Journal of Process Control xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Journal of Process Control

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

Survey on stochastic iterative learning control

Dong Shen*, Youqing Wang*

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

ARTICLE INFO

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

Keywords:
Iterative learning control
Stochastic signals
Linear systems
Nonlinear systems

ABSTRACT

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

1. Introduction

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

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

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

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

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

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

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

http://dx.doi.org/10.1016/j.jprocont.2014.04.013
0959-1524/© 2014 Published by Elsevier Ltd.

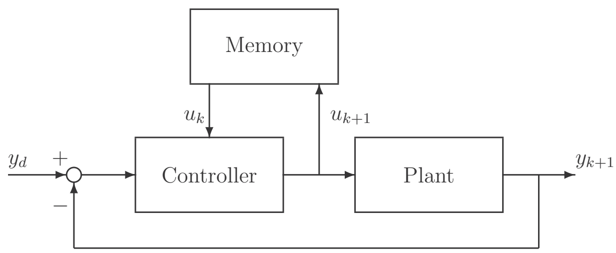

Fig. 1. Framework of ILC.

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

1.1. Background of ILC

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Denote

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

then one has

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

where

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

Similar to (8) and (9), define

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

then it leads to

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

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

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

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

Remark 2. Note that the focus of ILC is how to improve the tracking performance iteratively along the iteration index, as one can