

# Iterative Learning Control Based on Random Variance Reduction Gradient Method



Yihua Gao, Dong Shen, and Jiayi Qian

**Abstract** Traditional iterative learning control (ILC) algorithms usually assume that full system information and operation data can be utilized. However, due to the uncertainty and complexity of actual systems, it is difficult to access full system information and operation data accurately and completely. In this chapter, a novel ILC scheme based on stochastic variance reduced gradient (SVRG) is proposed. This scheme is not only suitable for resolving the incomplete information problem, but also converges efficiently under both strongly convex and non-strongly convex control objectives. To demonstrate the advantages, this chapter studied two scenarios, i.e., random error data dropout and model-free data-driven approach, and proposed two SVRG-based ILC algorithms for these two scenarios, respectively. It is theoretically demonstrated and experimentally verified that the proposed SVRG-based ILC scheme converges faster than both the full gradient and stochastic gradient methods for the two involved scenarios.

**Keywords** Iterative learning control · Variance reduction gradient method · ILC algorithms

---

Y. Gao  
Renmin University of China, Beijing, China  
e-mail: [gaoyihua@ruc.edu.cn](mailto:gaoyihua@ruc.edu.cn)

D. Shen (✉)  
Distributed Artificial Intelligence Laboratory, Renmin University of China, Beijing, China  
e-mail: [dshen@ruc.edu.cn](mailto:dshen@ruc.edu.cn)

J. Qian  
School of Mathematics, Renmin University of China, Beijing, China  
e-mail: [qianjx1917@ruc.edu.cn](mailto:qianjx1917@ruc.edu.cn)

© The Author(s) 2023  
Z. Zheng (ed.), *Proceedings of the Second International Forum on Financial Mathematics and Financial Technology*, Financial Mathematics and Fintech,  
[https://doi.org/10.1007/978-981-99-2366-3\\_5](https://doi.org/10.1007/978-981-99-2366-3_5)

# 1 Introduction

## 1.1 Background

Iterative learning control (ILC) is a control method applicable to systems doing repeated operations. The basic idea is to use the input and error signals from previous iteration to improve the input of the next iteration. Arimoto et al. first proposed iterative learning control for robotic arms in 1984 and clarified the basic idea of iterative learning control (Arimoto et al., 1984). Subsequently, academia has published numerous chapters around ILC. It has gradually become one of the important branches in the field of control and is widely used in robotics, industrial production and hard disk manufacturing, and other controlled systems with repeated operation.

To achieve excellent control performance, most ILC assume that full operational data and system information can be obtained and utilized. However, in real systems, data delays and dropouts often occur due to various uncertainties. On the other hand, when the system structure is complex or unstable, it is difficult to obtain the system information accurately. To solve the incomplete information problems, it is of great theoretical and practical significance to design ILC algorithms with high performance.

Information incompleteness can be classified into two categories, objective and subjective incompleteness. Information incompleteness caused by objective factors is often related to the uncertainty of the system itself. For example, during the transmission of the signal, the instability of the channel can cause data packet loss. Three main random packet dropout models have been developed for this problem: the random sequence model, the Bernoulli distribution model, and the Markov chain model. Shen (2018) designed an iterative learning control algorithm based on the stochastic approximation algorithm corresponding to the three models and proved that the algorithms satisfy mean-square convergence and probabilistic strong convergence. Information incompleteness due to subjective factors usually artificially assumes that the system information is unknown, thus avoiding the complexity of system modeling and system instability. For example, Oomen et al. (2014) designed a model-free data-driven iterative learning control algorithm for  $H_\infty$ -parametric estimation of multi-input multi-output (MIMO) systems, which obtains the full gradient by conducting  $n_0 \times n_1$  experiments on  $n_0 \times n_1$ -dimensional MIMO systems. However, this algorithm is difficult to be applied to large MIMO systems due to the excessive number of experiments. Subsequently, Aarnoudse et al. (Owens et al., 2009) designed an iterative learning control algorithm based on the stochastic approximation method by constructing a random matrix to estimate the gradient, which effectively reduces the number of experiments.

It is important to note that the effect of information incompleteness on ILC tracking performance is essentially the robustness of ILC. However, this robustness differs for objective and subjective-type information incompleteness problems. The former is usually model-based and emphasizes modeling to analyze the causes of information deficiency. While the latter is data-based and is generally not concerned with

the causes of information deficiency but with the inherent limitations of information deficiency on control performance. Model-based and data-based control methods are not opposed. To achieve the best control effect, the two control methods can also be used in combination. Existing studies on ILC for solving the information incompleteness problems are usually based on stochastic approximation method or other gradient methods. In this chapter, we will use a stochastic variance reduction gradient (SVRG) method to give a general framework for solving the system information incompleteness problems.

## *1.2 Design and Analysis of SVRG-Based ILC*

Modeling the control objective as an optimization function, for a deterministic discrete-time linear system, Owens et al. (Aarnoudse & Oomen, 2020) proposed a gradient-type ILC algorithm based on optimization ideas and analyzed the stability, monotonicity, and robustness of the algorithm. For noisy discrete-time linear systems, Yang and Ruan (2017) proposed an enhanced gradient-based ILC algorithm that can effectively converge in the presence of perturbations in the system. However, the above gradient-based ILC algorithm requires full error and system information for each iteration, and when this information is not fully available, the traditional gradient-based ILC algorithm is no longer applicable.

Notice that in Machine Learning, Stochastic Gradient Descent (SGD) method replaces the total gradient by randomly selecting a partial gradient each time. Corresponding to the control problems, the partial gradient can also be obtained when there is insufficient information about the error or the system. This correlation inspires us to find suitable stochastic gradient methods to solve errors or system information insufficient problems.

In order to improve the convergence speed and apply to non-smooth and non-strongly convex objective functions, recent research in Machine Learning has produced a large number of improved versions of stochastic gradient descent algorithms, including momentum method, variance reduction method, and incremental aggregated gradients. Allen-Zhu (2018) divided these algorithms to three types according to their complexity under strongly convex conditions. The first generation is the momentum-based gradient algorithm, the second generation includes the variance reduction-based gradient algorithm and the proximal stochastic variance reduction gradient algorithm, the third generation includes the Katyusha algorithm and incremental aggregated gradient algorithms. In most cases, the complexity of the algorithms decreases with the growth of generation. Considering specific control problems, the algorithms in first generation are slow to converge and often fail to meet the practical needs, while the algorithms in third generation require accurate system modeling to achieve faster convergence and are difficult to apply to data-driven ILC. Therefore, the research in this chapter is mainly based on the algorithm in second generation—Stochastic Variance Reduction Gradient (SVRG) algorithm.

### 1.3 Main Work and Organization

The purpose of this chapter is to construct SVRG-based ILC and use this framework to solve specific information incompleteness problems. As representatives of objective and subjective information incompleteness, two scenarios, error data random dropouts and model-free ILC, are selected in this chapter to give the corresponding SVRG-based ILC algorithms, respectively. The contribution is threefold.

1. Propose a SVRG-based ILC framework for single-input single-output (SISO) systems. The algorithm is shown to converge linearly under smooth and strongly convex conditions.
2. Apply the SVRG-based ILC framework to error data random data dropouts and give the convergence proof of the algorithm.
3. Extend the SVRG-based ILC framework to multi-input multi-output (MIMO) systems in model-free data-driven scenario and prove the convergence of the algorithm under smooth and non-strongly convex conditions.

Section 2 serves as the basis of the chapter, giving the SVRG-based ILC framework for SISO systems. Section 3 applies the framework to error data random dropouts problem. Section 4 extends the framework to MIMO systems in model-free scenario. Since Sect. 2 only gives the algorithm framework and does not cover the specific scenario, Sect. 2 does not give numerical simulations and contains only three parts: system description, algorithm design, and convergence analysis. Both Sects. 3 and 4 include four parts: system description, algorithm design, convergence analysis, and numerical simulation.

## 2 SVRG-Based ILC Framework

As the basis of the following sections, this section uses SISO systems to give the basic framework of SVRG-based ILC algorithm. This section includes three parts: system description, algorithm design, and convergence analysis.

### 2.1 System Description

Consider the following single-input single-output (SISO) discrete-time linear system

$$\begin{cases} x_k(t+1) = Ax_k(t) + Bu_k(t), \\ y_k(t) = Cx_k(t), \end{cases} \quad (1)$$

where  $t = 0, 1, \dots, N-1$  is time index, and  $k = 1, 2, 3, \dots$  denotes the iteration index.  $x_k(t) \in \mathbb{R}^n$ ,  $u_k(t) \in \mathbb{R}$ , and  $y_k(t) \in \mathbb{R}$  represent the system state, input and

output, respectively.  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^n$ , and  $C \in \mathbb{R}^{1 \times n}$  are the system matrices. The initial condition is the same for each iteration, i.e.,  $x_k(0) = x_0, \forall k \in \mathbb{N}^*$ .

Taking  $t = 0, 1, \dots, N - 1$  in (1) yields

$$\begin{aligned} y_k(1) &= CAx_k(0) + CBu_k(0) = CBu_k(0) + CAx_0, \\ y_k(2) &= CAx_k(1) + CBu_k(1) = CABu_k(0) + CBu_k(1) + CA^2x_0, \\ &\vdots \\ y_k(N) &= CAx_k(N-1) + CBu_k(N-1) \\ &= CA^{N-1}Bu_k(0) + CA^{N-2}Bu_k(1) + \dots \\ &\quad + CABu_k(N-2) + CBu_k(N-1) + CA^Nx_0. \end{aligned}$$

Combining the above equations, system (1) can be rewritten in the following equivalent form

$$y_k = Hu_k + Kx_0, \quad (2)$$

where  $u_k = [u_k(0), u_k(1), \dots, u_k(N-1)]^T \in \mathbb{R}^n$ ,  $y_k = [y_k(1), y_k(2), \dots, y_k(N)]^T \in \mathbb{R}^n$ ,

$$H = \begin{bmatrix} h_{11} & 0 & \dots & 0 \\ h_{21} & h_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1} & h_{N2} & \dots & h_{NN} \end{bmatrix} = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \dots & CB \end{bmatrix}, K = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}.$$

For further analysis, the following assumptions are required.

**Assumption 1** The input/output coupling matrix  $CB \neq 0$ .

**Assumption 2** For desired trajectory  $y_d(t)$ , there exists a unique desired input  $u_d(t)$  and initial state  $x_d(0)$  such that

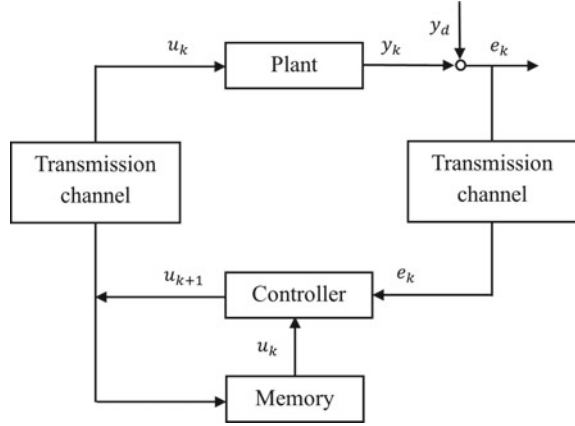
$$\begin{cases} x_d(t+1) = Ax_d(t) + Bu_d(t), \\ y_d(t) = Cx_d(t). \end{cases} \quad (3)$$

Also written in the form of (2), we have

$$y_d = Hu_d + Kx_d(0). \quad (4)$$

**Remark 1** Assumptions 1 and 2 describe the realizability of the system for desired trajectory  $y_d$ . To be specific, Assumption 1 means that the relative degree of the system is 1. Assumption 2 describes the existence of an input signal  $u_d$  that can precisely trace  $y_d$ . If the system does not satisfy the Assumption 2, the system output can only be as close as possible to the desired trajectory  $y_d$ .

**Fig. 1** Block diagram of ILC



**Assumption 3** The initial states of (1) and (3) are identical, i.e.,  $x_d(0) = x_k(0) = x_0, \forall k$ . Assume that  $x_0 = 0$ .

**Remark 2** Assumption 3 is based on the requirement for system repeatability in ILC. In order to simplify the algorithm, without loss of generality, take  $x_0 = 0$ . It is easy to verify that the result of this chapter is also valid when  $x_0 \neq 0$ .

In this chapter, the above three assumptions will be followed, but in fact, the SVRG-based ILC can also be established when Assumptions 1 and 2 are appropriately relaxed. Section 4 will give specific explanations on how to relax these assumptions.

Figure 1 illustrates the basic framework of ILC. The plant takes input  $u_k$  and generates output  $y_k$  and gets the error  $e_k = y_d - y_k$  between the output and the desired trajectory  $y_d$ , which is transmitted to the controller. The controller uses error  $e_k$  and input  $u_k$  to calculate the input signal  $u_{k+1}$  for the next batch and transmits it to the plant. Our goal is to find a sequence of input  $\{u_k\}$ , s.t.

$$\lim_{k \rightarrow \infty} \|e_k\| = \lim_{k \rightarrow \infty} \|y_d - y_k\| = 0, \quad (5)$$

where  $\|\cdot\|$  is the vector 2-norm and its induced matrix norm, and henceforth refers to this norm if not otherwise specified.

By Assumptions 2 and 3, (5) is equivalent to the optimization problem of function  $F$ :

$$F(u_k) \triangleq \frac{1}{2N} \|e_k\|^2 = \frac{1}{2N} \|y_d - H u_k\|^2, \quad \lim_{k \rightarrow \infty} F(u_k) = 0. \quad (6)$$

## 2.2 Algorithm Design

The traditional gradient-based ILC updating law (Gu et al., 2019) is

$$u_{k+1} = u_k - \eta_k \nabla_k, \quad (7)$$

where  $\eta_k$  denotes step length, and  $\nabla_k$  is the gradient of the objective function. From (6), we have

$$\nabla_k = \nabla F(u_k) = -\frac{1}{N} H^T e_k. \quad (8)$$

In (8), calculating the full gradient requires all the information of conjugate matrix  $H^T$  and error  $e_k$ . To give the gradient under partial information, consider decomposing the error  $e_k = [e_k(1), e_k(2), \dots, e_k(N)]^T$  according to the time index. Let  $f_i(u_k) = \frac{1}{2} \|e_k(i)\|^2 = \frac{1}{2} (y_d(i) - h_i^T u_k)^2$ , where  $h_i = [h_{i1}, \dots, h_{in}, 0, \dots, 0]^T$  denotes the  $i$ -th row of the matrix  $H$ . Then, equation (6) can be rewritten as

$$F(u_k) = \frac{1}{N} \sum_{i=1}^n f_i(u_k). \quad (9)$$

Take the gradient of both sides, we have

$$\nabla F(u_k) = \frac{1}{N} \sum_{i=1}^n \nabla f_i(u_k) = \frac{1}{N} \sum_{i=1}^n -h_i e_k(i). \quad (10)$$

Define random gradient  $\tilde{\nabla}_k$  as a discrete random variable that takes value uniformly over  $\{\nabla f_i(u_k)\}_{i=1}^n$ , satisfying  $P(\tilde{\nabla}_k = \nabla f_i(u_k)) = \frac{1}{N}$ . Therefore  $\mathbb{E}[\tilde{\nabla}_k] = \nabla_k$ , i.e.,  $\tilde{\nabla}_k$  is unbiased estimation of  $\nabla_k$ .

Note that by decomposing (6)–(9), calculating the specific value  $\nabla f_i(u_k)$  of random vector  $\tilde{\nabla}_k$  only requires one row of the system matrix  $H$  and one-dimensional information of the error  $e_k$ . Therefore the decomposition can effectively reduce the information required for each iteration. This technique of gradient decomposition is the basis for solving the ILC of information incompleteness using SVRG method in this chapter. In Sects. 3 and 4, two specific decomposition methods are presented for incompleteness of error and system information, respectively.

Consider the stochastic gradient descent (SGD) method used in Machine Learning. Replacing the full gradient  $\nabla_k$  with the stochastic gradient  $\tilde{\nabla}_k$  in the ILC updating law (7), we can obtain the SGD-based ILC algorithm. However, the convergence rate of SGD algorithm is  $O(1/\sqrt{k})$  even under strongly convex condition (Allen-Zhu, 2018), which cannot meet the practical requirements. This is because although the stochastic gradient  $\tilde{\nabla}_k$  is unbiased estimate of the full gradient  $\nabla_k$ , the variance accumulates as the iteration increases. To reduce the variance, Johnson and Zhang (2013) proposed a general stochastic variance reduced gradient (SVRG) descent method. By recording a ‘‘snapshot’’  $\tilde{u}^s$  every few updates to construct an converging upper bound of the gradient, the rate of convergence of SVRG method is  $O(\rho^k)$  under strongly convex condition and  $O(1/k)$  under non-strongly convex

condition. Based on this method, the input updated with “snapshot”  $\tilde{u}^s$  is denoted as  $u_{s,k}$ , and the SVRG-based ILC updating law is

$$u_{s,k+1} = u_{s,k} - \eta (\nabla f_i(u_{s,k}) - \nabla f_i(\tilde{u}^s) + \nabla F(\tilde{u}^s)). \quad (11)$$

For system (2), the SVRG-based ILC algorithm with updating law (11) is shown in Algorithm 1.

---

**Algorithm 1** SISO SVRG-based ILC framework for SISO systems

---

**Input:**  $\eta, u_{0,0};$   
 $m \leftarrow 2N; \tilde{u}^0 \leftarrow u_{0,0};$   
**for**  $s \leftarrow 0$  to  $S - 1$  **do**  
 $u_{s,0} \leftarrow \tilde{u}^s, \mu_s \leftarrow \nabla F(\tilde{u}^s);$   
**for**  $k \leftarrow 0$  to  $m - 1$  **do**  
 $w_k \leftarrow \nabla f_i(u_{s,k}) - \nabla f_i(\tilde{u}^s) + \mu_s;$  where  $i$  from  $\{1, 2, \dots, N\}$  randomly  
 $u_{s,k+1} \leftarrow u_{s,k} - \eta w_k;$   
**end for**  
**Option I:**  $\tilde{u}^{s+1} \leftarrow \frac{1}{m} \sum_{k=0}^{m-1} u_{s,k};$   
**Option II:**  $\tilde{u}^{s+1} \leftarrow u_{s,m};$   
**end for**

---

Algorithm 1 has two loops. The outer loop updates the “snapshot”  $\tilde{u}^s$  once when the inner loop iterates  $m$  times. The iteration length  $m$  is taken as an integer multiple of  $N$ , which is empirically set to  $2N$ . Line 9 and 10 of Algorithm 1 shows two ways of updating the “snapshot”, Option I and Option II. Option I takes the average of the first  $m - 1$  inputs as the “snapshot”, without using  $u_{s,m}$ , so actually the inner loop only requires  $m - 1$  iterations. The corresponding Option II takes the  $m$ th-iteration and uses  $u_{s,m}$  as the “snapshot”. The two “snapshot” updating methods do not change the convergence of Algorithm 1 (Bottou et al., 2018). Due to the limitation of space, we only prove the convergence of Option I under strongly convex conditions and Option II under non-strongly convex conditions in this section and Sect. 4, respectively.

## 2.3 Convergence Analysis

This subsection is divided into two parts, first giving the convex optimization knowledge required for the proof of this chapter and then analyzing the convergence of the system (2) when the “snapshot” update method of Algorithm 1 is set for Option I.

### 2.3.1 Preliminaries of Convex Optimization

The basics of convex optimization required for this chapter are given below (Lyubashevsky, 2005).



**Definition 1** (*Smoothness*) Suppose  $S$  is a nonempty convex subset of  $\mathbb{R}^d$ ,  $f : S \rightarrow \mathbb{R} \in C^1$ . If  $\exists L > 0$ , s.t.  $\forall x, y \in S$ ,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|,$$

then we say that  $f$  is  $L$ -smooth or  $\nabla f(x)$  is  $L$ -Lipschitz continuous on  $S$ , where  $L$  is the Lipschitz constant.

**Definition 2** (*Strong convexity*) Suppose  $S$  is a nonempty convex subset of  $\mathbb{R}^d$ ,  $f : S \rightarrow \mathbb{R} \in C^1$ . If  $\exists \sigma > 0$ , s.t.  $\forall x, y \in S$ ,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\sigma}{2}\|x - y\|^2,$$

then we say that  $f$  is  $\sigma$ -strongly convex on  $S$ . When  $\sigma = 0$ ,  $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$ ,  $f$  is convex.

**Definition 3** (*Conditional number*) If  $f$  is  $L$ -smooth and  $\sigma$ -strongly convex,  $\kappa = L/\sigma$  is the conditional number of  $f$ .

**Theorem 1** For convex function  $f$ , the followings are equivalent:

- a.  $\nabla f(x)$  is  $L$ -Lipschitz continuous,
- b.  $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2$ ,
- c.  $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2L}\|\nabla f(y) - \nabla f(x)\|^2$ ,
- d.  $\frac{1}{L}\|\nabla f(y) - \nabla f(x)\|^2 \leq \langle \nabla f(x) - \nabla f(y), x - y \rangle$ .

**Proof**  $a \rightarrow b$ : Denote  $g(t) = f(t(y - x) + x)$ , then  $f(x) = g(0)$ ,  $f(y) = g(1)$ , and  $g'(t) = \langle \nabla f(t(y - x) + x), y - x \rangle$ . Therefore,

$$\begin{aligned} f(y) - f(x) - \langle \nabla f(x), y - x \rangle &= g(1) - g(0) - \langle \nabla f(x), y - x \rangle \\ &= \int_0^1 g'(t)dt - \langle \nabla f(x), y - x \rangle \\ &= \int_0^1 \langle \nabla f(t(y - x) + x) - \nabla f(x), y - x \rangle dt \\ &\leq \int_0^1 \|\nabla f(t(y - x) + x) - \nabla f(x)\| \cdot \|y - x\| dt \\ &\leq \int_0^1 L\|t(y - x)\| \cdot \|y - x\| dt = \frac{L}{2}\|y - x\|^2. \end{aligned}$$

$b \rightarrow c$ : Denote  $f_x(z) = f(z) - \langle \nabla f(x), z \rangle$ , for  $\forall z, z' \in \mathbb{R}^d$ ,

$$\begin{aligned}
f(z') - f(z) &\leq \langle \nabla f(z), z' - z \rangle + \frac{L}{2} \|z' - z\|^2, \\
f(z') - f(z) - \langle \nabla f(x), z' - z \rangle &\leq \langle \nabla f(z) - \nabla f(x), z' - z \rangle + \frac{L}{2} \|z' - z\|^2, \\
f_x(z') - f_x(z) &\leq \langle \nabla f_x(z), z' - z \rangle + \frac{L}{2} \|z' - z\|^2. \tag{12}
\end{aligned}$$

By using the convexity of  $f$ , we have

$$\begin{aligned}
f_x(z') - f_x(z) &= f(z') - f(z) - \langle \nabla f(x), z' - z \rangle \\
&\geq \langle \nabla f(z), z - z' \rangle - \langle \nabla f(x), z - z' \rangle = \langle \nabla f_x(z), z - z' \rangle.
\end{aligned}$$

Therefore  $f_x(z)$  is also convex, since  $\nabla f_x(z) = \nabla f(z) - \nabla f(x)$ ,  $f_x(z)$  achieves its minimum at  $z = x$ . By (12),

$$\begin{aligned}
f_x(x) = \min_{z'} f_x(z') &\leq \min_{z'} \left\{ f_x(z) + \langle \nabla f_x(z), z' - z \rangle + \frac{L}{2} \|z' - z\|^2 \right\} \\
&= f_x(z) + \min_{\|y\|=1} \min_{t \geq 0} \left\{ t \langle \nabla f_x(z), y \rangle + \frac{L}{2} t^2 \right\} \\
&= f_x(z) + \min_{\|y\|=1} \left\{ -\frac{\langle \nabla f_x(z), y \rangle^2}{2L} \right\} \\
&= f_x(z) - \frac{1}{2L} \|\nabla f_x(z)\|^2.
\end{aligned}$$

Therefore  $f_x(z) - f_x(x) \geq \frac{1}{2L} \|\nabla f_x(z)\|^2$ , which implies

$$\begin{aligned}
f(y) - f(x) - \langle \nabla f(x), y - x \rangle &= f_x(y) - f_x(x) \\
&\geq \frac{1}{2L} \|\nabla f_x(y)\|^2 = \frac{1}{2L} \|\nabla f(y) - \nabla f(x)\|^2.
\end{aligned}$$

$c \rightarrow d$  : Swapping  $x$  and  $y$  in  $c$  , we have

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2L} \|\nabla f(y) - \nabla f(x)\|^2.$$

Summing the two equations, we have

$$\frac{1}{L} \|\nabla f(y) - \nabla f(x)\|^2 \leq \langle \nabla f(x) - \nabla f(y), x - y \rangle.$$

$d \rightarrow a$ :  $\|\nabla f(y) - \nabla f(x)\|^2 \leq L \langle \nabla f(x) - \nabla f(y), x - y \rangle \leq L \|\nabla f(x) - \nabla f(y)\| \cdot \|x - y\|$  by Cauchy inequality, thus  $\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$ .  $\square$

**Theorem 2** Let  $f(x) = \frac{1}{2}x^T Qx + q^T x + c$ , where  $Q$  is positive definite. Then  $f(x)$  is  $L$ -smooth and  $\sigma$ -strongly convex, where  $L = \lambda_M$ , and  $\sigma = \lambda_m$ .  $\lambda_M$ , and  $\lambda_m$  are the maximum and minimum eigenvalues of  $Q$ , respectively.

**Proof** Since  $\nabla f(x) = Qx + q$ , we have

$$\|\nabla f(x) - \nabla f(y)\| \leq \|Q(x - y)\| \leq \|Q\| \cdot \|x - y\| = \lambda_M \|x - y\|.$$

Hence  $f(x)$  is  $\lambda_M$ -smooth. It is easy to verify that

$$f(x) - f(y) - \langle \nabla f(y), x - y \rangle = \frac{1}{2}(x - y)^T Q(x - y).$$

Since  $Q$  is positive definite, the orthogonal similarity can be diagonalized as  $Q = P^T \Lambda P$ , where  $P$  is the orthogonal matrix,  $\Lambda$  is the diagonal matrix of eigenvalues, and  $\lambda_m > 0$ . Thus

$$\frac{1}{2}(x - y)^T Q(x - y) = \frac{1}{2}z^T \Lambda z = \frac{1}{2} \sum_i \lambda_i z_i^2 \geq \frac{1}{2} \lambda_m \|z\|^2 = \frac{1}{2} \lambda_m \|x - y\|^2.$$

Thus  $f(x)$  is  $\lambda_m$ -strongly convex.  $\square$

For system (2) and objective function (6), we have:

**Proposition 1** Each  $f_i$  is convex and  $L$ -smooth.

**Proof** For  $f(x) = \frac{1}{2}(q^T x + c)^2$ , where  $q = [q_1, q_2, \dots, q_n]^T \in \mathbb{R}^n$ ,  $x \in \mathbb{R}^n$ ,  $c \in \mathbb{R}$ . Obviously,  $f$  is convex, and  $\nabla f(x) = qq^T x + cq$ ,  $\|\nabla f(x) - \nabla f(y)\| = \|qq^T(x - y)\| \leq \|qq^T\| \cdot \|x - y\| \leq \|q\|^2 \cdot \|x - y\|$ . Therefore, for each  $f_i$  in (9), let  $L = \max_i \{\|h_i\|^2\} > 0$ ,  $h_i = [h_{i1}, \dots, h_{ii}, 0, \dots, 0]^T$ , then  $f_i$  is  $L$ -smooth.  $\square$

**Proposition 2**  $F$  is  $L$ -smooth and  $\sigma$ -strongly convex.

**Proof** By (6), we have

$$\begin{aligned} F(u_k) &= \frac{1}{2N} (y_d - Hu_k)^T (y_d - Hu_k) \\ &= \frac{1}{2N} (u_k^T H^T Hu_k - y_d^T Hu_k - u_k^T H^T y_d + y_d^T y_d), \end{aligned}$$

where  $H^T H$  is positive definite. Then by Theorem 2,  $F(u_k)$  is  $L$ -smooth and  $\sigma$ -strongly convex, where  $L$  and  $\sigma$  are the  $\frac{1}{2N}$  of the maximal and minimal eigenvalues of  $H^T H$ , respectively.  $\square$

### 2.3.2 Proof of Convergence

In Algorithm 1, set the ‘‘snapshot’’ updating as Option I. Then, under the assumptions of system (2), the convergence of Algorithm 1 is given by the following theorem.

**Theorem 3** *If each  $f_i$  is convex and  $L$ -smooth, and  $F$  is  $\sigma$ -strongly convex. We denote the optimal point  $u^* = \operatorname{argmin}_u F(u)$ , and assume that  $m$  is large enough such that*

$$\alpha = \frac{1}{\sigma\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1.$$

*Then the convergence of Algorithm 1 satisfies*

$$\mathbb{E} [F(\tilde{u}^S) - F(u^*)] \leq \alpha^S (F(\tilde{u}^0) - F(u^*)).$$

**Proof** Since  $f_i$  is convex and  $L$ -smooth, for any  $i$ , by Theorem 1,

$$\|\nabla f_i(u) - \nabla f_i(u^*)\|^2 \leq 2L [f_i(u) - f_i(u^*) - \langle \nabla f_i(u^*), u - u^* \rangle]. \quad (13)$$

Since  $\frac{1}{N} \sum_{i=1}^n \nabla f_i(u) = \nabla F(u)$ , and  $\nabla F(u^*) = 0$ , we regard  $\nabla f_i$  as random vectors which take values from  $\{\nabla f_i\}_{i=1}^N$ , then

$$\mathbb{E} [\|\nabla f_i(u) - \nabla f_i(u^*)\|^2] = \frac{1}{N} \sum_{i=1}^n \|\nabla f_i(u) - \nabla f_i(u^*)\|^2 \leq 2L [F(u) - F(u^*)]. \quad (14)$$

For any fixed  $s$ , we set  $w_k = \nabla f_i(u_{s,k}) - \nabla f_i(\tilde{u}^s) + \nabla F(\tilde{u}^s)$ , then

$$\begin{aligned} \mathbb{E} [\|w_k\|^2] &\leq 2\mathbb{E} [\|\nabla f_i(u_{s,k}) - \nabla f_i(u^*)\|^2] \\ &\quad + 2\mathbb{E} [\|\nabla f_i(\tilde{u}^s) - \nabla f_i(u^*) - \nabla F(\tilde{u}^s)\|^2] \\ &= 2\mathbb{E} [\|\nabla f_i(u_{s,k}) - \nabla f_i(u^*)\|^2] \\ &\quad + 2\mathbb{E} [\|\nabla f_i(\tilde{u}^s) - \nabla f_i(u^*)\|^2] - \mathbb{E} [\langle \nabla f_i(\tilde{u}^s) - \nabla f_i(u^*), \nabla f_i(\tilde{u}^s) - \nabla f_i(u^*) \rangle] \\ &\leq 2\mathbb{E} [\|\nabla f_i(u_{s,k}) - \nabla f_i(u^*)\|^2] + 2\mathbb{E} [\|\nabla f_i(\tilde{u}^s) - \nabla f_i(u^*)\|^2] \\ &\leq 4L [F(u_{s,k}) - F(u^*) + F(\tilde{u}^s) - F(u^*)]. \end{aligned} \quad (15)$$

In the above, we have used the inequality  $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ , and the property  $\mathbb{E} [\|\zeta - \mathbb{E}\zeta\|^2] = \mathbb{E}\|\zeta\|^2 - \|\mathbb{E}\zeta\|^2 \leq \mathbb{E}\|\zeta\|^2$ , as well as (14). Notice that  $\mathbb{E} [w_k] = \nabla F(u_{s,k})$ , thus

$$\begin{aligned}
 & \mathbb{E} \left[ \|u_{s,k+1} - u^*\|^2 \right] \\
 &= \|u_{s,k} - u^*\|^2 - 2\eta \mathbb{E} [\langle w_k, u_{s,k} - u^* \rangle] + \eta^2 \mathbb{E} [\|w_k\|^2] \\
 &\leq \|u_{s,k} - u^*\|^2 - 2\eta [\langle \nabla F(u_{s,k}), u_{s,k} - u^* \rangle] \\
 &\quad + 4L\eta^2 [F(u_{s,k}) - F(u^*) + F(\tilde{u}^s) - F(u^*)] \\
 &\leq \|u_{s,k} - u^*\|^2 - 2\eta [F(u_{s,k}) - F(u^*)] \\
 &\quad + 4L\eta^2 [F(u_{s,k}) - F(u^*) + F(\tilde{u}^s) - F(u^*)] \\
 &= \|u_{s,k} - u^*\|^2 - 2\eta(1 - 2L\eta) [F(u_{s,k}) - F(u^*)] \\
 &\quad + 4L\eta^2 [F(\tilde{u}^s) - F(u^*)].
 \end{aligned}$$

In the above, we have used (15) and the convexity of  $F$ , i.e.,  $\langle \nabla F(u_{s,k}), u_{s,k} - u^* \rangle \geq F(u_{s,k}) - F(u^*)$ .

We sum up the expectations of the above equation for  $k = 0, 1, \dots, m-1$ . Using the convexity of  $F$  and the selection of  $\tilde{u}^{s+1}$  under Option I, we have  $F(\tilde{u}^{s+1}) = F\left(\frac{1}{m} \sum_{k=0}^{m-1} u_{s,k}\right) \leq \frac{1}{m} \sum_{k=0}^{m-1} F(u_{s,k})$ . Therefore,

$$\begin{aligned}
 & \mathbb{E} \left[ \|u_{s,m} - u^*\|^2 \right] + 2\eta(1 - 2L\eta)m \mathbb{E} [F(\tilde{u}^{s+1}) - F(u^*)] \\
 &\leq \mathbb{E} \left[ \|u_{s,0} - u^*\|^2 \right] + 4L\eta^2 m \mathbb{E} [F(\tilde{u}^s) - F(u^*)] \\
 &\leq \frac{2}{\sigma} \mathbb{E} [F(\tilde{u}^s) - F(u^*)] + 4L\eta^2 m \mathbb{E} [F(\tilde{u}^s) - F(u^*)].
 \end{aligned}$$

Thus, we obtain

$$\begin{aligned}
 \mathbb{E} [F(\tilde{u}^{s+1}) - F(u^*)] &\leq \left( \frac{1}{\sigma\eta(1 - 2L\eta)m} - \frac{2L\eta}{1 - 2L\eta} \right) \mathbb{E} [F(\tilde{u}^s) - F(u^*)] \\
 &= \alpha \mathbb{E} [F(\tilde{u}^s) - F(u^*)].
 \end{aligned}$$

Summing up the above equation for  $s = 0, 1, \dots, S-1$ , we have

$$\mathbb{E} [F(\tilde{u}^S) - F(u^*)] \leq \alpha^S \mathbb{E} [F(\tilde{u}^0) - F(u^*)]. \quad \square$$

**Remark 3** Theorem 3 indicates that Algorithm 1 has the rate of convergence  $O(\alpha^S)$ . And this convergence rate is related to the value of  $\alpha$ . If the information of the system  $H$  is known, to make  $\alpha$  as small as possible, we generally take  $\eta = \frac{0.1}{L}$ ,  $m = \Theta(n)$ , so that the value of  $\alpha$  is close to  $\frac{1}{2}$ . If the system information is unknown, we need to find the appropriate  $\eta$  and  $m$  by experiment.

### 3 SVRG-Based ILC Under Random Data Dropouts

This section follows the SISO system in Sect. 2, but assumes that random data dropouts occur in the error signal transmission. This section consists of four parts: system description, algorithm design, performance analysis, and numerical simulation.

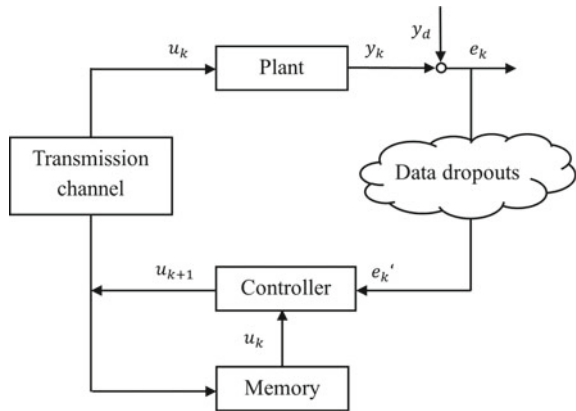
#### 3.1 System Description

In SISO system (2), we still hold Assumptions 1–3, but assuming that data dropouts occur in the transmission of the error signal, as shown in Fig. 2. We further assume that the dropouts satisfy the Bernoulli distribution model (Shen, 2018). Therefore, the ILC updating law (7) becomes

$$u_{k+1} = u_k + \eta \frac{1}{N} H^T \Gamma_k e_k, \tag{16}$$

where  $\Gamma_k = \text{diag} \{ \gamma_k(1), \gamma_k(2), \dots, \gamma_k(N) \}$ .  $\{ \gamma_k(i) \}_{i=1}^N$  is i.i.d following Bernoulli distribution. Let  $\gamma \triangleq \mathbb{E} [ \gamma_k(i) ]$  be the successful transmission rate, where  $\gamma_k(i) = 0$  means data dropout occurs in the  $i$ -th time of  $k$ -th batch, and otherwise, data dropout does not occur.

**Fig. 2** ILC with random data dropouts



### 3.2 Algorithm Design

Based on gradient descent method, there are two main approaches to solve the data dropouts problem:

- (1) Obtain the full gradient by retransmitting. For each transmission, the controller stores the successfully transmitted data and asks the lost data to be retransmitted until all data are received. This method eliminates the effect of data dropouts by retransmitting.
- (2) Use successfully transmitted data to construct random gradient. The data is updated directly using successfully transmitted data each iteration, as shown in the update law (16).

The first method requires a lot of wasted time when data retransmission is slow. Although the second method saves the time of data retransmission, the actual running time may be larger than first method when data retransmission.

Based on the framework of Algorithm 1, the SVRG-based ILC under error data dropouts can be constructed by utilizing the second method for each iteration, but calculating the full gradient every several iterations using the first method. This algorithm does not require data retransmission in most cases compared to the first method and has a significant improvement in convergence speed compared to the second method. Thus it can achieve a good balance between convergence rate and data retransmission speed, and it is more suitable for general data dropout cases.

The formal construction of the algorithm is given below.

Firstly, we take the random gradient  $\tilde{\nabla}_k \triangleq -\frac{1}{\gamma N} H^T \Gamma_k e_k$ , and we note that  $\mathbb{E}[\tilde{\nabla}_k] = -\frac{1}{N} H^T e_k = \nabla_k$ . For the convenience of proof, we present  $\tilde{\nabla}_k$  as

$$\nabla \tilde{F}_k(u_k) = \frac{1}{\gamma N} \sum_{\{i|\gamma_k(i)=1\}} \nabla f_i(u_k), \quad (17)$$

where  $\nabla f_i$  is defined in the same way as (10). Notice that if there is no dropout,  $\gamma = 1$ , and therefore (17) is equivalent to (10).

**Remark 4** Equation (17) is similar to the Batch Gradient Descent (BGD) method in Machine Learning, but they are fundamentally different. In (17), the number of  $\nabla f_i$  in each summation  $\sum_i \nabla f_i$  varies according to the value of the random vector  $\{\gamma_k(i)\}_{i=1}^N$ . But in BGD, the number of  $\nabla f_i$  is fixed. Therefore, the algorithm based on gradient  $\nabla \tilde{F}_k$  cannot be directly applied to BGD.

Secondly, similar to (11), ILC updating law under random data dropouts is constructed:

$$u_{s,k+1} = u_{s,k} - \eta \left( \nabla \tilde{F}_k(u_{s,k}) - \nabla \tilde{F}_k(\tilde{u}^s) + \nabla F(\tilde{u}^s) \right). \quad (18)$$

Finally, change the iteration length  $m$  in Algorithm 1 from  $2N$  to  $\lceil 2\gamma N \rceil$ . Because the number of summation in each batch is  $\mathbb{E} \left[ \sum_{i=1}^N \gamma_k(i) \right] = \gamma N$  in the desired sense,  $\nabla \tilde{F}_k$  is equivalent to  $\gamma N$  sum of  $\nabla f_i$ .

In conclusion, SVRG-based ILC under random data dropouts is shown in Algorithm 2.

---

**Algorithm 2** Data dropout SISO SVRG-based ILC
 

---

**Input:**  $\eta, u_{0,0};$   
 $m \leftarrow 2\gamma N; \tilde{u}^0 \leftarrow u_{0,0};$   
**for**  $s \leftarrow 0$  to  $S - 1$  **do**  
 $u_{s,0} \leftarrow \tilde{u}^s, \mu_s \leftarrow \nabla F(\tilde{u}^s);$   
**for**  $k \leftarrow 0$  to  $m - 1$  **do**  
 $w_k \leftarrow \nabla f_i(u_{s,k}) - \nabla f_i(\tilde{u}^s) + \mu_s;$   
 $u_{s,k+1} \leftarrow u_{s,k} - \eta w_k;$   
**end for**  
 $\tilde{u}^{s+1} \leftarrow \frac{1}{m} \sum_{k=0}^{m-1} u_{s,k};$   
**end for**

---

For the “snapshot” of Algorithm 2, the update method is taken as Option I in Algorithm 1, and the recommended iteration length is set to  $\lceil 2\gamma N \rceil$ . When  $\gamma$  is unknown, we need to find the appropriate  $m$  by experiments.

### 3.3 Convergence Analysis

By Proposition 1, every  $f_i$  is convex and  $L$ -smooth, and the following proposition holds:

**Proposition 3** *Each value of  $\nabla \tilde{F}_k(u_k)$  is convex and  $L'$ -smooth, where  $L' = L/\gamma$ , and  $L$  is the Lipschitz constant corresponding to the smoothness of  $f_i$  in Proposition 1.*

**Proof** Since every  $f_i$  is convex and  $L$ -smooth,  $\frac{1}{\gamma N} \sum_{i=1}^N \nabla f_i(u_k)$  is  $L'$ -Lipschitz continuous, where  $L' = L/\gamma$ . Because each value of  $\nabla \tilde{F}_k(u_k)$  is a linear combination of  $\nabla f_i$ , the summation number does not exceed  $\frac{1}{\gamma N} \sum_{i=1}^N \nabla f_i(u_k)$ . Therefore  $\nabla f_i(u_k)$ ; as a result,  $\nabla \tilde{F}_k(u_k)$  is convex and  $L'$ -smooth.  $\square$

For the convergence of Algorithm 2, we have the following theorem.

**Theorem 4** *If each value of  $\nabla \tilde{F}_k(u_k)$  is convex and  $L'$ -smooth,  $F$  is  $L$ -smooth and  $\sigma$ -strongly convex, and for the optimal point  $u^* = \operatorname{argmin}_u F(u)$ , assuming that  $m$  is large enough s.t.*

$$\alpha = \frac{1}{\sigma \eta (1 - 2L'\eta) m} + \frac{2L'\eta}{1 - 2L'\eta} < 1.$$



Then the convergence of Algorithm 2 satisfies

$$\mathbb{E} [F(\tilde{u}^S) - F(u^*)] \leq \alpha^S (F(\tilde{u}^0) - F(u^*)).$$

**Proof** By Proposition 3, replacing  $f_i$  in the proof of Theorem 3 with  $\nabla \tilde{F}_k$ , equation (13) is rewritten as:

$$\|\nabla \tilde{F}_k(u) - \nabla \tilde{F}_k(u^*)\|^2 \leq \frac{1}{\gamma N} \sum_{\{i|\gamma_k(i)=1\}} 2L' [f_i(u) - f_i(u^*) - \langle \nabla f_i(u^*), u - u^* \rangle].$$

The corresponding Eq. (14) is

$$\begin{aligned} & \mathbb{E} \left[ \left\| \nabla \tilde{F}_k(u) - \nabla \tilde{F}_k(u^*) \right\|^2 \right] \\ & \leq \frac{1}{\gamma N} \mathbb{E} \left[ \sum_{\{i|\gamma_k(i)=1\}} 2L' [f_i(u) - f_i(u^*) - \langle \nabla f_i(u^*), u - u^* \rangle] \right] \\ & = \frac{1}{\gamma N} \gamma \mathbb{E} [2L' [F(u) - F(u^*) - \langle \nabla F(u^*), u - u^* \rangle]] \\ & = 2L' \mathbb{E} [F(u) - F(u^*)]. \end{aligned}$$

The rest of the proof repeats the proof of Theorem 3.  $\square$

**Remark 5** Theorem 4 shows that Algorithm 2 also converges linearly, and the speed of convergence is related to  $\alpha$ . For the choice of  $m$ , note that Theorem 4 differs from Theorem 3 in the Lipschitz constant corresponding to the smoothness of the condition. By Proposition 3, with  $L' = L/\gamma$ , for

$$\alpha = \frac{1}{\sigma \eta (1 - 2L'\eta) m} + \frac{2L'\eta}{1 - 2L'\eta}.$$

We can consider multiplying  $m$  by  $\gamma$  times, i.e., changing  $m$  from  $2N$  to  $[2\gamma N]$ , to approximately keep the convergence of Algorithm 2.

### 3.4 Numerical Simulation

In SISO system (1), take the system matrix  $(A, B, C)$  as

$$A = \begin{bmatrix} 0.50 & -0.25 & 1.00 \\ 0.15 & 0.30 & -0.50 \\ -0.75 & 0.25 & -0.25 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C = [0 \ 0 \ 1.0].$$

Take desired trajectory  $y_d(t) = \sin(2\pi t/50)$ , time length  $N = 50$ , initial state  $x_0 = 0$ , and initial input  $u_0 = 0$ . When  $\gamma = 0.9$  and  $\gamma = 0.6$ , the ILC based on full gradient (GD), stochastic gradient (SGD) and stochastic variance reduced gradient (SVRG) is shown in Fig. 3.

When calculating the full gradient, each data retransmission increases 1 to the iteration number, which means that the controller skips one round of computation until all error information is completely transmitted. Take the optimal step that the three methods can converge.

When  $\gamma = 0.9$ , Fig. 3a shows that the SVRG-based ILC converges slightly faster than the GD- and SGD-based ILC. When  $\gamma = 0.6$ , Fig. 3b illustrates a significant difference in the convergence speed of the three types ILC, from fast to slow for SVRG-, SGD-, and GD-based ILC. In summary, the SVRG-based ILC under error data dropouts, i.e., Algorithm 2, outperforms the GD- and SGD-based ILC under different successful transmission rates, and the difference becomes more significant as the  $\gamma$  decreases.

## 4 Model-Free SVRG-Based ILC for MIMO Systems

This section extends the Algorithm 1 in Sect. 2 from SISO systems to MIMO systems. Firstly, a system description of the discrete linear MIMO system is given. Secondly, the existing model-free data-driven methods are introduced, and a new model-free data-driven ILC based on SVRG method is constructed. Thirdly, the convergence of the algorithm under non-strongly convex conditions is proved. Finally, numerical simulations are established to verify the convergence performance of SVRG-based ILC in deterministic and noisy systems.

### 4.1 System Description

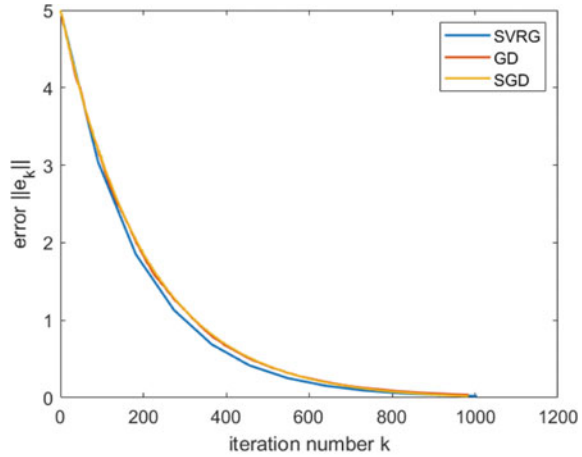
Consider the following discrete linear multi-input multi-output (MIMO) system  $\mathcal{J}$ , which has  $q$  inputs  $u_k^1, u_k^2, \dots, u_k^q$ , and  $p$  outputs  $y_k^1, y_k^2, \dots, y_k^p$ . Rewrite the system in the form of (2),

$$\mathbf{y}_k = \mathcal{J} \mathbf{u}_k, \quad (19)$$

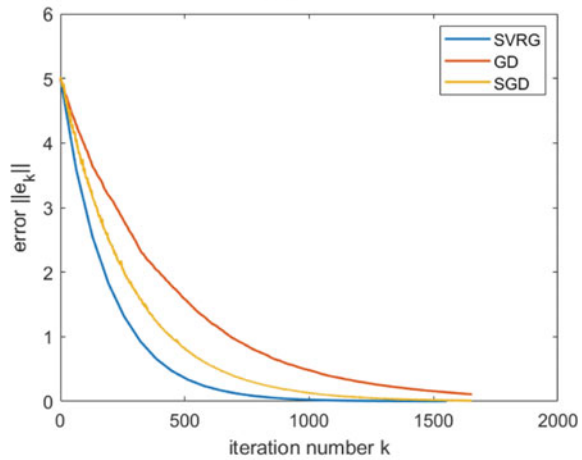
where

$$\mathcal{J} = \begin{bmatrix} J_{11} & J_{12} & \cdots & J_{1q} \\ J_{21} & J_{22} & \cdots & J_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ J_{p1} & J_{p2} & \cdots & J_{pq} \end{bmatrix}, \quad \mathbf{u}_k = \begin{bmatrix} u_k^1 \\ u_k^2 \\ \vdots \\ u_k^q \end{bmatrix}, \quad \mathbf{y}_k = \begin{bmatrix} y_k^1 \\ y_k^2 \\ \vdots \\ y_k^p \end{bmatrix}.$$

**Fig. 3** Comparison of three gradient-based ILC under error data dropouts



(a)  $\gamma = 0.9$



(b)  $\gamma = 0.6$

Each  $J_{ij} \in \mathbb{R}^{N \times N}$  has the same properties as matrix  $H$  in (2).  $y_k^i = [y_k^i(1), \dots, y_k^i(N)]^T$ ,  $u_k^j = [u_k^j(0), \dots, u_k^j(N-1)]^T$ ,  $N$  is the length of time, and the desired trajectory is

$$y_d = [(y_d^1)^T, (y_d^2)^T, \dots, (y_d^p)^T]^T.$$

For this system, consider the following assumptions:

**Assumption 4** System matrix  $\mathcal{J} \neq 0$ .

**Assumption 5** The dimension of input signal does not exceed the dimension of output signal, i.e.,  $p \geq q$ .

**Remark 6** If  $p = q = 1$ , the system (19) degenerates to SISO system (2). Unlike Assumptions 1 and 4 can no longer guarantee that the system matrix  $\mathcal{J}$  is of full rank. Assumption 4 is the most fundamental, since if  $\mathcal{J} = 0$ , any input signal cannot track  $y_d$ . Assumption 2 is also relaxed from the system, the reasons will be given in the proof of the convergence. In addition, Assumption 5 is added for the MIMO system, because if the input dimension is larger than the output dimension, it means that there is a redundant information.

For desired trajectory  $y_d$ , consider control objective similar to (6), i.e., to find a sequence  $\{\mathbf{u}_k\}$ , s.t.

$$G(\mathbf{u}_k) = \frac{1}{2p} \|\mathbf{e}_k\|^2 = \frac{1}{2p} \|\mathbf{y}_d - \mathcal{J}\mathbf{u}_k\|^2, \quad \lim_{k \rightarrow \infty} G(\mathbf{u}_k) = G(\mathbf{u}^*), \quad (20)$$

where  $\mathbf{u}^*$  is the input when  $G$  takes the minimum value, and the error signal  $\mathbf{e}_k$  is

$$\mathbf{e}_k = \mathbf{y}_d - \mathbf{y}_k = \begin{bmatrix} y_k^1 - y_d^1 \\ y_k^2 - y_d^2 \\ \vdots \\ y_k^p - y_d^p \end{bmatrix} = \begin{bmatrix} e_k^1 \\ e_k^2 \\ \vdots \\ e_k^p \end{bmatrix}.$$

## 4.2 Algorithm Design

The full gradient of  $G$  in (20) is  $\nabla_k = \nabla G(\mathbf{u}_k) = -\frac{1}{p} \mathcal{J}^T (\mathbf{y}_d - \mathcal{J}\mathbf{u}_k)$ . We need the information of  $\mathcal{J}^T$  to calculate the full gradient. However, in model-free learning, we want to obtain the gradient by conducting experiments on system  $\mathcal{J}$  only. For this purpose, Oomen et al. (2014) gives the following method to estimate  $\mathcal{J}^T$ .

**Lemma 1** For SISO system  $\mathcal{J} = J_{11}$ , its transpose  $\mathcal{J}^T$  can be obtained by matrix multiplication

$$(J_{11})^T = \mathcal{T}_N J_{11} \mathcal{T}_N,$$

where  $\mathcal{T}$  is the  $N$ -order permutation matrix whose anti-diagonal is 1, i.e.,

$$\mathcal{T}_N = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & & \vdots & \vdots \\ 1 & \dots & 0 & 0 \end{bmatrix}.$$

Therefore the full gradient of SISO system  $-\frac{1}{p} (J_{11})^T \mathbf{e}_k = -\frac{1}{p} \mathcal{T}_N J_{11} \mathcal{T}_N \mathbf{e}_k$  can be obtained by a single experiment.

**Lemma 2** For MIMO system  $\mathcal{J}$ , whose transpose  $\mathcal{J}^T$  is

$$\mathcal{J}^T = \begin{bmatrix} (J_{11})^T & \cdots & (J_{p1})^T \\ \vdots & \ddots & \vdots \\ (J_{1q})^T & \cdots & (J_{pq})^T \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{T}_N & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathcal{T}_N \end{bmatrix}}_{\mathcal{T}^{qN}} \underbrace{\begin{bmatrix} J_{11} & \cdots & J_{p1} \\ \vdots & \ddots & \vdots \\ J_{1q} & \cdots & J_{pq} \end{bmatrix}}_{\tilde{\mathcal{J}}} \underbrace{\begin{bmatrix} \mathcal{T}_N & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathcal{T}_N \end{bmatrix}}_{\mathcal{T}^{pN}}.$$

For symmetric MIMO systems,  $-\tilde{\mathcal{J}} \neq \mathcal{J}$ , so the full gradient  $-\frac{1}{p} \mathcal{J}^T \mathbf{e}_k$  of MIMO system cannot be obtained from a single experiment on system  $\mathcal{J}$ . The method proposed by Oomen et al. (2014) estimates  $\mathcal{J}^T$  from  $pq$  experiments:

$$\mathcal{J}^T = \mathcal{J}^{qN} \left( \sum_{i=1}^q \sum_{j=1}^p \mathcal{L}^{ij} \mathcal{J} \mathcal{L}^{ij} \right) \mathcal{J}^{pN}, \quad (21)$$

where  $\mathcal{L}^{ij}$  is a matrix consisting of  $q \times p$  blocks. In  $\mathcal{L}^{ij}$ , the  $(i, j)$  block is unit matrix of order  $N$ , and the remaining blocks are all 0:

$$\mathcal{L}^{ij} = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & I_N & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{qN \times pN}.$$

In (21), the left multiplication matrix  $\mathcal{L}^{ij}$  takes the  $i$ -th row of  $\mathcal{J}$ , and the right multiplication matrix  $\mathcal{L}^{ij}$  takes out the  $j$ -th column of  $\mathcal{J}$ . The two multiplications lead to a great loss of system information. We would like to improve the above method by extracting as much system information as possible. Therefore, consider the following decomposition as (9).

Set  $g_i(\mathbf{u}_k) = \frac{1}{2} \|\mathbf{e}_k^i\|^2 = \frac{1}{2} \left\| y_d^i - \sum_{j=1}^q J_{ij} \mathbf{u}_k^j \right\|^2$ , then (20) can be written as

$$G(\mathbf{u}_k) = \frac{1}{p} \sum_{i=1}^n g_i(\mathbf{u}_k). \quad (22)$$

Taking gradient on the both sides, we have

$$\nabla G(\mathbf{u}_k) = \frac{1}{p} \sum_{i=1}^n \nabla g_i(\mathbf{u}_k), \quad (23)$$

where

$$\nabla g_i(\mathbf{u}_k) = - \begin{bmatrix} J_{i1}^T e_k^i \\ J_{i2}^T e_k^i \\ \vdots \\ J_{iq}^T e_k^i \end{bmatrix} \in \mathbb{R}^{qN}.$$

Note that the  $\nabla g_i(\mathbf{u}_k)$  can be calculate by one line of the system matrix. The following lemma can help us design a controller to take out this information.

**Lemma 3** *Calculating  $\nabla g_i(\mathbf{u}_k)$  only needs a single experiment.*

**Proof** First, we note that

$$\underbrace{[0, \dots, 0, \mathcal{I}_N, 0, \dots, 0]}_{\mathcal{T}_i^{qN}} \underbrace{\begin{bmatrix} J_{11} & J_{12} & \cdots & J_{1q} \\ J_{21} & J_{22} & \cdots & J_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ J_{p1} & J_{p2} & \cdots & J_{pq} \end{bmatrix}}_{\mathcal{J}} \underbrace{\begin{bmatrix} \mathcal{T}_N & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathcal{T}_N \end{bmatrix}}_{\mathcal{T}^{qN}} = [J_{i1}^T, \dots, J_{iq}^T] \in \mathbb{R}^{N \times qN},$$

where  $\mathcal{T}_i^{pN} \in \mathbb{R}^{N \times pN}$  is the matrix whose  $i$ -th block is  $\mathcal{T}_N$  and the rest blocks are 0.  
For  $e_k^i \in \mathbb{R}^N$ ,

$$e_k^i = \underbrace{[0, \dots, 0, I_N, 0, \dots, 0]}_{\mathcal{L}_i} e_k,$$

where  $\mathcal{L}_i \in \mathbb{R}^{N \times qN}$  is the matrix whose  $i$ -th block is an identity matrix of order  $N$  and the rest blocks are 0.

The matrix multiplication method can retrieve a row of information of the system, but it cannot directly obtain the matrix for further computation. Therefore, simple and easy-to-implement linear mappings are considered to change the matrix to suitable dimension.

Set  $E_k^i \in \mathbb{R}^{qN \times q}$  and define a linear mapping  $\Psi$ :

$$\Psi e_k^i = E_k^i = \begin{bmatrix} e_k^i & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e_k^i \end{bmatrix},$$

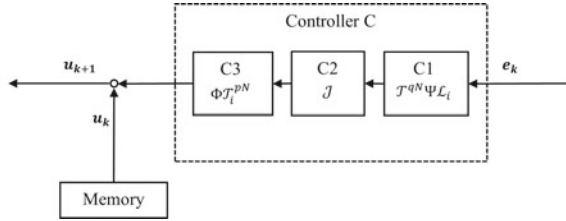
where  $E_k^i$  is the matrix blocked by  $N \times 1$ , with  $e_k^i$  on the diagonal and 0 in the rest of the blocks.

Since

$$\mathcal{T}_i^{pN} \mathcal{J} \mathcal{T}^{qN} E_k^i = [J_{i1}^T e_k^i, \dots, J_{iq}^T e_k^i] \in \mathbb{R}^{N \times q},$$

we can define the linear map  $\Phi: \mathbb{R}^{N \times q} \rightarrow \mathbb{R}^{qN}$ . It maps matrix in  $\mathbb{R}^{N \times q}$  to  $\mathbb{R}^{qN}$  by arranging each column of the matrix in order to a vector, i.e.,

**Fig. 4** Controller for model-free MIMO systems



$$\Phi [J_{i1}^T e_k^i, \dots, J_{iq}^T e_k^i] = \begin{bmatrix} J_{i1}^T e_k^i \\ J_{i2}^T e_k^i \\ \vdots \\ J_{iq}^T e_k^i \end{bmatrix}.$$

Combining above, we have

$$\nabla g_i(\mathbf{u}_k) = -\Phi \mathcal{J}_i^{pN} \mathcal{J} T^{qN} L_i e_k.$$

Thus,  $\nabla g_i(\mathbf{u}_k)$  can be calculated in a single experiment.  $\square$

Based on Lemma 3, a controller can be designed as shown in Fig. 4.

This controller can reduce the calculation of the full gradient in (21) from  $pq$  experiments to  $p$  experiments. However, when the system is noisy, there is no guarantee that the partial gradient estimated for each experiment  $\nabla g_i(\mathbf{u}_k)$  all correspond to the same full gradient  $\nabla G(\mathbf{u}_k)$ . In noisy systems, we can use the random gradient  $\tilde{\nabla}_k$ , which takes values uniformly  $\{\nabla g_i(\mathbf{u}_k)\}_{i=1}^N$  s.t.  $P(\tilde{\nabla}_k = \nabla g_i(\mathbf{u}_k)) = \frac{1}{N}$ . However, the SGD method converges slowly. Combining the convergence speed and the effect of noise in the system, we consider to design a SVRG-based ILC algorithm similar to Algorithm 1, as shown in Algorithm 3.

---

### Algorithm 3 Data-driven MIMO SVRG-based ILC

---

**Input:**  $\eta, \mathbf{u}_{0,0};$   
 $m \leftarrow 2p; \tilde{\mathbf{u}}^0 \leftarrow \mathbf{u}_{0,0};$   
**for**  $s \leftarrow 0$  to  $S-1$  **do**  
    $\mathbf{u}_{s,0} \leftarrow \tilde{\mathbf{u}}^s, \boldsymbol{\mu}_s \leftarrow \nabla G(\tilde{\mathbf{u}}^s);$   
   **for**  $k \leftarrow 0$  to  $m-1$  **do**  
      $\mathbf{w}_k \leftarrow \nabla g_i(\mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s) + \boldsymbol{\mu}_s;$  where  $i$  from  $\{1, 2, \dots, p\}$  randomly  
      $\mathbf{u}_{s,k+1} \leftarrow \mathbf{u}_{s,k} - \eta \mathbf{w}_k;$   
   **end for**  
    $\tilde{\mathbf{u}}^{s+1} \leftarrow \mathbf{u}_{s,m};$   
**end for**

---

Algorithm 3 uses Option II in Algorithm 1 to update the “snapshot”. If  $m$  is twice as many as  $p$ , a total of  $2p$  experiments are required for each internal iteration.

$p$  experiments are required to compute the full gradient, so a total of  $3p$  system experiments are required for each iteration. Compared with the SGD-based ILC, Algorithm 3 requires  $p$  more systematic experiments per  $m$  iterations to compute the full gradient in order to accelerate the convergence.

### 4.3 Convergence Analysis

From the following two propositions, we will see that  $G$  is not necessarily strongly convex.

**Proposition 4**  $G$  and  $g_i$  are convex and  $L$ -smooth.

*Proof* We note that

$$\nabla g_i(\mathbf{x}) = - \begin{bmatrix} J_{i1}^T \\ J_{i2}^T \\ \vdots \\ J_{iq}^T \end{bmatrix} \left( y_d^i - \sum_{j=1}^q J_{ij} x^j \right) = -J_i^T (y_d^i - J_i \mathbf{x}),$$

where  $J_i = [J_{i1} J_{i2} \dots J_{iq}]$ ,  $\mathbf{x} \in \mathbb{R}^{qN}$ ,  $g_i$  is convex, and

$$\begin{aligned} \|\nabla g_i(\mathbf{x}) - \nabla g_i(\mathbf{y})\| &= \|J_i J_i^T (\mathbf{x} - \mathbf{y})\| \\ &\leq \|J_i J_i^T\| \cdot \|\mathbf{x} - \mathbf{y}\| \leq \sum_{j=1}^q \|J_{ij}\|^2 \cdot \|\mathbf{x} - \mathbf{y}\|. \end{aligned}$$

Let  $L = \max_i \left\{ \sum_{j=1}^q \lambda_{ij} \right\}$ , where  $\lambda_{ij}$  is the maximum eigenvalue of  $J_{ij}^T J_{ij}$ . Since  $J_{ij}^T J_{ij}$  is always semipositive definite,  $\lambda_{ij} = 0$  if and only if  $J_{ij} = 0$ . Hence by Assumption 4,  $L > 0$ ,  $g_i$  is convex and  $L$ -smooth.

Since  $G$  is a convex combination of  $g_i$ ,  $G$  is convex and  $L$ -smooth.  $\square$

When  $p = q = 1$ ,  $\mathcal{J} = J_{11} = \text{diag}\{1, 0, \dots, 0\}$ ,  $G$  is not strong convex. The following proposition gives a sufficient condition for  $G$  to be strongly convex.

**Proposition 5** If the system matrix  $\mathcal{J}$  is of full column rank, then  $G$  is  $\sigma$ -strongly convex.

*Proof* By Assumption 5,  $p \geq q$ , and

$$G(\mathbf{u}_k) = \frac{1}{2p} (\mathbf{u}_k^T \mathcal{J}^T \mathcal{J} \mathbf{u}_k - \mathbf{y}_d^T \mathcal{J} \mathbf{u}_k - \mathbf{u}_k^T \mathcal{J}^T \mathbf{y}_d + \mathbf{y}_d^T \mathbf{y}_d).$$



Since  $\mathcal{J}^T \mathcal{J}$  is positive definite if and only if  $\mathcal{J}$  has full column rank. Therefore by Theorem 2.2,  $G(u_k)$  is  $\sigma$ -strongly convex when  $\mathcal{J}$  has full column rank, and the strong convexity factor is  $\frac{1}{2p}$  of the minimum eigenvalue of matrix  $\mathcal{J}^T \mathcal{J}$ .  $\square$

When  $G$  is strongly convex, we can prove that Algorithm 3 converges linearly similar to Algorithm 1 (Bottou et al., 2018). The convergence proof of Algorithm 3 under the strongly convex condition is omitted because of limited space. We only give the convergence proof under non-strongly convex.

First we have the following lemma (Reddi et al., 2016).

**Lemma 4** Assume that  $c_k, c_{k+1}, \beta > 0$ ,

$$c_k = c_{k+1} (1 + \eta\beta + 2\eta^2 L^2) + \eta^2 L^3.$$

If  $\eta, \beta$  and  $c_{k+1}$  are chosen such that

$$\mathbf{T}_k = \left( \eta - \frac{c_{k+1}\eta}{\beta} - \eta^2 L - 2c_{k+1}\eta^2 \right) > 0,$$

then each iteration of Algorithm 3 has an upper bound

$$\mathbb{E} \left[ \|\nabla G(\mathbf{u}_{s,k})\|^2 \right] \leq \frac{R_{s,k} - R_{s,k+1}}{\mathbf{T}_k},$$

where  $R_{s,k} \triangleq \mathbb{E} \left[ G(\mathbf{u}_{s,k}) + c_t \|\mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s\|^2 \right]$ .

**Proof** Since  $g_i$  is  $L$ -smooth,

$$g_i(\mathbf{u}_{s,k+1}) \leq g_i(\mathbf{u}_{s,k}) + \langle \nabla g_i(\mathbf{u}_{s,k}), \mathbf{u}_{s,k+1} - \mathbf{u}_{s,k} \rangle + \frac{L}{2} \|\mathbf{u}_{s,k+1} - \mathbf{u}_{s,k}\|^2.$$

For fixed  $s$ , let  $\mathbf{w}_k = \nabla g_i(\mathbf{u}_{s,k}) - \nabla g_i(\tilde{\mathbf{u}}^s) + \boldsymbol{\mu}_s$ , then  $\mathbb{E}[\mathbf{w}_k] = \nabla G(\mathbf{u}_{s,k})$ . Since  $\mathbf{u}_{s,k+1} - \mathbf{u}_{s,k} = -\eta \mathbf{w}_k$ , we use the above equation and take the expectation on both sides to obtain

$$\mathbb{E} [G(\mathbf{u}_{s,k+1})] \leq \mathbb{E} \left[ G(\mathbf{u}_{s,k}) - \eta \|\nabla G(\mathbf{u}_{s,k})\|^2 \right] + \frac{L\eta^2}{2} \mathbb{E} [\|\mathbf{w}_k\|^2]. \quad (24)$$

In addition, for  $\|\mathbf{u}_{s,k+1} - \tilde{\mathbf{u}}^s\|^2$ , we have

$$\begin{aligned}
& \mathbb{E} \left[ \left\| \mathbf{u}_{s,k+1} - \tilde{\mathbf{u}}^s \right\|^2 \right] \\
&= \mathbb{E} \left[ \left\| \mathbf{u}_{s,k+1} - \mathbf{u}_{s,k} \right\|^2 + \left\| \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \right\|^2 + 2 \langle \mathbf{u}_{s,k+1} - \mathbf{u}_{s,k}, \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \rangle \right] \\
&= \mathbb{E} \left[ \eta^2 \|\mathbf{w}_k\|^2 + \left\| \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \right\|^2 \right] - 2\eta \mathbb{E} \left[ \langle \nabla G(\mathbf{u}_{s,k}), \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \rangle \right] \\
&\leq \mathbb{E} \left[ \eta^2 \|\mathbf{w}_k\|^2 + \left\| \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \right\|^2 \right] \\
&\quad - 2\eta \mathbb{E} \left[ \frac{1}{2\beta} \|\nabla G(\mathbf{u}_{s,k})\|^2 + \frac{\beta}{2} \left\| \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \right\|^2 \right]. \tag{25}
\end{aligned}$$

In the above, we have used Young's inequality  $\langle x, y \rangle \leq \frac{1}{2\beta} \|x\|^2 + \frac{\beta}{2} \|y\|^2$ .

For  $\mathbb{E} [\|\mathbf{w}_k\|^2]$ , we have the following estimation:

$$\begin{aligned}
\mathbb{E} [\|\mathbf{w}_k\|^2] &= \mathbb{E} \left[ \left\| \nabla g_i(\mathbf{u}_{s,k}) - \nabla g_i(\tilde{\mathbf{u}}^s) + \nabla G(\tilde{\mathbf{u}}^s) \right\|^2 \right] \\
&= \mathbb{E} \left[ \left\| \nabla g_i(\mathbf{u}_{s,k}) - \nabla g_i(\tilde{\mathbf{u}}^s) + \nabla G(\tilde{\mathbf{u}}^s) - \nabla G(\mathbf{u}_{s,k}) + \nabla G(\mathbf{u}_{s,k}) \right\|^2 \right] \\
&\leq 2\mathbb{E} \left[ \left\| \nabla g_i(\mathbf{u}_{s,k}) - \nabla g_i(\tilde{\mathbf{u}}^s) - (\nabla G(\mathbf{u}_{s,k}) - \nabla G(\tilde{\mathbf{u}}^s)) \right\|^2 \right] \\
&\quad + 2\mathbb{E} \left[ \left\| \nabla G(\mathbf{u}_{s,k}) \right\|^2 \right] \\
&\leq 2\mathbb{E} \left[ \left\| \nabla g_i(\mathbf{u}_{s,k}) - \nabla g_i(\tilde{\mathbf{u}}^s) \right\|^2 \right] + 2\mathbb{E} \left[ \left\| \nabla G(\mathbf{u}_{s,k}) \right\|^2 \right] \\
&\leq 2L^2 \mathbb{E} \left[ \left\| \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \right\|^2 \right] + 2\mathbb{E} \left[ \left\| \nabla G(\mathbf{u}_{s,k}) \right\|^2 \right], \tag{26}
\end{aligned}$$

where the inequality is given by  $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ ,  $\mathbb{E} [\|\zeta - \mathbb{E}\zeta\|^2] = \mathbb{E}\|\zeta\|^2 - \|\mathbb{E}\zeta\|^2 \leq \mathbb{E}\|\zeta\|^2$  and the smoothness of  $g_i$ , i.e.,  $\|\nabla g_i(\mathbf{u}_{s,k}) - \nabla g_i(\tilde{\mathbf{u}}^s)\| \leq \|\mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s\|^2$ .

Denoting  $R_{s,k} \triangleq \mathbb{E} \left[ G(\mathbf{u}_{s,k}) + c_k \left\| \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \right\|^2 \right]$ , by (24) and (25), we have

$$\begin{aligned}
R_{s,k+1} &\leq \mathbb{E} \left[ G(\mathbf{u}_{s,k}) - \eta \left\| \nabla G(\mathbf{u}_{s,k}) \right\|^2 \right] + \frac{L\eta^2}{2} \mathbb{E} [\|\mathbf{w}_k\|^2] \\
&\quad + c_{k+1} \mathbb{E} \left[ \eta^2 \|\mathbf{w}_k\|^2 + \left\| \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \right\|^2 \right] \\
&\quad - 2c_{k+1}\eta \mathbb{E} \left[ \frac{1}{2\beta} \left\| \nabla G(\mathbf{u}_{s,k}) \right\|^2 + \frac{\beta}{2} \left\| \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \right\|^2 \right] \\
&\leq \mathbb{E} \left[ G(\mathbf{u}_{s,k}) - \eta \left( 1 - \frac{c_{k+1}}{\beta} \right) \left\| \nabla G(\mathbf{u}_{s,k}) \right\|^2 \right] \\
&\quad + \eta^2 \left( \frac{L}{2} + c_{k+1} \right) \mathbb{E} [\|\mathbf{w}_k\|^2] \\
&\quad + c_{k+1}(1 + \eta\beta) \mathbb{E} \left[ \left\| \mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s \right\|^2 \right]
\end{aligned}$$

From (26), we have

$$\begin{aligned} R_{s,k+1} &\leq \mathbb{E} [G(\mathbf{u}_{s,k})] + (c_{k+1} (1 + \eta\beta + 2\eta^2 L^3) + \eta^2 L^3) \mathbb{E} [\|\mathbf{u}_{s,k} - \tilde{\mathbf{u}}^s\|^2] \\ &\quad - \left( \eta - \frac{c_{k+1}\eta}{\beta} - L\eta^2 - 2c_{k+1}\eta^2 \right) \mathbb{E} [\|\nabla G(\mathbf{u}_{s,k})\|^2] \\ &= R_{s,k} - \left( \eta - \frac{c_{k+1}\eta}{\beta} - L\eta^2 - 2c_{k+1}\eta^2 \right) \mathbb{E} [\|\nabla G(\mathbf{u}_{s,k})\|^2]. \end{aligned}$$

Let  $T_k \triangleq \left( \eta - \frac{c_{k+1}\eta}{\beta} - \eta^2 L - 2c_{k+1}\eta^2 \right)$ ,  $T_k > 0$ , then

$$\mathbb{E} [\|\nabla G(\mathbf{u}_{s,k})\|^2] \leq \frac{R_{s,k} - R_{s,k+1}}{T_k}. \quad \square$$

Because of the complexity of non-strongly convex problem, we do not consider convergence criteria in Theorems 3 and 4 such as  $\mathbb{E} [G(\mathbf{u}) - G(\mathbf{u}^*)] \leq \varepsilon$ , but instead proving  $\mathbb{E} [\|\nabla G(\mathbf{u})\|^2] \leq \varepsilon$  for Algorithm 3. Note that if  $G$  is  $\sigma$ -strongly convex, it is easy to verify that

$$G(\mathbf{u}) - G(\mathbf{u}^*) \leq \frac{1}{2\sigma} \|\nabla G(\mathbf{u})\|^2.$$

Thus by  $\mathbb{E} [\|\nabla G(\mathbf{u})\|^2] \leq \varepsilon$ , we have  $\mathbb{E} [G(\mathbf{u}) - G(\mathbf{u}^*)] \leq \varepsilon$ . However, this relationship does not always hold under non-strongly convex case (Ghadimi & Lan, 2013). The following theorem gives the proof of the convergence  $\mathbb{E} [\|\nabla G(\mathbf{u})\|^2] \leq \varepsilon$  of Algorithm 3.

**Theorem 5** *Suppose each  $g_i$  is convex and  $L$ -smooth, and  $G$  is convex. For  $0 \leq k \leq m - 1$ ,  $c_k, c_{k+1}, \beta > 0$ ,  $c_m = 0$  satisfying*

$$c_k = c_{k+1} (1 + \eta\beta + 2\eta^2 L^2) + \eta^2 L^3.$$

and  $\eta, \beta, c_{k+1}$  are chosen such that

$$T_k = \left( \eta - \frac{c_{k+1}\eta}{\beta} - \eta^2 L - 2c_{k+1}\eta^2 \right) > 0.$$

Let  $\tau_m = \min_k T_k$ ,  $\mathbf{u}_a$  be a uniformly distributed random vector with values  $\{\mathbf{u}_{s,k} \mid 0 \leq s \leq S - 1, 0 \leq k \leq m - 1\}$ . Denote that  $\mathbf{u}^* = \operatorname{argmin}_{\mathbf{u}} G(\mathbf{u})$ , then Algorithm 3 satisfies

$$\mathbb{E} [\|\nabla G(\mathbf{u}_a)\|^2] \leq \frac{\mathbb{E} [G(\tilde{\mathbf{u}}^0) - G(\mathbf{u}^*)]}{Sm\tau_m}.$$

**Proof** We take  $k = 0, 1, \dots, m - 1$  in Lemma 1 and sum up to obtain

$$\sum_{k=0}^{m-1} \mathbb{E} \left[ \|\nabla G(\mathbf{u}_{s,k})\|^2 \right] \leq \frac{R_{s,0} - R_{s,m}}{\tau_m} = \frac{\mathbb{E} \left[ G(\tilde{\mathbf{u}}^s) - G(\tilde{\mathbf{u}}^{s+1}) \right]}{\tau_m}.$$

By the definition of  $R_{s,k}$ , we choose  $\mathbf{u}_{s,0} = \tilde{\mathbf{u}}^s$ ,  $\mathbf{u}_{s,m} = \tilde{\mathbf{u}}^{s+1}$  in  $\tilde{\mathbf{u}}^s$ . We take  $s = 0, 1, \dots, S-1$  in the above and sum up to get

$$\begin{aligned} \mathbb{E} \left[ \|\nabla G(\mathbf{u}_a)\|^2 \right] &= \frac{1}{Sm} \sum_{s=0}^{S-1} \sum_{k=0}^{m-1} \mathbb{E} \left[ \|\nabla G(\mathbf{u}_{s,k})\|^2 \right] \\ &\leq \frac{\mathbb{E} \left[ G(\tilde{\mathbf{u}}^0) - G(\tilde{\mathbf{u}}^S) \right]}{Sm\tau_m} \leq \frac{\mathbb{E} \left[ G(\tilde{\mathbf{u}}^0) - G(\mathbf{u}^*) \right]}{Sm\tau_m}. \end{aligned}$$

In the above, use the definition of  $\mathbf{u}_a$  and  $G(\tilde{\mathbf{u}}^s) \geq G(\mathbf{u}^*)$ .  $\square$

**Remark 7** Theorem 5 shows that the convergence of Algorithm 3 is  $O(1/Sm)$  under non-strongly convex conditions. The convergence of the corresponding SGD-based ILC under non-strongly convex conditions is  $O(1/\sqrt{Sm})$  (Johnson & Zhang, 2013)). Moreover, the theorem states that the convergence of Algorithm 3 is only related to the step size  $\eta$  but not to the choice of the number of iterations  $m$ . Since the system information is unknown,  $\eta$  needs to be estimated by experiment.

**Remark 8** Theorem 5 indicates that  $G(\mathbf{u}_k)$  can approach the optimal value  $G(\mathbf{u}^*) = (1/2p)\|\mathbf{y}_d - \mathcal{J}\mathbf{u}^*\|^2$ , i.e.,  $\mathbf{u}_k$  can converge to the optimal input  $\mathbf{u}^*$ . Similarly, the Assumption 2 can also be changed to  $\lim_{k \rightarrow \infty} F(\mathbf{u}_k) = F(\mathbf{u}^*)$  without affecting the convergence of Algorithm 1 and Algorithm 2. If  $F(\mathbf{u}^*) = 0$ , then  $\lim_{k \rightarrow \infty} F(\mathbf{u}_k) = 0$ .

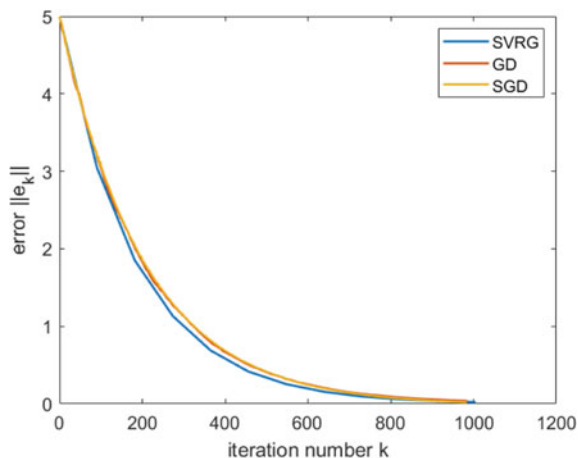
**Remark 9** System (21) degenerates to SISO system when both input and output are one dimension. At this point, the theorem indicates that when system (2) satisfies Assumption 4 (Assumption 1 need not to be satisfied), Algorithm 1 and Algorithm 2 still converge with updating the “snapshot” as Option II. But the convergence rate becomes  $O(1/Sm)$  when the objective function is not strongly convex.

#### 4.4 Numerical Simulation

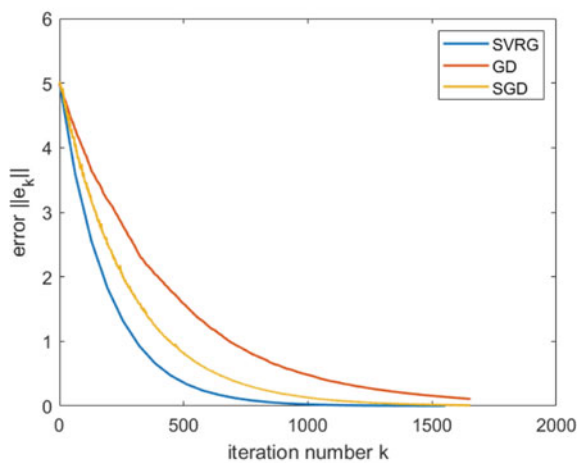
We take MIMO systems with  $21 \times 21$  input-output dimensions randomly generated by the Matlab `drss` function (Aarnoudse & Oomen, 2021) and set the time length to  $N = 42$ . The desired trajectory  $\mathbf{y}_d$  is 0.025 in each dimension. Model-free ILC based on full gradient (GD), stochastic gradient (SGD), and stochastic variance reduced gradient (SVRG) are performed in Fig. 5. We take the optimal step of each algorithm after multiple experiments.

As shown in Fig. 5a, the GD- and SVRG-based ILC converge similarly for deterministic systems, and both are faster than the SGD-based ILC algorithm.

**Fig. 5** Comparison of three data-driven gradient-based ILC under MIMO systems



(a) Deterministic MIMO system



(b) Noisy MIMO system

For randomly generated MIMO systems with input-output dimensions of  $21 \times 21$  and  $N = 42$ , we add Gaussian white noise to the system.

From Fig. 5b, we can see that both GD- and SGD-based ILC converge worse than SVRG type ILC when the system is noisy, and SVRG-based ILC can still maintain excellent convergence when the system is noisy.

## 5 Conclusions

This chapter focuses on exploring ILC based on the SVRG method. Firstly, Sect. 2 gives the basic framework of SVRG-based ILC and proves that the algorithm converges at a rate of  $O(\alpha^k)$  under smooth and strongly convex condition. Secondly, Sect. 3 designs a SVRG-based ILC algorithm to solve random error data dropouts and proves that the algorithm converges linearly. Finally, Sect. 4 constructs a model-free SVRG-based ILC by improving the existing model-free algorithm for MIMO systems and proves that the convergence rate is  $O(1/k)$  under smooth and convex condition. Compared to the GD- and SGD-based ILC, two numerical simulations in Sects. 3 and 4 verify that the SVRG-based ILC has superior convergence rate in both the random error dropouts and model-free contexts, respectively.

It should be noted that the SVRG-based ILC framework given in this chapter is not only applicable to the random error dropouts and model-free problems but can also be utilized to solve other error or system information deficient problems by properly decomposing the control objectives. Future research includes comparing its advantages and disadvantages with the stochastic approximation (SA) method, extending the framework to other information deficient problems, and attempting to develop algorithms with faster convergence based on this framework.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China (62173333), Beijing Natural Science Foundation (Z210002), and Research Fund of Renmin University of China (2021030187).

## References

- Aarnoudse, L., & Oomen, T. (2020). Model-free learning for massive MIMO systems: Stochastic approximation Adjoint iterative learning control. *IEEE Control Systems Letters*, 5(6), 1946–1951.
- Aarnoudse, L., & Oomen, T. (2021). Conjugate gradient MIMO iterative learning control using data-driven stochastic gradients. In *2021 60th IEEE Conference on Decision and Control (CDC)* (pp. 3749–3754).
- Allen-Zhu, Z. (2018). Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18, 1–51.
- Arimoto, S., Kawamura, S., & Miyazaki, F. (1984). Bettering operation of robots by learning. *The Journal of Intelligent and Robotic Systems*, 1(2), 123–140.
- Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, 60(2), 223–311.
- Ghadimi, S., & Lan, G. (2013). Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23, 2341–2368.
- Gu, P., Tian, S., & Chen, Y. (2019). Iterative learning control based on Nesterov accelerated gradient method. *IEEE Access*, 7, 115836–115842.
- Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 1, 315–323.

- Nesterov, Y. (2005). *Introductory lectures on convex programming volume: A basic course* (Vol. I). Kluwer Academic Publishers.
- Oomen, T., van der Maas, R., Rojas, C. R., & Hjalmarsson, H. (2014). Iterative data-driven H-infinity norm estimation of multivariable systems with application to robust active vibration isolation. *IEEE Transactions on Control Systems Technology*, 22(6), 2247–2260.
- Owens, D. H., Hatonen, J. J., & Daley, S. (2009). Robust monotone Gradient-based discrete-time iterative learning control. *The International Journal of Robust and Nonlinear Control*, 19(6), 634–661.
- Reddi, J. S., Hefny, A., Sra, S., Póczos, B., & Smola, A. (2016). Stochastic variance reduction for nonconvex optimization. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning* (Vol. 48, pp. 314–323).
- Shen, D. (2018). Iterative learning control with incomplete information: A survey. *IEEE/CAA Journal of Automatica Sinica*, 5(5), 885–901.
- Yang, X., & Ruan, X. (2017). Reinforced gradient-type iterative learning control for discrete linear time-invariant systems with parameters uncertainties and external noises. *IMA Journal of Mathematical Control and Information*, 34(4), 1117–1133.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

