Nonlinear Analysis: Real World Applications 14 (2013) 567–580

Contents lists available at SciVerse ScienceDirect

Nonlinear Analysis: Real World Applications

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

# Iterative learning control for a class of nonlinear systems with random packet losses

Xuhui Bu [a,b,*], Fashan Yu [a], Zhongsheng Hou [c], Fuzhong Wang [a]

[a] School of Electrical Engineering & Automation, Henan Polytechnic University, Jiaozuo, 454000, China

[b] Henan Provincial Open Lab. for Control Engineering Key Discipline, Henan Polytechnic University, Jiaozuo, 454000, China

[c] Advanced Control Systems Lab., School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing, 100044, China

ARTICLE INFO

*Article history:*
Received 8 March 2012
Accepted 12 July 2012

*Keywords:*
Iterative learning control
Nonlinear systems
Networked control systems
Packet dropout

ABSTRACT

This paper considers the problem of iterative learning control (ILC) for a class of nonlinear systems with random packet dropouts. It is assumed that an ILC scheme is implemented via a networked control system (NCS), and that during the packet transfer between the remote nonlinear plant and the ILC controller packet dropout occurs. A new formulation is employed to model the packet dropout case, where the random dropout rate is transformed into a stochastic parameter in the system's representation. Through rigorous analysis, it is shown that under some given conditions, the iterative learning control can guarantee the convergence of the tracking error although some packets are missing. The analysis is also supported by a numerical example.



## 1. Introduction

Iterative learning control (ILC) is an effective technique for systems with repetitive nature, which attempts to refine the desired control signal through repeated trials to achieve a perfect tracking of the desired output signal over a fixed time interval. This technique has been the center of interest of many researchers over the last two decades [1–6]. Recently, robustness has been studied in ILC from a number of different perspectives, such as model uncertainty [7–9], the initial condition and desired trajectory uncertainty [10–12], disturbance uncertainty and stochastic noise [13–15], and time delay uncertainty [16–18], etc.

On the other hand, networked control systems (NCSs) have been the focus of several research studies over the last few years [19–24]. Compared to conventional point-to-point system connections, using an NCS has advantages in installation, wiring, and maintenance cost and time. In an NCS, data travel through the communication channels from the sensors to the controller and from the controller to the actuators. The network can be modeled as a switch that opens and closes in a random manner. When a switch is open, its output is held at the previous value and the data packet is lost. Data packet dropout can occur due to node failures or network congestion and is a common problem in networked systems. In real-time control systems, it is normally advantageous to discard the old packets and consider the new ones so that the controller always receives fresh data for control calculation. Packet dropouts usually occur randomly. Because of random packet dropouts, classical estimation and control methods cannot be used directly in NCS systems.

It is still an open research area in ILC when the implementation of ILC in a networked systems setting, except for certain pioneer works that address linear systems with random packet dropout [25–28]. In [25,26], an optimal ILC controller is designed for a class of linear systems with random packet dropouts. The proposed ILC controller can compensate the packet

* Corresponding author at: School of Electrical Engineering & Automation, Henan Polytechnic University, Jiaozuo, 454000, China.
*E-mail address:* buxuhui@gmail.com (X. Bu).

1468-1218/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.nonrwa.2012.07.017

dropout effectively in the iteration domain. In [27], the convergence of first-order and high-order ILC for linear system with packet dropout is considered. Using the lifting technique or super vector transformation, such an ILC system can be modeled as an asynchronous dynamical system with rate constraints on events in the iteration domain, and then the convergence condition can be provided by solving a binary linear matrix inequality. It is worth noting that the binary linear matrix inequality is difficult to solve, especially for the high order ILC scheme. To avoid the problem, [28] give another convergence condition in the expectation sense.

This paper considers the ILC for a class of nonlinear systems with random packet dropouts. As depicted in previous works [25–28], there are two different types of packet dropouts in ILC systems: control input signal dropouts and output measurement signal dropouts. The first one occurs when the control input is updated, during the signal transfer from the controller to the plant the signal may be miss by actuator failure, network failure or data collision. The second packet dropout is the measurement data loss which is also called intermittent measurement. During the signal transfer from the sensor to the controller the signal may be missed due to the sensor failure or network failure. In this paper, the two kinds of packet dropouts are both considered.

The remainder of this paper is organized as follows. In Section 2, the problem formulation and some preliminaries are described. In Section 3, the single-input-single-output (SISO) system with packet dropout is considered, and the convergence analysis of the ILC scheme is given. The result of the SISO system is extended to a class of multi-input–multi-output (MIMO) systems in Section 4. In Section 5, an illustrative example is employed to verify the theoretical result, and conclusions follow in Section 6.

## 2. Problem formulation

Consider the following SISO nonlinear system

$$\begin{cases} x_k(t+1) = f(x_k(t)) + b(x_k(t))u_k(t), \\ y_k(t) = g(x_k(t)) + d(x_k(t))u_k(t), \end{cases} \tag{1}$$

where $k$ denotes the $k$th repetitive operation of the system, $t$ is the discrete time index and $t \in [0, N]$, which means that $t = \{0, 1, 2, \ldots, N\}$. $x_k(t) \in R^1$, $u_k(t) \in R^1$, $y_k(t) \in R^1$ are the state, control and output of the system, respectively.

The following assumptions for the system are first made in this paper.

**Assumption 1.** The nonlinear function $f(\cdot), b(\cdot), g(\cdot)$ and $d(\cdot)$ are uniformly globally *Lipschitz* in $x$, i.e. $\forall t \in [0, N]$, $\exists$ constants $k_f, k_b, k_c, k_d$, such that

$$\begin{aligned} &|f(x_1(t)) - f(x_2(t))| \le k_f\, |x_1(t) - x_2(t)|\,, \\ &|b(x_1(t)) - b(x_2(t))| \le k_b\, |x_1(t) - x_2(t)|\,, \\ &|g(x_1(t)) - g(x_2(t))| \le k_g\, |x_1(t) - x_2(t)|\,, \\ &|d(x_1(t)) - d(x_2(t))| \le k_d\, |x_1(t) - x_2(t)|\,, \end{aligned}$$

for any pair $(x_1(t), x_2(t))$ in $R \times R$.

**Assumption 2.** The resetting condition is satisfied for all the iteration, i.e.

$$x_k(0) = x_d(0),$$

where $k$ is the iteration number, $x_d(0)$ is the initial value of the desired state.

**Assumption 3.** For a desired trajectory $y_d(t)$, it exists $u_d(t)$ and $x_d(t)$ satisfying

$$\begin{cases} x_d(t+1) = f(x_d(t)) + b(x_d(t))u_d(t), \\ y_d(t) = g(x_d(t)) + d(x_d(t))u_d(t), \end{cases}$$

where $u_d(t)$ is the desired input and $x_d(t)$ is the desired state.

The setup of the control system (1) is illustrated as in Fig. 1. The sensor, actuator and the nonlinear system are remotely controlled by an iterative learning controller that interchanges measurement output and control input signals through a communication network. In the sensor and controller sides, they are time driven and in the actuator side it is event-driven. As shown in Fig. 1, the network can be modeled as a switch that opens and closes in a random manner. When a switch is open, its output is held at the previous value and the data packet is lost. The system output $y_k(t)$ is passed through the network and there may be random dropouts. Thus, the current observation $\tilde{y}_k(t)$ is the controller received output, which is the system output $y_k(t)$ with the probability of $\bar{\alpha}$. In the case of no new data, previous data will be used, so the previous data, $y_k(t-1)$, will be used with the probability of $1 - \bar{\alpha}$. Similarly, the plant input $u_k(t)$ is the current controller output,