Received July 25, 2019, accepted August 11, 2019, date of publication August 19, 2019, date of current version August 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936044

Iterative Learning Control Based on Nesterov Accelerated Gradient Method

PANPAN GU[1,2], SENPING TIAN[1], AND YANGQUAN CHEN[2]
[1]School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China
[2]School of Engineering (MESA-Lab), University of California at Merced, Merced, CA 95343, USA

Corresponding author: Senping Tian (ausptian@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61374104 and Grant 61773170, in part by the Natural Science Foundation of Guangdong Province under Grant 2016A030313505, and in part by the Scholarship from China Scholarship Council under Grant 201806150118.

ABSTRACT Based on Nesterov accelerated gradient method, the problem of iterative learning control for a class of linear discrete-time systems is considered in this paper. Firstly, the iterative learning control problem of linear discrete-time systems is transformed into an iterative least-squares problem. Then, the Nesterov accelerated gradient method is introduced into the iterative learning control framework. Note that the Nesterov accelerated gradient learning algorithm has the capability of fast convergence. It is shown that the algorithm presented in this paper can guarantee the output tracking error converges to zero with rate $O(1/k)$, where k is the iteration counter. Moreover, the monotonic convergence of the Nesterov accelerated gradient learning algorithm is analyzed and discussed. Finally, the effectiveness of the proposed method is verified by two simulation examples.

INDEX TERMS Iterative learning control, Nesterov accelerated gradient method, monotonic convergence, learning algorithm.

I. INTRODUCTION

As well known, iterative learning control (ILC) is an efficacious method to achieve perfect trajectory tracking for repetitive dynamic systems with complex modeling, uncertainty, and strong nonlinear coupling over a finite time interval (see [1]–[4]). The basic idea of ILC, that is inspired by human's learning capability, is to improve the tracking accuracy gradually by utilizing the previous control experience. On account of its simplicity and effectiveness, ILC has attracted extensive attention in theory and applications, and many significant achievements have been achieved in the past decades (see, e.g., [5]–[12] and references therein).

In ILC algorithm design, monotonic convergence should be considered first as an important issue, which means better and better. Recently, there are many efforts have been made on norm optimal ILC and parameter optimal ILC to improve the convergence rate. For example, in [13]–[16], the ILC algorithms were designed for discrete-time systems by using the quadratic performance index, which can ensure the tracking error converges monotonically to zero as the number of iterations increases. In [17]–[20], as effective optimization techniques, the Newton method and quasi-Newton method have been used to construct the optimal ILC laws. Moreover, a gradient-type ILC algorithm was designed in [21] for a class of linear discrete-time systems, then a complete analysis of the robust monotone convergence of the algorithm was presented with the help of necessary and sufficient matrix inequalities and frequency domain conditions. Based on [21] and [22], a combined inverse and gradient algorithm was developed in [23], which has a good convergence performance over the standard gradient-type algorithm. In [24], a reinforced gradient-type ILC algorithm was proposed for a class of linear discrete-time systems with model uncertainties and external bounded noises, where the ILC algorithm with symmetric learning gain matrix. It is noted that, however, the searching path of the gradient-type algorithm is a saw-tooth shape, which may lead to slow convergence speed and low efficiency. Fortunately, there exists an accelerated gradient method called as Nesterov accelerated gradient (NAG) method, which was proposed by Nesterov in [25] to minimize smooth convex functions. At each iteration, NAG method is used to evaluate point of the gradient and provides a larger and more timely correction to velocity. NAG scheme is

The associate editor coordinating the review of this article and approving it for publication was Dong Shen.

one of the most effective approaches among first-order optimization techniques. In recent years, many important works on first-order accelerated methods have been reported (see, e.g., [26]–[29] and references therein). Thus, it is beneficial to introduce the NAG method into the ILC framework. This motivates our present study.

In this paper, the ILC problem for a class of linear discrete-time systems is investigated based on NAG method. Then, a NAG learning algorithm is constructed for lifted systems. Under the action of the NAG learning algorithm, the convergence of the output tracking error is guaranteed with rate $O(1/k)$. Furthermore, the monotonic convergence of the proposed algorithm is analyzed and discussed. This paper is organized as follows. Section II gives the problem formulation of ILC in the form of the super-vector framework. In Section III, the NAG learning algorithm is developed and the corresponding convergence and monotonic convergence are analyzed, respectively. Numerical simulations are given to show the effectiveness of the proposed algorithm in section IV. Finally, a conclusion is drawn in Section V.

II. PROBLEM FORMULATION

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

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

where k denotes the iteration index and $t \in [0, N-1]$ is the discrete-time index. $x_k(t) \in R^n$, $u_k(t) \in R$ and $y_k(t) \in R$ represent the state, control input and output, respectively. A, B and C are real matrices with appropriate dimensions. The initial condition is the same for all iterations, i.e., $x_k(0) = x_0$, without loss of generality, the initial value is set as $x_0 = 0$. Assume that the system (1) has relative degree 1, i.e., $CB \neq 0$.

Taking $t = 0, 1, 2, \cdots, N-1$ in (1), and the relationships between the input and output can be expressed as follows:

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

Then, the system (1) can be written in an equivalent form

$$y_k = Gu_k, \tag{2}$$

where

$$\begin{aligned} u_k &= [u_k(0)\ u_k(1)\ \cdots\ u_k(N-1)]^{\mathrm{T}}, \\ y_k &= [y_k(1)\ y_k(2)\ \cdots\ y_k(N)]^{\mathrm{T}}, \end{aligned}$$

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

For a given desired trajectory $y_d = [y_d(1)\, y_d(2) \cdots y_d(N)]^{\mathrm{T}}$, our aim is to find the desired input u_d which is a solution of the equation $y_d = Gu$. Since the Markov matrix G is invertible under the assumption that $CB \neq 0$, there exists a unique u_d such that $y_d = Gu_d$, i.e., $u_d = G^{-1}y_d$. In practice, however, especially for the Markov matrix is ill-conditioned or large-scaled, inversion technique requires complex calculation process and is sensitive to the intervention of system parameters or the accumulation of calculation errors. Therefore, the control objective of this paper is to design an appropriate iterative learning algorithm to generate a control input sequence $\{u_k\}$ such that

$$\lim_{k\to\infty} ||e_k|| = 0, \quad \lim_{k\to\infty} ||u_d - u_k|| = 0,$$

where $e_k = y_d - y_k$ is the output tracking error and notation $|| \cdot ||$ denotes 2-norm of a vector and its compatible matrix norm.

The ILC problem of the lifted system (2) can be seen as equivalent to finding the minimizing input u_d for the following least-squares problem

$$\min_{u_k} J(u_k) = \frac{1}{2}||y_d - Gu_k||^2. \tag{3}$$

III. NESTEROV ACCELERATED GRADIENT LEARNING ALGORITHM

We can derive from (3) that the gradient of the function $J(u_k)$ with respect to u_k is $\nabla J(u_k) = -G^{\mathrm{T}}(y_d - Gu_k)$. Subsequently, its Hessian matrix is $H = \nabla^2 J(u_k) = G^{\mathrm{T}}G$ and H is a positive definite matrix, which means that the function $J(u_k)$ is convex. Note that the traditional gradient-type learning algorithm is constructed as

$$u_{k+1} = u_k + \beta G^{\mathrm{T}} e_k,$$

where $\beta > 0$ is the learning gain. In this paper, the ILC algorithm based on NAG method for the system (2) is designed as follows [25]:

$$\begin{cases} u_{k+1} = z_k - \omega \nabla J(z_k), \\ z_k = u_k + \dfrac{a_k - 1}{a_{k+1}}(u_k - u_{k-1}), \\ a_{k+1} = \dfrac{1 + \sqrt{4a_k^2 + 1}}{2}, \end{cases} \tag{4}$$

where $\omega \in R$ is the step size, let $u_{-1} = u_0$ and $a_0 = 1$.

For carrying out the analysis, we further give the following definitions.

Definition 1 [30]: A function f: $U \to R$ is said to be convex if U is convex set and if for all $x, y \in U$, and $0 \leq \alpha \leq 1$, we have

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y).$$

Definition 2 [8]: In higher-order ILC, the ILC process is monotonically convergent in an appropriate norm topology if $\|e_{k+1}\| < \max\{\|e_i\|, i = k, \cdots, k-n\}$.

Correspondingly, we have the following theorem.

Theorem 1: Consider the system (2) with the NAG learning algorithm (4). If we choose the step size ω such that $0 < \omega \le 1/||H||$, then the output tracking error $\|e_k\|$ converges to zero with rate $O(1/k)$.

Proof: For the function $J(u)$, there exists a $\xi = v + \theta(u-v)(0 < \theta < 1)$ such that

$$\begin{aligned} J(u) &= J(v) + [\nabla J(v)]^{\mathrm{T}}(u-v) + \frac{1}{2}(u-v)^{\mathrm{T}}\nabla^2 J(\xi)(u-v) \\ &= J(v) + [\nabla J(v)]^{\mathrm{T}}(u-v) + \frac{1}{2}(u-v)^{\mathrm{T}}H(u-v) \\ &\le J(v) + [\nabla J(v)]^{\mathrm{T}}(u-v) + \frac{1}{2}||H||||u-v||^2. \end{aligned}$$

Since $u_{k+1} = z_k - \omega\nabla J(z_k)$, we obtain

$$J(u_{k+1}) \le J(z_k) - \omega||\nabla J(z_k)||^2 + \tfrac{1}{2}\omega^2||H||||\nabla J(z_k)||^2.$$

Note that $\omega \le 1/||H||$, we further have

$$\begin{aligned} J(z_k) - J(u_{k+1}) &\ge \frac{1}{2}\omega(2-\omega||H||)||\nabla J(z_k)||^2 \\ &\ge \frac{1}{2}\omega||\nabla J(z_k)||^2. \end{aligned} \tag{5}$$

Denote $T_k = (a_k - 1)(u_{k-1} - u_k)$, then we derive

$$\begin{aligned} &T_{k+1} - u_{k+1} \\ &\quad = (a_{k+1}-1)(u_k - u_{k+1}) - u_{k+1} \\ &\quad = (a_{k+1}-1)u_k - a_{k+1}u_{k+1} \\ &\quad = (a_{k+1}-1)u_k - a_{k+1}(z_k - \omega\nabla J(z_k)) \\ &\quad = (a_{k+1}-1)u_k + \omega a_{k+1}\nabla J(z_k) - a_{k+1}\left(u_k - \frac{T_k}{a_{k+1}}\right) \\ &\quad = T_k - u_k + \omega a_{k+1}\nabla J(z_k). \end{aligned}$$

We further have

$$\begin{aligned} &||T_{k+1} - u_{k+1} + u_d||^2 \\ &\quad = ||T_k - u_k + u_d + \omega a_{k+1}\nabla J(z_k)||^2 \\ &\quad = ||T_k - u_k + u_d||^2 + \omega^2 a_{k+1}^2||\nabla J(z_k)||^2 \\ &\qquad + 2\omega a_{k+1}[\nabla J(z_k)]^{\mathrm{T}}(T_k - u_k + u_d) \\ &\quad = ||T_k - u_k + u_d||^2 + \omega^2 a_{k+1}^2||\nabla J(z_k)||^2 \\ &\qquad + 2\omega a_{k+1}[\nabla J(z_k)]^{\mathrm{T}}\left(T_k - z_k - \frac{T_k}{a_{k+1}} + u_d\right) \\ &\quad = ||T_k - u_k + u_d||^2 + \omega^2 a_{k+1}^2||\nabla J(z_k)||^2 \\ &\qquad + 2\omega a_{k+1}[\nabla J(z_k)]^{\mathrm{T}}(u_d - z_k) \\ &\qquad + 2\omega(a_{k+1}-1)[\nabla J(z_k)]^{\mathrm{T}}T_k. \end{aligned} \tag{6}$$

By mean of the expression (5) and the convexity of the function $J(u)$, we get

$$\begin{aligned} &J(u_{k+1}) - J(u_d) + \frac{1}{2}\omega||\nabla J(z_k)||^2 \\ &\quad \le J(u_{k+1}) - J(u_d) + J(z_k) - J(u_{k+1}) \\ &\quad = J(z_k) - J(u_d) \\ &\quad \le [\nabla J(z_k)]^{\mathrm{T}}(z_k - u_d). \end{aligned}$$

Note that $J(u_d) = 0$, thus

$$[\nabla J(z_k)]^{\mathrm{T}}(u_d - z_k) \le -J(u_{k+1}) - \frac{1}{2}\omega||\nabla J(z_k)||^2. \tag{7}$$

Since $z_k = u_k - \frac{T_k}{a_{k+1}}$, we have

$$\begin{aligned} J(z_k) &\le J(u_k) + [\nabla J(z_k)]^{\mathrm{T}}(z_k - u_k) \\ &= J(u_k) - \frac{1}{a_{k+1}}[\nabla J(z_k)]^{\mathrm{T}}T_k, \end{aligned}$$

it is obvious that

$$J(z_k) - J(u_{k+1}) \le J(u_k) - J(u_{k+1}) - \frac{1}{a_{k+1}}[\nabla J(z_k)]^{\mathrm{T}}T_k,$$

which together with (5) yields

$$\frac{1}{2}\omega||\nabla J(z_k)||^2 \le J(u_k) - J(u_{k+1}) - \frac{1}{a_{k+1}}[\nabla J(z_k)]^{\mathrm{T}}T_k,$$

which means that

$$[\nabla J(z_k)]^{\mathrm{T}}T_k \le a_{k+1}[J(u_k) - J(u_{k+1})] - \frac{1}{2}\omega a_{k+1}||\nabla J(z_k)||^2. \tag{8}$$

Substituting (7) and (8) into (6) becomes

$$\begin{aligned} &||T_{k+1} - u_{k+1} + u_d||^2 - ||T_k - u_k + u_d||^2 \\ &\quad = \omega^2 a_{k+1}^2||\nabla J(z_k)||^2 + 2\omega a_{k+1}[\nabla J(z_k)]^{\mathrm{T}}(u_d - z_k) \\ &\qquad + 2\omega(a_{k+1}-1)[\nabla J(z_k)]^{\mathrm{T}}T_k \\ &\quad \le \omega^2(a_{k+1}^2 - a_{k+1})||\nabla J(z_k)||^2 - 2\omega a_{k+1}J(u_{k+1}) \\ &\qquad + 2\omega(a_{k+1}-1)[\nabla J(z_k)]^{\mathrm{T}}T_k \\ &\quad \le 2\omega(a_{k+1}^2 - a_{k+1})[J(u_k) - J(u_{k+1})] - 2\omega a_{k+1}J(u_{k+1}) \\ &\quad = 2\omega(a_{k+1}^2 - a_{k+1})J(u_k) - 2\omega(a_{k+1}^2 - a_{k+1})J(u_{k+1}) \\ &\qquad - 2\omega a_{k+1}J(u_{k+1}) \\ &\quad = 2\omega a_k^2 J(u_k) - 2\omega a_{k+1}^2 J(u_{k+1}). \end{aligned}$$

Therefore, we have

$$\begin{aligned} &2\omega a_{k+1}^2 J(u_{k+1}) \\ &\quad \le 2\omega a_{k+1}^2 J(u_{k+1}) + ||T_{k+1} - u_{k+1} + u_d||^2 \\ &\quad \le 2\omega a_k^2 J(u_k) + ||T_k - u_k + u_d||^2 \\ &\quad \le 2\omega a_0^2 J(u_0) + ||T_0 - u_0 + u_d||^2 \\ &\quad \le 2\omega J(u_0) + ||u_0 - u_d||^2 \\ &\quad = \omega||y_d - Gu_0||^2 + ||u_0 - u_d||^2 \\ &\quad \le \omega||H||||u_0 - u_d||^2 + ||u_0 - u_d||^2 \\ &\quad \le 2||u_0 - u_d||^2. \end{aligned}$$

It is easy to yield that

$$||e_{k+1}||^2 = 2J(u_{k+1}) \le \frac{2||u_0 - u_d||^2}{\omega a_{k+1}^2}.$$

Note that

$$\begin{aligned} a_{k+1} &= \frac{1+\sqrt{4a_k^2+1}}{2} > \frac{1+\sqrt{4a_k^2}}{2} \\ &= a_k + \frac{1}{2} > a_0 + \frac{1}{2}(k+1) \\ &= \frac{1}{2}(k+3), \end{aligned}$$

correspondingly, we get

$$||e_{k+1}|| \leq \frac{\sqrt{2}||u_0 - u_d||}{\sqrt{\omega}a_{k+1}} < \frac{2\sqrt{2}||u_0 - u_d||}{\sqrt{\omega}(k+3)},$$

which means that NAG learning algorithm (4) can ensure to have an $O(1/k)$ convergence rate. Furthermore, we have

$$\lim_{k\to\infty} ||e_k|| = 0.$$

This completes the proof. ■

Remark 1: The result of theorem 1 can be extended to system with high relative degree. Assume that the system (1) has relative degree γ ($2 \leq \gamma \leq N$), i.e.,

$$CB = CAB = \cdots = CA^{\gamma-2}B = 0, \quad CA^{\gamma-1}B \neq 0.$$

Then, the system (1) can be described by the following lifted system form

$$y_k = Gu_k,$$

where

$$\begin{aligned} u_k &= [u_k(0)\ u_k(1)\ \cdots\ u_k(N-\gamma)]^{\mathrm{T}}, \\ y_k &= [y_k(\gamma)\ y_k(\gamma+1)\ \cdots\ y_k(N)]^{\mathrm{T}}, \\ G &= \begin{bmatrix} CA^{\gamma-1}B & 0 & \cdots & 0 \\ CA^{\gamma}B & CA^{\gamma-1}B & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{N-1}B & \cdots & CA^{\gamma}B & CA^{\gamma-1}B \end{bmatrix}. \end{aligned}$$

Let $y_d = [y_d(\gamma)\ y_d(\gamma+1)\ \cdots\ y_d(N)]^{\mathrm{T}}$ represent the desired trajectory. Similarly, we can conclude that $\|e_k\|$ converges to zero with rate $O(1/k)$ by using the NAG learning algorithm (4).

Now, we further analyze and discuss the monotonic convergence of the NAG learning algorithm.

Theorem 2: Consider the system (2) with the NAG learning algorithm (4). If the step size ω satisfying

$$\rho = \|I - \omega GG^{\mathrm{T}}\| \leq \frac{1}{3}, \tag{9}$$

then the output tracking error $\|e_k\|$ is monotonically convergent.

Proof: From the algorithm (4), we can obtain

$$\begin{aligned} u_{k+1} &= z_k - \omega\nabla J(z_k) = z_k + \omega G^{\mathrm{T}}(y_d - Gz_k) \\ &= u_k + \frac{a_k - 1}{a_{k+1}}(u_k - u_{k-1}) + \omega G^{\mathrm{T}}y_d - \omega G^{\mathrm{T}}Gz_k \\ &= u_k + \frac{a_k - 1}{a_{k+1}}(u_k - u_{k-1}) + \omega G^{\mathrm{T}}y_d - \omega G^{\mathrm{T}}Gu_k \\ &\quad - \frac{a_k - 1}{a_{k+1}}\omega G^{\mathrm{T}}G(u_k - u_{k-1}) \\ &= u_k + \frac{a_k - 1}{a_{k+1}}(I - \omega G^{\mathrm{T}}G)(u_k - u_{k-1}) \\ &\quad + \omega G^{\mathrm{T}}(y_d - Gu_k). \end{aligned} \tag{10}$$

Note that the matrix G is invertible, it is obvious from (2) that

$$u_k = G^{-1}y_k, \tag{11}$$

Substituting (11) into (10) becomes

$$\begin{aligned} G^{-1}y_{k+1} = G^{-1}y_k + \frac{a_k - 1}{a_{k+1}}(I - \omega G^{\mathrm{T}}G) \\ \times (G^{-1}y_k - G^{-1}y_{k-1}) + \omega G^{\mathrm{T}}(y_d - y_k) \end{aligned}$$

By inserting $y_k = y_d - e_k$ into the above expression, it yields

$$\begin{aligned} &G^{-1}(y_d - e_{k+1}) \\ &\quad = G^{-1}(y_d - e_k) + \frac{a_k - 1}{a_{k+1}}(I - \omega G^{\mathrm{T}}G) \\ &\qquad \times [G^{-1}(y_d - e_k) - G^{-1}(y_d - e_{k-1})] + \omega G^{\mathrm{T}}e_k. \end{aligned}$$

Furthermore, we get

$$\begin{aligned} G^{-1}e_{k+1} = G^{-1}e_k + \frac{a_k - 1}{a_{k+1}}(I - \omega G^{\mathrm{T}}G)G^{-1}(e_k - e_{k-1}) \\ - \omega G^{\mathrm{T}}e_k. \end{aligned} \tag{12}$$

Left multiplying both sides of (12) by matrix G, it gives

$$\begin{aligned} e_{k+1} &= e_k + \frac{a_k - 1}{a_{k+1}}(G - \omega GG^{\mathrm{T}}G)G^{-1}(e_k - e_{k-1}) \\ &\quad - \omega GG^T e_k \\ &= (I - \omega GG^{\mathrm{T}})e_k + \frac{a_k - 1}{a_{k+1}}(I - \omega GG^T)(e_k - e_{k-1}) \\ &= (I - \omega GG^{\mathrm{T}})e_k + \frac{a_k - 1}{a_{k+1}}(I - \omega GG^{\mathrm{T}})e_k \\ &\quad - \frac{a_k - 1}{a_{k+1}}(I - \omega GG^{\mathrm{T}})e_{k-1} \\ &= \frac{a_{k+1} + a_k - 1}{a_{k+1}}(I - \omega GG^{\mathrm{T}})e_k \\ &\quad - \frac{a_k - 1}{a_{k+1}}(I - \omega GG^{\mathrm{T}})e_{k-1}. \end{aligned} \tag{13}$$

Taking norm to the above expression and combining with (4) and (9), we get

$$\begin{aligned} \|e_{k+1}\| &\leq \frac{a_{k+1} + a_k - 1}{a_{k+1}}\|I - \omega GG^{\mathrm{T}}\|\|e_k\| \\ &\quad + \frac{a_k - 1}{a_{k+1}}\|I - \omega GG^{\mathrm{T}}\|\|e_{k-1}\| \\ &\leq \frac{a_{k+1} + 2a_k - 2}{a_{k+1}}\rho \max\{\|e_i\|, i = k, k-1\} \\ &< \left(1 + \frac{2a_k}{a_{k+1}}\right)\rho \max\{\|e_i\|, i = k, k-1\} \\ &= \left(1 + \frac{4a_k}{1 + \sqrt{4a_k^2 + 1}}\right)\rho \max\{\|e_i\|, i = k, k-1\} \\ &< \left(1 + \frac{4a_k}{1 + 2a_k}\right)\rho \max\{\|e_i\|, i = k, k-1\} \\ &< 3\rho \max\{\|e_i\|, i = k, k-1\}. \end{aligned}$$

Since $\rho \leq 1/3$, we have $\|e_{k+1}\| < \max\{\|e_i\|, i = k, k-1\}$. Based on definition 2, we know that $\|e_k\|$ is strictly monotonically convergent. This completes the proof. ■

Remark 2: In Theorem 2, a sufficient condition for monotone convergence of the algorithm (4) is given, but it is difficult to achieve the condition (9). The argument is as follows. Note that the matrix GG^{T} is positive definite, which means that all eigenvalues of GG^{T} are positive. Let λ_i be the ith eigenvalue of GG^{T}, then the eigenvalues of $I - \omega GG^{\mathrm{T}}$ are $1 - \omega\lambda_i$. Therefore, the condition (9) holds if we choose the step size ω such that $\frac{2}{3\omega} \leq \lambda_i \leq \frac{4}{3\omega}$.

IV. NUMERICAL SIMULATIONS

Two numerical examples are constructed in this section to demonstrate the effectiveness of the proposed NAG learning algorithm.

Example 1: Consider the following SISO discrete-time system:

$$\begin{cases} x_k(t+1) = \begin{bmatrix} 0.02 & 0 & 0.1 \\ 0 & 0.1 & 0 \\ 0 & 0.1 & 0.01 \end{bmatrix} x_k(t) + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} u_k(t), \\ y_k(t) = [1 \quad 0 \quad 1] x_k(t), \end{cases}$$

where $t \in [0, 7]$, and the initial state vector is set as $x_k(0) = 0$. It is easy to see that the relative degree of above system is 1. The desired output trajectory is taken as $y_d(t) = \frac{t}{2\pi}$, then

$$y_d = [\frac{1}{2\pi} \quad \frac{1}{\pi} \quad \frac{3}{2\pi} \quad \frac{2}{\pi} \quad \frac{5}{2\pi} \quad \frac{3}{\pi} \quad \frac{7}{2\pi} \quad \frac{4}{\pi}]^{\mathrm{T}}.$$

And the initial control is chosen as

$$u_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^{\mathrm{T}}.$$

Furthermore, we can compute that the eigenvalues of the matrix GG^{T} are 0.8158, 0.8384, 0.8789, 0.9399, 1.0211, 1.1164, 1.2109, 1.2822. For the NAG learning algorithm (4), take the step size $\omega = 1$ according to Remark 2, then the monotonic convergence of $||e_k||$ is guaranteed. Figure 1 gives the tracking situation of the output $y_k(t)$ to the desired trajectory $y_d(t)$ at the 2nd, 6th and 12th iterations, respectively. It is seen from Figure 2 that, as the iteration number increases, the output tracking error converges monotonically to zero in the sense of 2-norm. It is found that the NAG learning algorithm performs better than the gradient-type learning algorithm in the speed of convergence.

Example 2: Consider the following SISO discrete-time system:

$$\begin{cases} x_k(t+1) = \begin{bmatrix} 0.8 & 0.6 & 0.3 \\ 0 & 0.6 & 0.3 \\ 0.3 & 0 & 0.1 \end{bmatrix} x_k(t) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u_k(t), \\ y_k(t) = [1 \quad 1 \quad 1] x_k(t), \end{cases}$$

where $t \in [0, 7]$, and the initial state vector is set as $x_k(0) = 0$. Here, the relative degree of above system is 1. The desired output trajectory is taken as $y_d(t) = t$, then

$$y_d = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8]^{\mathrm{T}}.$$

Furthermore, the initial control is chosen as

$$u_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^{\mathrm{T}}.$$

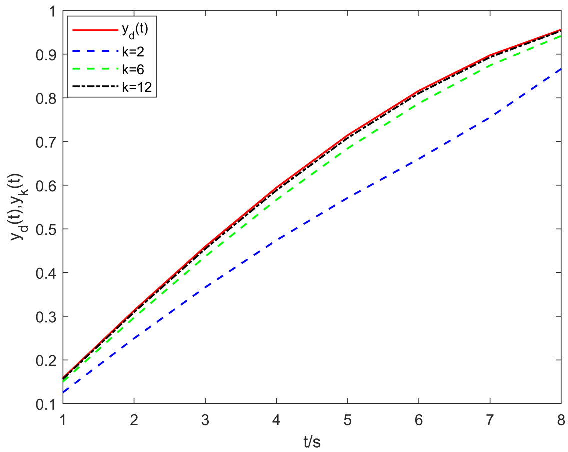


FIGURE 1. The output trajectories $y_d(t)$ and $y_k(t)$ at 2nd, 6th and 12th iterations by using the NAG learning algorithm.

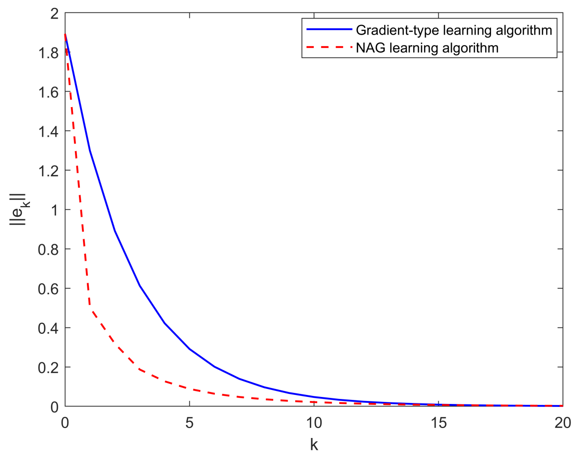


FIGURE 2. The maximum output tracking error with iterations.

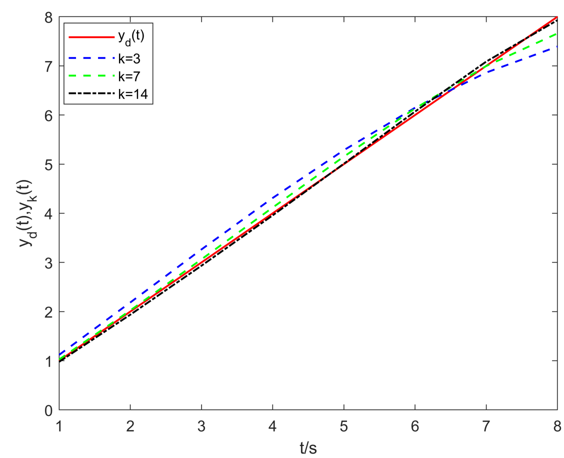


FIGURE 3. The output trajectories $y_d(t)$ and $y_k(t)$ at 3rd, 7th and 14th iterations by using the NAG learning algorithm.

Accordingly, the eigenvalues of the matrix GG^{T} are 2.5101, 2.7001, 3.085, 3.842, 5.5127, 10.1149, 28.9524, 331.9192. Obviously, the monotone convergence condition (9) is not satisfied. For the NAG learning algorithm (4), take the step size $\omega = 1/||H|| = 0.003$ according to Theorem 1, which

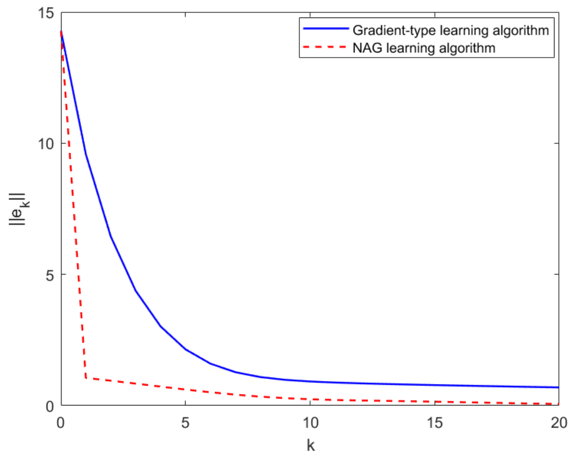


FIGURE 4. The maximum output tracking error with iterations.

means that $\|e_k\|$ can converge to zero with rate $O(1/k)$. Figure 3 shows that the output trajectory $y_k(t)$ at 14th iteration is close to the desired trajectory $y_d(t)$. We observe from Figure 4 that $||e_k||$ is monotonically convergent under the action of the NAG learning algorithm, and the convergence rate is much faster at the 1st iteration than the gradient-type learning algorithm.

V. CONCLUSION

In this paper, the ILC for a class of discrete-time systems is studied by using the NAG method. And the ILC problem for discrete-time systems is transformed into an optimization problem. Furthermore, the NAG learning algorithm is constructed and the convergence of the proposed algorithm is analyzed. We show that the algorithm can ensure the output tracking error converges to zero with rate $O(1/k)$ along the iteration axis. Besides, the monotonic convergence of the NAG learning algorithm is discussed and a sufficient convergence condition is established. In the end, two simulation examples are given to verify the theoretical results. Although the quasi-Newton-type ILC algorithms have been proposed in [19], [20], these algorithms are still often slow in practice. In future work, we will investigate the ILC problem for linear discrete-time systems based on Nesterov's accelerated quasi-Newton method [31].

REFERENCES

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, no. 2, pp. 123–140, 1984.

[2] K. L. Moore, *Iterative Learning Control for Deterministic Systems*. Berlin, Germany: Springer-Verlag, 1993.

[3] Y. Chen and C. Wen, *Iterative Learning Control: Convergence, Robustness and Applications*. London, U.K.: Springer-Verlag, 1999.

[4] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control*. Berlin, Germany: Springer-Verlag, 2003.

[5] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.

[6] H.-S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1099–1121, Nov. 2007.

[7] J.-X. Xu, "A survey on iterative learning control for nonlinear systems," *Int. J. Control*, vol. 84, no. 7, pp. 1275–1294, Jul. 2011.

[8] H.-S. Ahn, K. L. Moore, and Y. Chen, *Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems*. New York, NY, USA: Springer-Verlag, 2007.

[9] D. Shen and Y. Wang, "Survey on stochastic iterative learning control," *J. Process Control*, vol. 24, no. 12, pp. 64–77, Dec. 2014.

[10] M. Sun, S. S. Ge, and I. M. Y. Mareels, "Adaptive repetitive learning control of robotic manipulators without the requirement for initial repositioning," *IEEE Trans. Robot.*, vol. 22, no. 3, pp. 563–568, Jun. 2006.

[11] Z. Hou, J.-X. Xu, and H. Zhong, "Freeway traffic control using iterative learning control-based ramp metering and speed signaling," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 466–477, Mar. 2007.

[12] Y. Chen, K. L. Moore, J. Yu, and T. Zhang, "Iterative learning control and repetitive control in hard disk drive industry—A tutorial," *Int. J. Adapt. Control Signal Process.*, vol. 22, no. 4, pp. 325–343, May 2008.

[13] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proc.-Control Theory Appl.*, vol. 143, no. 2, pp. 217–224, 1996.

[14] D. H. Owens and B. Chu, "Modelling of non-minimum phase effects in discrete-time norm optimal iterative learning control," *Int. J. Control*, vol. 83, no. 10, pp. 2012–2027, Oct. 2010.

[15] D. H. Owens, B. Chu, and M. Songjun, "Parameter-optimal iterative learning control using polynomial representations of the inverse plant," *Int. J. Control*, vol. 85, no. 5, pp. 533–544, 2012.

[16] D. H. Owens, "Multivariable norm optimal and parameter optimal iterative learning control: A unified formulation," *Int. J. Control*, vol. 85, no. 8, pp. 1010–1025, 2012.

[17] T. Lin, D. H. Owens, and J. Hätönen, "Newton method based iterative learning control for discrete non-linear systems," *Int. J. Control*, vol. 79, no. 10, pp. 1263–1276, Oct. 2006.

[18] H.-F. Tao, W. Paszke, E. Rogers, K. Gałkowski, and H. Yang, "Modified Newton method based iterative learning control design for discrete nonlinear systems with constraints," *Syst. Control Lett.*, vol. 118, pp. 35–43, Aug. 2018.

[19] Y. Geng and X. Ruan, "Quasi-Newton-type optimized iterative learning control for discrete linear time invariant systems," *Control Theory Technol.*, vol. 13, no. 3, pp. 256–265, 2015.

[20] X. Wang, D. Cong, Z. Yang, S. Xu, and J. Han, "Modified Quasi-Newton optimization algorithm-based iterative learning control for multi-axial road durability test rig," *IEEE Access*, vol. 14, pp. 31286–31296, 2019.

[21] D. H. Owens, J. J. Hatonen, and S. Daley, "Robust monotone gradient-based discrete-time iterative learning control," *Int. J. Robust Nonlinear Control*, vol. 19, no. 6, pp. 634–661, 2009.

[22] T. J. Harte, J. Hatonen, and D. H. Owens, "Discrete-time inverse model-based iterative learning control: Stability, monotonicity and robustness," *Int. J. Control*, vol. 78, no. 8, pp. 577–586, May 2005.

[23] D. H. Owens and B. Chu, "Combined inverse and gradient iterative learning control: Performance, monotonicity, robustness and non-minimum-phase zeros," *Int. J. Robust Nonlinear Control*, vol. 24, no. 3, pp. 406–431, Feb. 2014.

[24] X. Yang and X. Ruan, "Reinforced gradient-type iterative learning control for discrete linear time-invariant systems with parameters uncertainties and external noises," *IMA J. Math. Control Info.*, vol. 34, no. 4, pp. 1117–1133, Dec. 2017.

[25] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Sov. Math. Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[26] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. New York, NY, USA: Springer-Verlag, 2004.

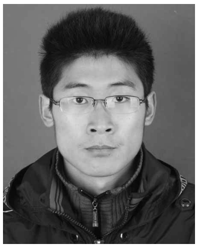
[27] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, Aug. 2013.

[28] W. Su, S. Boyd, and E. J. Candes, "A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights," *J. Mach. Learn. Res.*, vol. 17, no. 153, pp. 1–43, 2016.

[29] G. Qu and N. Li, "Accelerated distributed Nesterov gradient descent," May 2017, *arXiv:1705.07176*. [Online]. Available: https://arxiv.org/abs/1705.07176

[30] C. Niculescu and L. E. Persson, *Convex Functions and their Applications*. New York, NY, USA: Springer-Verlag, 2006.

[31] H. Ninomiya, "A novel Quasi-Newton-based optimization for neural network training incorporating Nesterov's accelerated gradient," *Nonlinear Theory Appl.*, vol. 8, no. 4, pp. 289–301, 2017.



PANPAN GU received the B.S. degree from Chaohu University, Hefei, China, in 2013, and the M.S. degree from the Suzhou University of Science and Technology, Suzhou, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Automation Science and Engineering, South China University of Technology. He is also an Exchange Ph.D. Student at the School of Engineering, University of California at Merced. His research interest includes iterative learning control.

SENPING TIAN received the B.S. and M.S. degrees from Central China Normal University, Wuhan, China, in 1982 and 1988, respectively, and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 1999, where he has been a Professor with the School of Automation Science and Engineering, since 2008. His research interests include theory and algorithms on iterative learning control for nonlinear systems, optimization and control of large-scale systems, and the stability and qualitative theory of differential equations.

YANGQUAN CHEN received the Ph.D. degree in advanced control and instrumentation from Nanyang Technological University, Singapore, in 1998. He was with the Faculty of Electrical and Computer Engineering, Utah State University, before he joined the School of Engineering, University of California at Merced, in 2012, where he teaches mechatronics for juniors and fractional-order mechanics for graduates. His research interests include mechatronics for sustainability, cognitive process control and hybrid lighting control, multi-UAV-based cooperative multi-spectral personal remote sensing and applications, applied fractional calculus in controls, signal processing and energy informatics, distributed measurement, and the distributed control of distributed parameter systems using mobile actuators and sensor networks.