nonlinearities,

$$\begin{aligned} u(t,k+1) = {} & [u(t,k) + a_k \text{sgn}(c^+b_k(t))e(t+1,k)]\\ & \times I_{[|u(t,k)+a_k\text{sgn}(c^+b_k(t))e(t+1,k)| \le M_{\sigma_k(t)}]} \end{aligned} \tag{29}$$

$$\sigma_k(t) = \sum_{i=1}^{k-1} I_{[|u(t,i)+a_i\text{sgn}(c^+b_i(t))e(t+1,i)| > M_{\sigma_i(t)}]} \tag{30}$$

$$\sigma_0(t) = 0 \tag{31}$$

where a_k, M_k are defined by (17) and (19). The input sequence generated by the algorithm is bounded and optimal almost surely.

On this basis, the nonlinearity is allowed to appear not only at the input side but also at the output side in [12]. Moreover, the nonlinearity is of general form not restricted to a specific class as mentioned above. Specifically, the system is established as

$$\begin{aligned} \eta(t,k) = {} & f(t,u(t,k))\\ x(t+1,k) = {} & A(t)x(t,k) + B(t)\eta(t,k)\\ & + \varepsilon(t+1,k)\\ z(t,k) = {} & C(t)x(t,k) + \epsilon(t,k)\\ y(t,k) = {} & g(t,z(t,k)) + v(t,k) \end{aligned} \tag{32}$$

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

For this system, under suitable conditions, the authors prove that the input sequence generated by (20)-(23) will converge almost surely to the optimal control minimizing the index (14).

The results [10], [12] show that the KW algorithm based SILC algorithm behaves some advantages when dealing with nonlinear stochastic system. However, the continuity requirements on nonlinear functions are one of the restrictions waiting for a break-through.

IV. SILC for System with Other Stochastic Signal

A. Data Dropout

With the development of network technologies, networked control system (NCS) has been increasing in applications. This kind of system combines sensors, actuators, and controllers by network, thus enhances the flexibility and reliability of the system. However, the problem of data dropout in NCS may reduce the performance. There have been some research on ILC for NCS with data dropout [14]–[16].

Ahn et al propose develop SILC algorithms by the same techniques as [4] for NCS with data dropout. They also obtain convergence in the mean square sense. To make the statement clear, let us first repeat the formulation of data dropout.

Take the data dropout at the channel that transmits output measurement to the control center in consideration. If data dropout happens, then there is no output signal transmitted back, thus the input could not be updated. Let $\gamma e(t,k)$ be the new tracking error, where $\gamma \in \{0,1\}$ denote whether data dropout happens or not. In other words, $\gamma = 0$ means data dropout happens, while $\gamma = 1$ means no data dropout. Since data dropout is a random event, γ is a random variable with Bernoulli distribution. Denote $\overline{\gamma} = \mathbb{E}\gamma$.

Consider the time-invariant case of (1), i.e. $A(t) \equiv A$, $B(t) \equiv B$, $C(t) \equiv C$. The ILC update law adopted by [14] is

$$u(t,k+1) = u(t,k) + K(t,k)\gamma e(t+1,k) \tag{33}$$

Completely similar derivation to [4], [6], [7], it is easy to give the recursive computational algorithm for $K(t,k)$. Under the learning gain matrix, the input error covariance matrix would converge to zero as long as $\overline{\gamma} \neq 0$. In other words, as long as the data are not missed by 100%, the similar convergence properties to Saab's work still hold with a lower rate.

Note that in [14] a data packet is assumed either whole missed during the network transmission or delivered successfully. However, for a q-dimension output, maybe only part information is lost while the others are still safely delivered during a transmission. [15] considers this case for a special class of (1), i.e. $p = q$ and time-invariant. The update law (33) is revised into

$$u(t, k+1) = u(t, k) + K(t, k)\Gamma e(t+1, k) \qquad (34)$$

where $\Gamma = diag[\gamma_i]$ and $\gamma_i \in \{0, 1\}$, $i = 1, \cdots, p$ with Bernoulli distribution. Similar analysis techniques to [4], [6], [14] leads to similar convergence result.

So far only the data dropout of error signals is considered [14], [15]. Following the same route, the above research is expanded to the case where there could be data dropout in both error signals and control signals [16]. However, the authors does not provide a computational algorithm for learning gain matrix as is done in [14], [15].

B. Stochastic Asynchronism

Many industrial systems are large-scale system, where by large-scale system we mean that the whole system is composed of many subsystems which are connected via the large state vector but each subsystem is controlled on the basis of its own input and output information. Due to different efficiencies among subsystems, data missing and/or communication delay, it is hard to achieve synchronous updates of all subsystems. This motivates the research on stochastic asynchrony update problem for ILC.

A class of discrete-time large scale systems with nonlinearly connected subsystems, each of which is affine nonlinear, and the observation equations are with noises, are considered in [17]. The formulation of each subsystem is the SISO case of (24). The control objective of each subsystem is to minimize the averaged tracking error (14).

The stochastic asynchrony of large scale system are assumed as follows. Denote by $S_k \subset \{1, \cdots, n\}$ the set of those subsystems which are updated at the kth iteration and denote by $\tau(i, k)$ the number of control updates occurred up to and including the kth iteration in system i: $\tau(i, k) \triangleq \sum_{j=1}^{k} I_{[i \in S_j]}$. It is assumed S_k and $\tau(i, k)$ are random variables in [17]. Combined with these notations, an asynchronous distributed ILC algorithm is proposed based on asynchronous stochastic approximation.

In order to guarantee the almost surely convergence of the input sequence to the optimal control, each subsystem should update frequently enough: there exist integer K large enough such that $\forall k, i$, $\tau(i, k+K) - \tau(i, k) > 0$. Note that it is only required the existence of K rather than specific value. How to further relax this condition is still a open problem.

V. Conclusions

ILC is an excellent control method applied widely in many batch-type industrial fields. Since introduced it has drawn much attentions from researchers and engineers in the past decades. However, there are only a few publications on stochastic ILC. This note briefly reviews some recent literature on stochastic ILC from three aspects, namely, SILC for linear system, SILC for nonlinear system, and SILC for system with other stochastic signals. Some comparisons of related research as well as some future research directions are provided. Due to limited space, some publications are not included in this note. A more detailed analysis and comparison is given in a coming paper.

References

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

[2] H.-S. Ahn, Y.Q. Chen and Kevin L. Moore. Iterative Learning Control: Survey and Categorization from 1998 to 2004. IEEE Trans. System Man and Cybernetics Part C, 37(6):1099-1121, 2007.

[3] Y. Wang, F. Gao and F.J. Doyle III. Survey on iterative learning control, repetive control and run-to-run control. Journal of Process Control, 19(10): 1589-1600, 2009.

[4] S.S. Saab. A discrete-time stochastic learning control algorithm. IEEE Trans. Automatic Control, 46(6): 877-887, 2001.

[5] S.S. Saab. On a discrete-time stochastic learning control algorithm. IEEE Trans. Automatic Control, 46(8): 1333-1336, 2001.

[6] S.S. Saab. Stochastic P-type/D-type iterative learning control algorithms. International Journal of Control, 76(2): 139-148, 2003.

[7] S.S. Saab. A stochastic iterative learning control algorithm with application to an induction motor. International Journal of Control, 77(2): 144-163, 2004.

[8] H.F. Chen. Almost sure convergence of iterative learning control for stochastic systems. Science in China (Series F), 46(1): 67-79, 2003.

[9] S.S. Saab. Selection of the learning gain matrix of an iterative learning control algorithm in presence of measurement noise. IEEE Trans. Automatic Control, 50(11): 1761-1774, 2005.

[10] H.F. Chen, H.T. Fang. Output tracking for nonlinear stochastic systems by iterative leanring control. IEEE Trans. Automatic Control, 49(4): 583-588, 2004.

[11] D. Shen, H.F. Chen. Iterative learning control for a class of nonlinear systems. Journal of System Science and Mathematical Science, 28(9): 1053-1064, 2008.

[12] D. Shen, H.F. Chen. A Kiefer-Wolfowitz algorithm based iterative learning control for Hammerstein-Wiener systems. Asian Journal of Control, accepted for publication.

[13] H.F. Chen. Stochastic Approximation and Its Applications. Dordrecht, the Netherlands: Kluwer, 2002.

[14] H.S. Ahn, Y.Q. Chen, K.L. Moore. Intermittent iterative learning control. Proc. the 2006 IEEE Int. Symposium on Intelligent Control, Munich, Germany, Oct. 4-6, 2006, pp. 832-837.

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

[16] H.S. Ahn, K.L. Moore, Y.Q. Chen. Stability of discrete-time iterative learning control with random data dropouts and delayed controlled signals in networked control systems. Proc. the IEEE Int. Conf. Control Automation, Robotics, and Vision, Dec. 2008.

[17] D. Shen, H.F. Chen. Iterative learning control for large scale nonlinear systems with observation noise. Automatica, accepted for publication.