ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2024, Vol. 85, No. 8, pp. 819–834.

State Observer-Based Iterative Learning Control Design for Discrete Systems Using the Heavy Ball Method

P. Pakshin[*,a], **J. Emelianova**[*,b], **and E. Rogers**[**,c]

[*] *Arzamas Polytechnic Institute of R.E. Alekseev Nizhny Novgorod State Technical University, Arzamas, Russia*
[**] *School of Electronics and Computer Science, University of Southampton, Southampton, United Kingdom*
e-mail: [a]*pakshinpv@gmail.com*, [b]*EmelianovaJulia@gmail.com*, [c]*etar@ecs.soton.ac.uk*

Received April 6, 2024
Revised June 27, 2024
Accepted July 1, 2024

Abstract—The paper considers a state observer-based iterative learning control design problem for discrete linear systems. To accelerate the convergence of the learning error, a combination of the heavy ball method from optimization theory and the vector Lyapunov function method for a class of two-dimensional systems known as repetitive processes is used to develop a new design. A supporting numerical example is given, including a comparison with an existing design.

Keywords: iterative learning control, repetitive processes, stability, convergence, state observer, heavy ball method, vector Lyapunov function, linear matrix inequalities

DOI: 10.31857/S0005117924080071

1. INTRODUCTION

Iterative learning control (ILC) arose from research aimed at increasing the accuracy of repetitive operations performed by robots [1], such as pick-and-place operations. Early results in this actively developing area can be found, e.g., in the survey papers [2, 3], with more recent results in, e.g., [4]. Currently, ILC designs are effectively used in additive manufacturing, in particular in high-precision multilayer laser deposition installation [5, 6], in medical robots for the rehabilitation of patients who have suffered a stroke [7, 8], in ventricular support devices [9], and in numerous other applications.

The class of systems that ILC can be applied to are those that iteratively execute the same finite-duration task. An example is a robot operating in a pick-and-place mode, where it collects a payload from a specified location, transfers it over a finite duration, deposits it, and then returns to the starting location to repeat this sequence of tasks. Each execution, or trial, is of a finite duration, termed the trial length. Once a trial is complete, all the information generated over the trial length is available to update the subsequent trial's control signal, highlighting the iterative nature of ILC. Batch processing is extensively used in chemical processing, where the same finite-duration operation is applied to a sequence of objects with a time gap between the end of processing one batch and the start of processing the next. Examples in this last area, generally known as batch processing, include manufacturing chemicals, pharmaceuticals, consumer products, and bioproducts. Since batch processes are repetitive, ILC can be applied, e.g., [10].

Consider an application where a reference trajectory is known. In the pick-and-place operation, this trajectory represents the ideal path for the robot to follow in each trial. The error between this trajectory and the trial's output forms a sequence of trial errors. The control design problem involves the construction of a sequence of trial inputs. These inputs are designed to guide the robot's movement so that the error sequence converges (in the trial number) to zero or within a specified tolerance. The response over the trial length can be assessed based on knowledge of the application under consideration on a particular trial.

As is well known, two-step methods such as the heavy ball method and the conjugate gradient method can significantly speed up the convergence of gradient descent algorithms in optimization problems [11–13]. In this paper, the problem investigated is the speed-up of the trial-to-trial error convergence through a control law where, on any trial, the input is an explicit function of the input used on the previous trial and other data from this trial. The analysis uses the heavy ball method [11], and the key idea here is the similarity of the structures of gradient descent and ILC.

Other strategies for using data from previous trials have been considered in the literature to speed up learning error convergence. In particular, there has been research on ILC design, where information from a finite number, greater than one, trials is used to generate the input for a subsequent trial; for example, see [14–20]. ILC laws that use control inputs from two or more previous trials to compute the input for the subsequent trial are termed higher-order laws. In these earlier studies, information from past trials is included using heuristic approaches. In general, these approaches proceed by first defining the structure of the ILC update law. Then, the parameters of this law are found using a trial-to-trial error convergence condition. Selecting the number of previous trials for the most significant effect is still an open research problem.

In [21], an ILC law based on Nesterov's method was described, but the convergence rate achieved was lower than that established in [22]. One reason for this is the use of a causal ILC law. In ILC systems, all data generated during a current trial are available to compute the control law for the subsequent trial. As one example, for discrete dynamics at a sample instant p, data generated on all previous trials at sample $p+\lambda$ can be used. Moreover, as is known [2], neglecting such information may impact the benefits of using an ILC law. The advantages of using higher-order laws are still an open question.

Throughout this paper, the notation for variables is of the form $h_k(p),\ 0 \leqslant p \leqslant N-1$, where h is the vector or scalar-valued variable under consideration, the integer $k \geqslant 0$ is the trial number, and N denotes the (finite) number of samples along the trial (N times the sampling period gives the trial length). Also, $\succ 0$, $\prec 0$, $\succeq 0$, and $\preceq 0$ denote, respectively, a symmetric positive define matrix, a symmetric negative definite matrix, a symmetric positive semi-definite matrix, and a symmetric negative semi-definite matrix.

2. PROBLEM FORMULATION

The systems considered are described by the following state-space model in the ILC setting on trial k :

$$\begin{aligned} x_k(p+1) &= Ax_k(p) + Bu_k(p), \\ y_k(p) &= Cx_k(p), \quad 0 \leqslant p \leqslant N-1, k \geqslant 0, \end{aligned} \tag{2.1}$$

where $x_k(p) \in \mathbb{R}^n$ is the state vector, and $u_k(p) \in \mathbb{R}$ and $y_k(p) \in \mathbb{R}$ are the input and output scalar variables, respectively. In the class of problems under consideration, the variable k is usually termed the trial number (trial for short), and $y_k(p)$ is the trial profile. The matrices A, B, and C are known and, by assumption, $CB \neq 0$. This assumption can be removed, e.g., using the result of [23] but is retained here for ease of presentation. The boundary conditions have the form $x_k(0) = x_0,\ k \geqslant 0$, where x_0 is given (on completing a trial, the system resets to the original state); $u_0(p)$ is given and bounded, and for $k \geqslant 1$, it is calculated using an updating law described below. Also, the generalization to multiple input and output systems with the same input and output dimensions is straightforward.

Let the system be completely controllable and observable, and let the variable $y_k(p)$ at each trial k be available for measurement. The state vector is estimated using a full-order asymptotic observer

$$\widehat{x}_k(p+1) = A\widehat{x}_k(p) + Bu_k(p) + G(y_k(p) - C\widehat{x}_k(p)),\ \widehat{x}_k(0) = x_0,\ 0 \leqslant p \leqslant N-1, k \geqslant 0, \quad (2.2)$$

where the matrix G is chosen by any known method such that the estimation error satisfies the condition

$$\widetilde{x}_k(p) = x_k(p) - \widehat{x}_k(p) \to 0$$

as $p \to \infty$.

Remark 1. Under the specified initial conditions, the observer accurately reconstructs the state vector. If x_0 is not available to the observer and or changes after a particular trial, the response of the estimation error will lead to a decrease in accuracy. This case requires further investigation and is left as an open research problem.

Let $y_{ref}(p),\ 0 \leqslant p \leqslant N-1,\ y_{ref}(0) = Cx_0$, denote a desired reference trajectory for (2.1). Then the error on trial k is

$$e_k(p) = y_{ref}(p) - y_k(p). \quad (2.3)$$

The ILC design problem is to construct a sequence of trial inputs $\{u_k\}$ such that for $0 \leqslant p \leqslant N-1$, the following conditions on the error $e_k(p)$ and input $u_k(p)$ hold:

$$|e_k(p)| \leqslant \varkappa\rho^k,\ \varkappa > 0,\ 0 < \rho < 1, \quad (2.4)$$

$$\lim_{k\to\infty} |u_k(p)| = |u_\infty(p)| < \infty, \quad (2.5)$$

where, in some parts of the literature, $u_\infty(p)$ is termed the learned control. In ILC applications, the trial length is finite; hence, it is possible to achieve trial-to-trial error convergence (in k) for unstable examples (in p). Two options then exist: a) design a stabilizing feedback control law and then apply the ILC law to the resulting controlled dynamics, or b) design a control law that simultaneously enforces trial-to-trial error convergence and regulates the dynamics along the trials. In the latter class of designs, the route is to treat ILC as a 2D system where the directions of information propagation are from trial to trial (k) and along the trial (p), respectively.

3. CONVERGENCE ANALYSIS AND DESIGN

The lifted model approach of ILC design for discrete-time systems involves the use of supervectors [14]. As the trial length is finite, the values of a variable over the trial length can be assembled into a column vector, e.g., for the input $\mathbf{U}_k = [u_k(0)\dots u_k(N-1)]^T$. As a result, trial-to-trial updating of variables can be expressed in standard difference equations. This approach can lead to matrices of very large dimensions. Moreover, feedback control must be applied in many cases to regulate the dynamics along a trial. An alternative represents the dynamics as a discrete repetitive process [24], a particular form of 2D systems. In this setting, it is possible to simultaneously design for trial-to-trial error convergence along the trial performance. This paper uses this latter setting for analysis and design, using the incremental variables

$$\begin{aligned}&\eta_{k+1}(p+1) = x_{k+1}(p) - x_k(p),\ \widehat{\eta}_{k+1}(p+1) = \widehat{x}_{k+1}(p) - \widehat{x}_k(p),\\ &\widetilde{\eta}_{k+1}(p+1) = \widetilde{x}_{k+1}(p) - \widetilde{x}_k(p),\ 0 \leqslant p \leqslant N-1, k \geqslant 0, \dots.\end{aligned} \quad (3.1)$$

Using (2.1), (2.2), (2.3) and these incremental variables gives the following description of the dynamics:

$$\begin{aligned} \widetilde{\eta}_{k+1}(p+1) &= (A-GC)\widetilde{\eta}_{k+1}(p), \\ \widehat{\eta}_{k+1}(p+1) &= GC\widetilde{\eta}_{k+1}(p) + A\widehat{\eta}_{k+1}(p) + B\Delta u_{k+1}(p-1), \\ e_{k+1}(p) &= -CGC\widetilde{\eta}_{k+1}(p) - CA\widehat{\eta}_{k+1}(p) - CB\Delta u_{k+1}(p-1) + e_k(p), \end{aligned} \tag{3.2}$$

where $\Delta u_{k+1}(p-1) = u_{k+1}(p-1) - u_k(p-1)$. In ILC design, the most common starting point is to specify that the input for the subsequent trial is the sum of the previous trial input plus a correction term (update law), where previous trial information is used. At the end of any trial, all information from all previous trials is available for use in constructing the control input for the subsequent trial. Many designs use only previous trial data, but the question is: would the use of a finite number, greater than one, of previous trials, be beneficial? This paper considers a control law that uses the control signals from the previous two trials and an optimization procedure to speed up the trial-to-trial error convergence.

The ILC law for (3.2) has the following structure:

$$u_{k+1}(p-1) = u_k(p-1) + \Delta u_{k+1}(p-1). \tag{3.3}$$

Hence, the problem is to find an update law $\Delta u_{k+1}(p-1)$ such that the convergence conditions (2.4) and (2.5) hold. Let this law be defined by

$$\Delta u_{k+1}(p-1) = \alpha\Delta u_k(p-1) + \beta\nabla\left(\frac{1}{2}e_{k+1}(p)^2\right) + \gamma[\Delta u_k(p-1) - \Delta u_{k-1}(p-1)], \tag{3.4}$$

where

$$\nabla\left(\frac{1}{2}e_{k+1}(p)^2\right) = \frac{\partial}{\partial\Delta u_{k+1}(p-1)}\left(\frac{1}{2}e_{k+1}(p)^2\right). \tag{3.5}$$

The update law (3.4) corresponds to the heavy ball method [11] in optimization theory. This simplest two-step method accelerates the convergence of gradient descent and, as will be shown, the same effect is achieved when the ILC law with the update term (3.4) is applied. Moreover, it is one option for using data beyond the previous trial in an ILC law. Other possibilities are discussed later in the paper.

Remark 2. In the original paper [11] $\alpha = 1$, but in the application considered (3.4) will be combined with the system dynamics (3.2) and, in contrast with [11], the argument $\Delta u_{k+1}(p-1)$ is parameterized by the variable p. Hence, the parameters α, β, and γ ensuring convergence may differ from those recommended in [11].

Due to the last equation of (3.2), the expression under the derivative sign in (3.5) can be written as

$$\begin{aligned} \frac{1}{2}e_{k+1}(p)^2 &= [-CGC\widetilde{\eta}_{k+1}(p) - CA\widehat{\eta}_{k+1}(p) - CB\Delta u_{k+1}(p-1) + e_k(p)]^T \\ &\times [-CGC\widetilde{\eta}_{k+1}(p) - CA\widehat{\eta}_{k+1}(p) - CB\Delta u_{k+1}(p-1) + e_k(p)]. \end{aligned}$$

Calculating the derivative of the right-hand side with respect to $\Delta u_{k+1}(p-1)$ gives

$$\nabla\left(\frac{1}{2}e_{k+1}(p)^2\right) = (CB)^2\Delta u_{k+1}(p-1) + CBCA\widehat{\eta}_{k+1}(p) + CBCGC\widetilde{\eta}_{k+1}(p) - CBe_k(p).$$

Substituting this expression into (3.4) after simple transformations, yields

$$\begin{aligned}\Delta u_{k+1}(p-1) = K_1\Delta u_k(p-1) + K_2CGC\widetilde{\eta}_{k+1}(p) + K_2CA\widehat{\eta}_{k+1}(p)\\ -K_2e_k(p) - K_3\Delta u_{k-1}(p-1),\end{aligned} \tag{3.6}$$

where

$$K_1 = \frac{\alpha+\gamma}{1-\beta(CB)^2},\ K_2 = \frac{\beta CB}{1-\beta(CB)^2},\ K_3 = \frac{\gamma}{1-\beta(CB)^2}.$$

Further, to apply the technique of linear matrix inequalities (LMIs), K_1, K_2, and K_3 will be used instead of the initial parameters α, β and γ. Introducing

$$\zeta_k(p) = \widetilde{\eta}_{k+1}(p),\ \xi_k(p) = \widehat{\eta}_{k+1}(p),\ \Delta\bar{u}_k(p) = \Delta u_k(p-1)$$

and substituting (3.6) into (3.2) give

$$\begin{aligned}
\zeta_k(p+1) &= (A-GC)\zeta_k(p),\\
\xi_k(p+1) &= (I+BK_2C)GC\zeta_k(p) + (I+BK_2C)A\xi_k(p)\\
&\quad + BK_1\Delta\bar{u}_k(p) - BK_3\Delta\bar{v}_k(p) - BK_2e_k(p),\\
\Delta\bar{u}_{k+1}(p) &= K_2CA\xi_k(p) + K_1\Delta\bar{u}_k(p) - K_3\Delta\bar{v}_k(p) - K_2e_k(p),\\
\Delta\bar{v}_{k+1}(p) &= \Delta\bar{u}_k(p),\\
e_{k+1}(p) &= -C(I+BK_2C)GC\zeta_k(p) - C(I+BK_2C)A\xi_k(p) - CBK_1\Delta\bar{u}_k(p)\\
&\quad + CBK_3\Delta\bar{v}_k(p) + (1+CBK_2)e_k(p).
\end{aligned} \tag{3.7}$$

In (3.7), the vectors $\zeta_k(p)$ and $\xi_k(p)$ evolve in p, the along-the-trial variable, and $\Delta\tilde{u}_k(p)$, $\Delta\bar{v}_k(p)$, and $e_k(p)$ evolve in k, the trial-to-trial variable. Systems with this type of dynamics are known as discrete linear repetitive processes [24]. These processes make a series of sweeps through dynamics defined over a finite duration. Once each sweep, termed a trial, is complete, the system resets to the starting location and is ready for the start of the subsequent trial. The unique feature is that the output on any trial contributes to the dynamics of the next one, which can result in oscillations that increase from trial to trial and cannot be controlled by standard control action. A stability theory for the case of linear dynamics has been developed [24] and applied to ILC design with experimental validation, e.g., [25].

As in stability analysis and control law design for standard linear systems theory, one approach to the same design problem for the dynamics considered in this paper would be to use a suitably constructed Lyapunov function, which should be constructed from the sum of terms in the trial-to-trial propagation and the along-the-trial dynamics, respectively. But the two equations in (3.7) are coupled and hence the gradient of the Lyapunov function can only be found if the solutions to these equations are available, which is a very stringent requirement, and instead a vector Lyapunov function must be used and the gradient replaced by the divergence, see [26] for the required background. (In the case of linear dynamics, the stability analysis methods in [24] and [26] are equivalent.)

In this paper, the vector Lyapunov function for dynamics described by (3.7) has the form

$$V(\bar{\xi}_k(p), \bar{e}_k(p)) = \begin{bmatrix} V_1(\bar{\xi}_k(p)) \\ V_2(\bar{e}_k(p)) \end{bmatrix}, \tag{3.8}$$

where $\bar{\xi}_k(p) = [\zeta_k(p)^T \ \xi_k(p)^T]^T$, $\bar{e}_k(p) = \begin{bmatrix} \Delta\bar{u}_k(p) & \Delta\bar{v}_k(p) & e_k(p) \end{bmatrix}^T$, $V_1(\bar{\xi}_k(p)) > 0$, $\bar{\xi}_k(p)) \neq 0$, $V_2(\bar{e}_k(p)) > 0$, $\bar{e}_k(p) \neq 0$, $V_1(0) = 0$, $V_2(0) = 0$. Also, let a discrete counterpart of the divergence operator of this function along the trajectories of system (3.7) be defined as

$$\mathcal{D}V(\bar{\xi}_k(p), \bar{e}_k(p)) = V_1(\bar{\xi}_k(p+1)) - V_1(\bar{\xi}_k(p)) + V_2(\bar{e}_{k+1}(p)) - V_2(\bar{e}_k(p)). \tag{3.9}$$

The following result can be established.

Theorem 1. *Assume that there exists a vector Lyapunov function* (3.8) *and positive scalars* c_1, c_2, *and* c_3 *such that the following inequalities are valid on the trajectories of the system* (3.7) (*where* $||\cdot||$ *denotes the Euclidean norm*):

$$c_1||\bar{\xi}_k(p)||^2 \leqslant V_1(\bar{\xi}_k(p)) \leqslant c_2||\bar{\xi}_k(p)||^2, \tag{3.10}$$

$$c_1||\bar{e}_k(p)||^2 \leqslant V_2(\bar{e}_k(p)) \leqslant c_2||\bar{e}_k(p)||^2, \tag{3.11}$$

$$\mathcal{D}V(\bar{\xi}_k(p), e_k(p)) \leqslant -c_3(||\bar{\xi}_k(p)||^2 + ||\bar{e}_k(p)||^2). \tag{3.12}$$

Then conditions (2.4) *and* (2.5) *hold.*

Proof. In [26] (Theorem 1) it was shown that if (3.10), (3.11), and (3.12) are satisfied, then

$$||\bar{\xi}_k(p)||^2 + ||\bar{e}_k(p)||^2 \leqslant \bar{\varkappa}\varrho^k, \ 0 \leqslant p \leqslant N-1, \ k \geqslant 0, \tag{3.13}$$

where $\bar{\varkappa}$ and $0 < \varrho < 1$ depend on c_1, c_2, and c_3. Also it follows from (3.13) and the definition of $\bar{e}_k(p)$ that

$$|e_k(p)|^2 \leqslant \bar{\varkappa}\varrho^k, \ \ |\Delta\bar{u}_k(p)|^2 \leqslant \bar{\varkappa}\varrho^k. \tag{3.14}$$

Due to the first inequality in (3.14), condition (2.4) holds with $\varkappa = \bar{\varkappa}^{\frac{1}{2}}$ and $\rho = \varrho^{\frac{1}{2}}$. The boundedness of control (2.5) follows from the obvious relation

$$|u_k(p)| \leqslant |u_{k-1}(p)| + |\Delta\bar{u}_k(p)|, \ \ k = 1, 2, \dots,$$

and the boundedness of $u_0(p)$ and the second inequality in (3.14).

Introduce the following matrices and vectors:

$$\bar{A} = \begin{bmatrix} A-GC & 0 & 0 & 0 & 0 \\ GC & A & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -CGC & -CA & 0 & 0 & 1 \end{bmatrix}, \ \bar{B} = \begin{bmatrix} 0 \\ B \\ 1 \\ 0 \\ -CB \end{bmatrix}, \ \bar{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ CGC & CA & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix}.$$

Then the following result enables LMI based ILC design.

Theorem 2. *Suppose that for some matrix* $Q \succ 0$ *and a positive scalar* R,

$$\begin{bmatrix} X & (\bar{A}X + \bar{B}Y\bar{C})^T & X & (Y\bar{C})^T \\ \bar{A}X + \bar{B}Y\bar{C} & X & 0 & 0 \\ X & 0 & Q^{-1} & 0 \\ Y\bar{C} & 0 & 0 & R^{-1} \end{bmatrix} \succeq 0, \ \bar{C}X = Z\bar{C}, \tag{3.15}$$

is feasible for $X = \mathrm{diag}[X_1 \ X_2]$, Y, Z, *where* X_1 *and* X_2 *are positive definite matrices of dimensions* $2n \times 2n$ *and* $3n \times 3n$, *respectively. Then the ILC law*

$$\begin{aligned} u_{k+1}(p) &= u_k(p) + \Delta u_{k+1}(p), \\ \Delta u_{k+1}(p) &= K_1\Delta u_k(p) + K_2CA[\widehat{x}_{k+1}(p) - \widehat{x}_k(p)] \\ &\quad + K_2CG[(y_{k+1}(p) - y_k(p)) - (\widehat{y}_{k+1}(p) - \widehat{y}_k(p)] \\ &\quad - K_2[y_{ref}(p+1) - Cx_k(p+1)] - K_3\Delta u_{k-1}(p), \end{aligned} \tag{3.16}$$

where

$$K = \begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix} = YZ^{-1},$$

ensures that the required conditions (2.4) *and* (2.5) *hold.*

Proof. Applying the Schur's complement formula to inequality (3.15) gives

$$X - (\bar{A}X + \bar{B}Y\bar{C})^T X^{-1}(\bar{A}X + \bar{B}Y\bar{C}) - XQX - (Y\bar{C})^T R(Y\bar{C}) \succeq 0.$$

Further, using $Y = KZ$ and $CX = ZC$, this last inequality can be written in the form

$$X - (\bar{A}X + \bar{B}K\bar{C}X)^T X^{-1}(\bar{A}X + \bar{B}K\bar{C}X) - XQX - (K\bar{C}X)^T R(K\bar{C}X) \succeq 0.$$

Multiplying the last inequality by $P = X^{-1}$ on the left and right, respectively, and then rearranging the terms give

$$(\bar{A} + \bar{B}K\bar{C})^T P(\bar{A} + \bar{B}K\bar{C}) - P + Q + (K\bar{C})^T R(K\bar{C}) \preceq 0. \tag{3.17}$$

Let the entries in the vector Lyapunov function (3.8) be chosen as the quadratic forms

$$\begin{aligned} V_1(\bar{\xi}_k(p)) &= \bar{\xi}_k^T(p)P_1\bar{\xi}_k(p), \\ V_2(\bar{e}_k(p)) &= \bar{e}_k^T(p)P_2\bar{e}_k(p), \end{aligned} \tag{3.18}$$

where $P_1 \succ 0$ and $P_2 \succ 0$. Then calculating the divergence (3.9) along the trajectories of system (3.7) gives that conditions (3.10), (3.11), and (3.12) will be satisfied if

$$(\bar{A} + \bar{B}K\bar{C})^T P(\bar{A} + \bar{B}K\bar{C}) - P \prec 0. \tag{3.19}$$

Finally, (3.19) holds due to (3.17) and (3.16) follows from (3.6) on using (3.1) and (2.3).

Remark 3. The system (3.15) is similar in structure to the relations of the linear-quadratic regulator (LQR) problem in LMI formulation. The matrix $K = \begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix}$ depends on Q and R, which are selected based on achieving the desired rate of convergence of the learning error. As in the LQR problem, there is no complete formalization of such a choice and a heuristic approach based on knowledge of the dynamics of the system under consideration is recommended.

For further comparative analysis, consider the most commonly used update law, see, for example, [27]:

$$\Delta u_k(p) = \hat{K}_1\hat{\eta}_k(p) + \hat{K}_2 e_{k-1}(p+1). \tag{3.20}$$

Substituting (3.20) in (3.2) and applying Theorem 1 with the entries in (3.8) chosen as

$$\begin{aligned} V_1(\bar{\xi}_k(p)) &= \bar{\xi}_k^T(p)\hat{P}_1\bar{\xi}_k(p), \\ V_2(e_k(p)) &= e_k^T(p)\hat{P}_2 e_k(p), \end{aligned} \tag{3.21}$$

give the divergence

$$\mathcal{D}V(\bar{\xi}_k(p), e_k(p)) = [\bar{\xi}_k^T(p)\ e_k(p)][(\hat{A} + \hat{B}K\hat{C})^T \hat{P}(\hat{A} + \hat{B}K\hat{C}) - \hat{P}][\bar{\xi}_k^T(p)\ e_k(p)]^T, \tag{3.22}$$

where $\hat{P} = \operatorname{diag}[\hat{P}_1\ \hat{P}_2]$, and

$$\hat{A} = \begin{bmatrix} A - GC & 0 & 0 \\ GC & A & 0 \\ -CGC & -CA & 1 \end{bmatrix}, \ \hat{B} = \begin{bmatrix} 0 \\ B \\ -CB \end{bmatrix} \ \hat{C} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \hat{K} = [\hat{K}_1\ \hat{K}_2].$$

Repeating the derivations from the proof of Theorem 2 gives that if

$$\begin{bmatrix} \hat{X} & (\hat{A}\hat{X}+\hat{B}\hat{Y}\hat{C})^T & \hat{X} & (\hat{Y}\hat{C})^T \\ \hat{A}\hat{X}+\hat{B}\hat{Y}\hat{C} & \hat{X} & 0 & 0 \\ \hat{X} & 0 & \hat{Q}^{-1} & 0 \\ \hat{Y}\hat{C} & 0 & 0 & \hat{R}^{-1} \end{bmatrix} \succeq 0, \; \hat{C}\hat{X} = \hat{Z}\hat{C} \tag{3.23}$$

has a solution $\hat{X} = \mathrm{diag}[\hat{X}_1 \; \hat{X}_2]$, $\hat{Y}$, $\hat{Z}$, where $\hat{X}_1 \succ 0$ is a $2n \times 2n$ matrix and $\hat{X}_2$ is a positive scalar, then the ILC law with the update term (3.20) and the matrix $\hat{K} = \hat{Y}\hat{Z}^{-1}$ ensures conditions (2.4) and (2.5).

Remark 4. The approach based on the heavy ball method leads to a new structure of the control law (3.16), which differs from that in the known literature. If the parameter values ensure conditions (2.4) and (2.5), then the update term, in accordance with (3.4), changes along the gradient of the quadratic function of this term. In the case of the update law (3.20), this cannot be guaranteed. Apparently, this property provides faster convergence of the learning error when using the control law (3.16), but rigorous proof has not yet been obtained.

4. NUMERICAL CASE STUDY

In, e.g., [25], a gantry robot, specifically designed and constructed to emulate the pick-and-place operation to which ILC is applicable, has been used to verify ILC designs experimentally. This system has three mutually perpendicular axes, one of which X is directed along the conveyor belt, and the second Y is perpendicular to this axis in the conveyor plane. The third Z is perpendicular to the XY plane. The system is designed such that the interaction between axes can be neglected, and models for control design have been obtained from frequency response tests.

Figure 1 shows the 3D reference signal used in experimental tests; the trial length is 2 s.

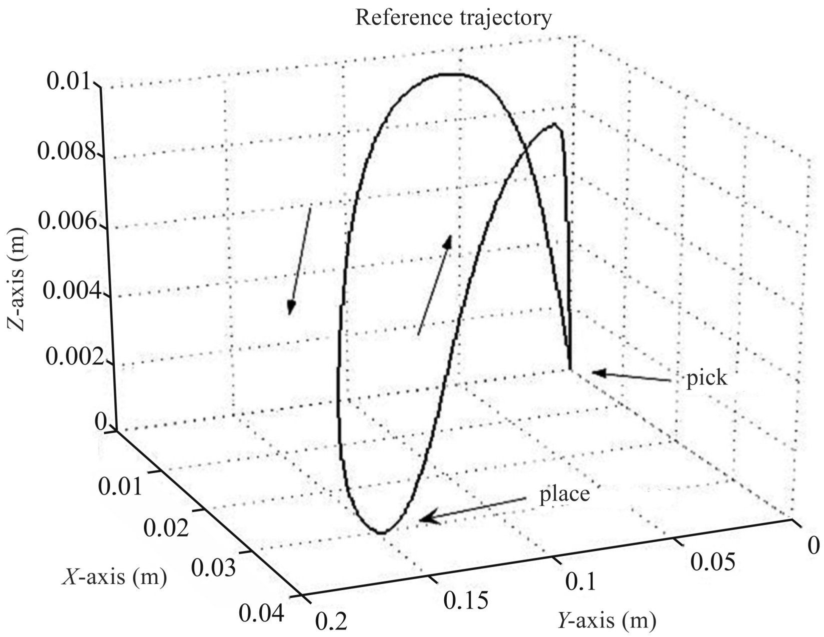

Fig. 1. The 3D reference trajectory.

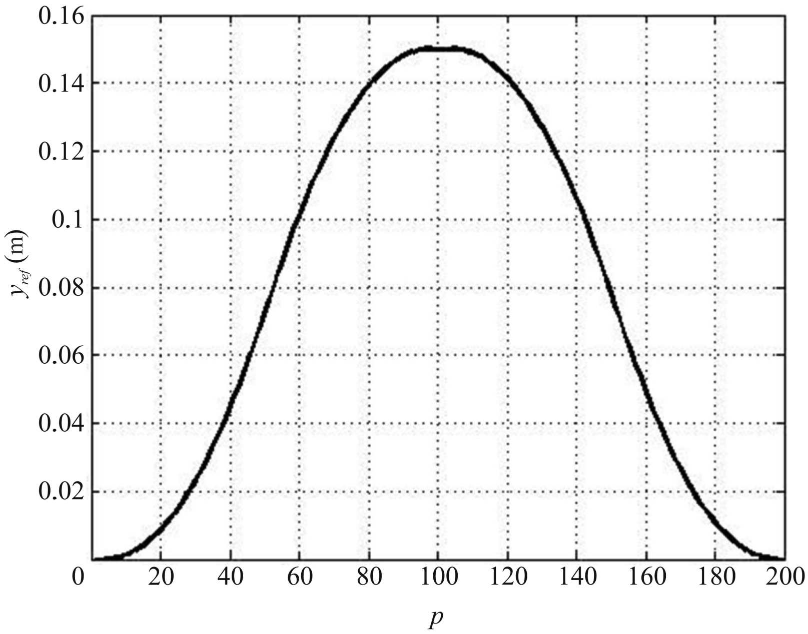

Fig. 2. The reference trajectory for the Y axis.

The transfer function for the Y axis is

$$G_Y(s) = \frac{23.736(s + 661.2)}{s(s^2 + 426.7s + 1.744 \times 10^5)}, \tag{4.1}$$

and the reference trajectory for this axis is shown in Fig. 2. The sampling period for discrete implementation is $T_s = 0.01$ s. The discrete model has been constructed using the standard functions `ss` and `c2d` of MATLAB. The corresponding matrices are

$$A = \begin{bmatrix} -0.0762 & 0.0487 & 0 \\ -0.0732 & -0.1372 & 0 \\ 0.0033 & 0.0026 & 1.0000 \end{bmatrix}, \ B = \begin{bmatrix} -0.0006 \\ 0.0134 \\ 0.0001 \end{bmatrix}, \ C = [0 \ \ 0.0116 \ \ 7.6631].$$

Evaluation of the performance of an ILC design must consider trial-to-trial error convergence and performance along the trials. The latter of these can be evaluated using standard measures for each trial. For trial-to-trial error convergence, one commonly used measure is the root mean square learning error on each trial plotted against the trial number, where on trial k,

$$\mathrm{E}(k) = \sqrt{\frac{1}{N}\sum_{p=0}^{N-1} |e_k(p)|^2}. \tag{4.2}$$

The observer parameters have been found using the results of modal control theory, implemented using the standard function `place` of MATLAB. Assigning the eigenvalues $\lambda_1 = 0.9$ and $\lambda_{2,3} = -0.1 \pm 0.01i$ for the matrix $(A - GC)$ and applying this function give

$$G = [-0.3070 \ \ 2.6427 \ \ 0.0073]^T.$$

According to the problem formulation, under the assumptions made, any observer matrix G for which the eigenvalues of $(A - GC)$ are inside the unit circle accurately reconstructs the state vector. Therefore, the eigenvalues have been assigned to make the observer gains (the entries in the matrix G) relatively small.

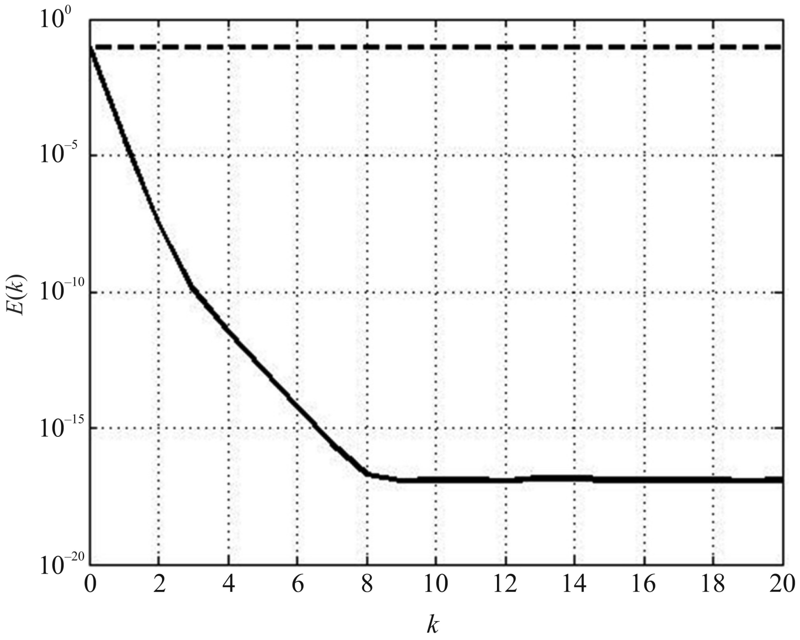

Fig. 3. Progression of $E(k)$ for the Y axis with $Q = I$ and $R = 1$ for (3.16) (solid line) and (3.20) (dashed line).

Selecting $Q = I$, i.e., the identity matrix of compatible dimensions and $R = 1$ and applying Theorem 2 give

$$K_1 = -1.09 \times 10^{-7}, \quad K_2 = -1196.0, \quad K_3 = -1.61 \times 10^{-8}.$$

For this case, the ILC law (3.16) reduces the learning error by 10^5 times in one trial, and after 8 trials, this error has reached numerical zero. For comparison, the ILC algorithm with the update law (3.20) gives an error reduction of 1% for 70 trials (Fig. 3).

In this case, the control law parameters K_1 and K_3 are very small, and numerical investigations demonstrate that a minimal change will occur if they are set to zero. Hence, the ILC law with the update (3.20) in which $\hat{K}_1 = K_2CA$ and $\hat{K}_2 = -K_2$ is obtained. Thus, applying the heavy ball method imposes restrictions on $\hat{K}_1$ and $\hat{K}_2$, under which a high trial-to-trial error convergence rate is ensured. It is, however, difficult to use this feature to provide rules for choosing Q and R for a given example. Establishing a connection between the convergence rate of the learning error and Q and R is an open problem. (It is possible to select matrices providing approximately a 10 times lower convergence rate but at the cost of a significant computation time.) Considering this feature, reducing the convergence rate by decreasing K_2 and checking the LMI (3.19) is possible. In practice, this may be important if actuator saturation is possible, leading to the need for a compromise solution.

For $K_2 = -50$, the convergence rate decreases by approximately 80 times relative to the result obtained for unit Q and R in (3.15), however, this speed remains higher than for $Q = I$ and $R = 1$ in (3.23) (Fig. 4); the control law matrices in the case of $K_2 = -50$ and $K_1 = K_3 = 0$ are

$$\hat{K}_1 = [-1.24 \ -0.93 \ -383.2], \quad \hat{K}_2 = 50.0, \tag{4.3}$$

and, in the case of ILC with the update law (3.20) with $Q = I$ and $R = 1$

$$\hat{K}_1 = [-21.8 \ -18.4 \ -7465.4], \quad \hat{K}_2 = 1.9. \tag{4.4}$$

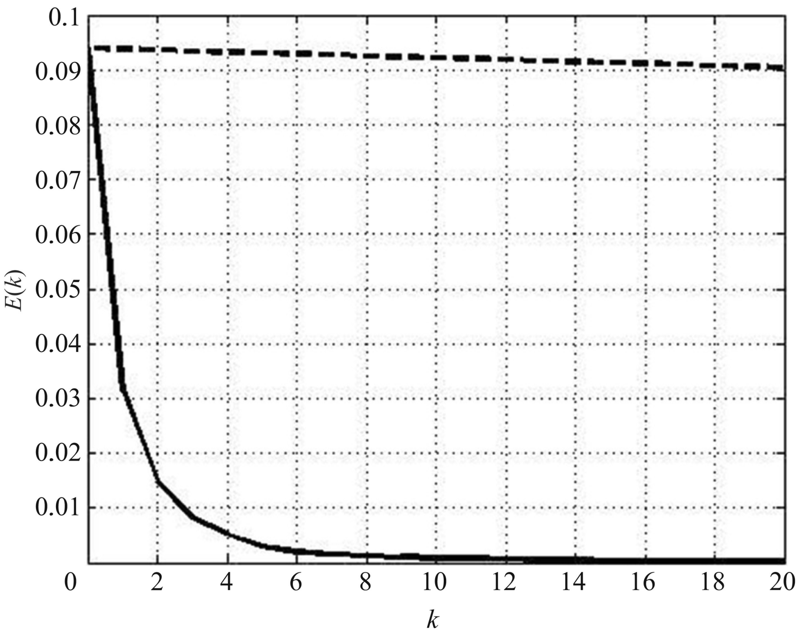

Fig. 4. Progression of $E(k)$ for the Y axis with $Q = I$ and $R = 1$ for (3.16) and $K_2 = -50$ (solid line) and (3.20) (dashed line).

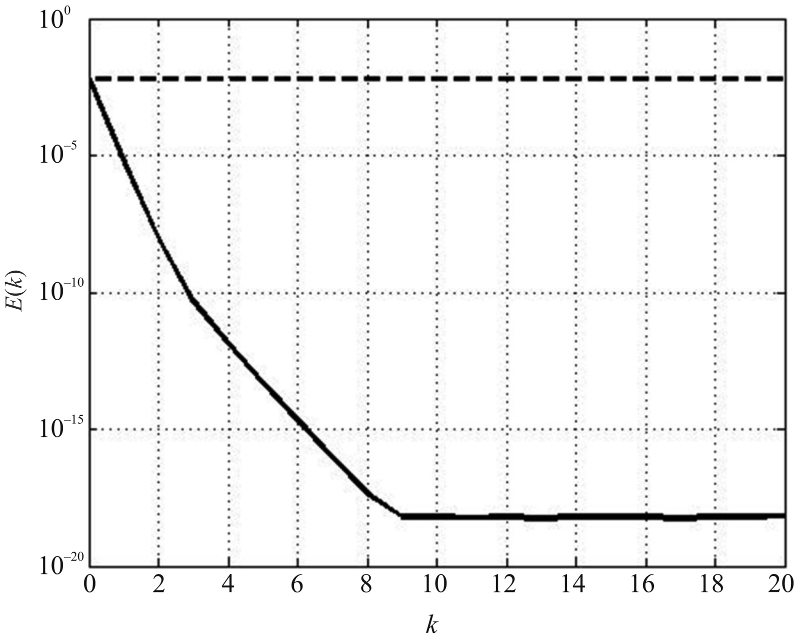

Fig. 5. Progression of $E(k)$ for the Z axis with $Q = I$ and $R = 1$ f (3.16) (solid line) and (3.20) (dashed line).

The linear matrix equations and inequalities have been solved as before but use is also made of the `YALMIP` parser. Analysis of (4.3) and (4.4) shows that, in the absence of structural constraints on $\hat{K}_1$ and $\hat{K}_2$, `SeDuMi`'s internal algorithm finds a solution in a domain characterized by a low convergence rate; at the same time, the constraints imposed by the heavy ball method direct the solution to a domain with accelerated convergence.

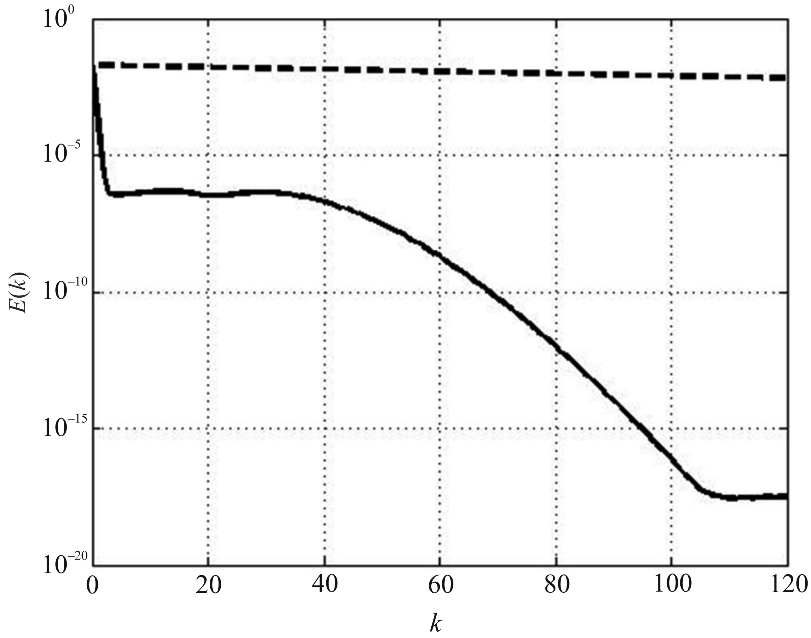

Fig. 6. Progression of $E(k)$ for the X-axis with Q and R given by (4.5) for (3.16) (solid line) and (3.20) (dashed line).

Now consider the ILC design along the Z and X axes. The transfer functions, for these axes are

$$G_Z(s) = \frac{15.8869(s + 661.2)}{s(s^2 + 707.6s + 3.377 \times 10^5)},$$

and

$$G_X(s) = \frac{13077183.4436(s + 113.4)(s^2 + 227.9s + 5.467 \times 10^4)}{s(s^2 + 61.57s + 1.125 \times 10^4)(s^2 + 227.9s + 5.467 \times 10^4)(s^2 + 466.1s + 6.142 \times 10^5)}.$$

To construct an observer along the Z axis, the same approach as in the previous case is used, with the observer poles taken as $\lambda_1 = 0.9,\ \lambda_{2,3} = -0.1 \pm 0.01i$. Applying the `place` function of MATLAB yields

$$G = [-0.405 \quad -28.01 \quad 0.078]^T.$$

Theorem 2 with unit $Q = I$ and $R = 1$ gives

$$K_1 = -6.33 \times 10^{-8},\ K_2 = -2.673 \times 10^3,\ K_3 = -8.33 \times 10^{-9}.$$

The simulation results are given in Fig. 5.

The observer poles along the X axis are selected as

$$\lambda_1 = 0.95, \quad \lambda_2 = -0.4, \quad \lambda_3 = 0.4, \quad \lambda_{4.5} = 0.4\left(\cos\frac{\pi}{3} \pm \sin\frac{\pi}{3}\right), \quad \lambda_{6,7} = 0.4\left(\cos\frac{\pi}{3} \pm \sin\frac{\pi}{3}\right).$$

Applying the `place` function of MATLAB yields

$$G = [38.1 \quad 40.161 \quad -44.093 \quad 15.913 \quad -16.428 \quad -12.67 \quad 0.309]^T.$$

In this case, Q and R are selected as

$$Q = \text{diag}[I \ 10I \ 1 \ 1 \ 1], \quad R = 1. \tag{4.5}$$

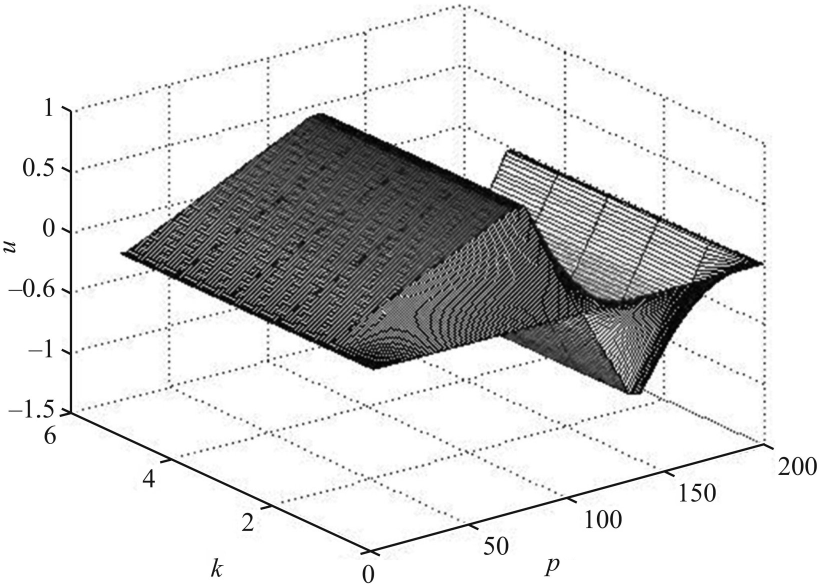

Fig. 7. Control progression along the X axis using (3.16).

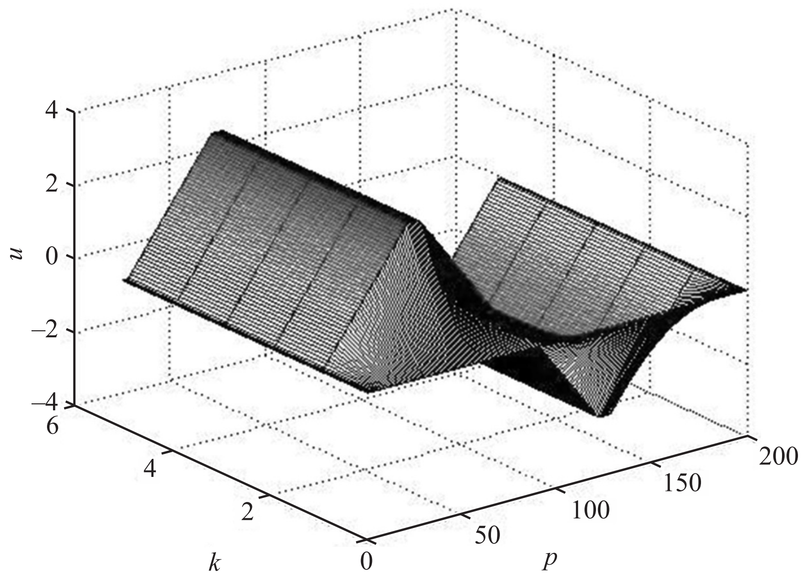

Fig. 8. Control progression along the Y axis with the update law (3.16).

Theorem 2 gives

$$K_1 = -5.715 \times 10^{-9}, \quad K_2 = -1308.2, \quad K_3 = -6.868 \times 10^{-10}.$$

The simulation results are given in Fig. 6. The control inputs for the X, Y, and Z axes for (3.16) are given in Figs. 7, 8, and 9.Note that the learned control on all axes is achieved rather quickly: after three or four trials, the controls on the axes do not change depending on k.

According to Fig. 5, the convergence of the learning error to numerical zero is achieved after several trials. A different picture is observed for the X axis (Fig. 6): $E(k)$ decreases by 10^7 times in four trials, then the convergence rate decreases, and numerical zero is achieved only after 107 trials; at the same time, this rate significantly exceeds the convergence rate of the alternative algorithm.

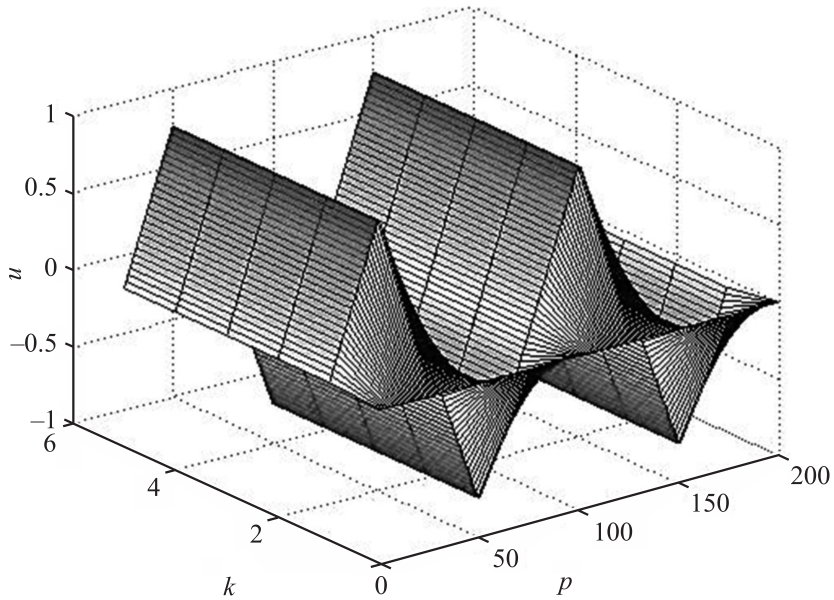

Fig. 9. Control progression along the Z axis with the update law (3.16).

Thus, in this example, the ILC design using the heavy ball method provides a high convergence rate that cannot be achieved by the alternative method. Note that for all three axes, K_1 and K_3 have quite small entries. Whether this is a general property of the new design, or is it related to the peculiarities of the dynamics of the system under consideration, remains unclear and requires additional research.

5. CONCLUSIONS

The ILC law (3.16) differs from those known in the literature in the special structure of the gain matrices. Judging by the simulation results, this structure is the main factor accelerating convergence. In the case of unstructured matrices, such as (3.20), it is probably possible to achieve high convergence rates, but this will demand a time-consuming heuristic procedure for selecting Q and R. Thus, it can be preliminarily concluded that using the heavy ball method gives an ILC law with a high convergence rate. The final conclusion requires a strict justification, which has not been obtained in this paper and will be the subject of further research. In the near future, the use of conventional gradient descent and Newton's method will be considered in ILC design; perhaps, the resulting ILC structures will provide the key to a general proof of accelerated convergence. In the longer term, it is planned to study the use of the conjugate gradient method and Nesterov's method.

Overall, there is much room for further research in this general area.

ACKNOWLEDGMENT

The authors dedicate this paper to the memory of Boris Polyak, the author of the heavy ball method and a number of other very significant results in optimization theory and control theory.

FUNDING

This work was partially supported by the Russian Science Foundation, project no. 23-71-01044, https://rscf.ru/en/project/23-71-01044/.

REFERENCES

1. Arimoto, S., Kawamura, S., and Miyazaki, F., Bettering Operation of Robots by Learning, *J. Robot. Syst.*, 1984, vol. 1, pp. 123–140.
2. Bristow, D.A., Tharayil, M., and Alleyne, A.G., A Survey of Iterative Learning Control: A Learning-Based Method for High-Performance Tracking Control *IEEE Control Syst. Magaz.*, 2006, vol. 26, no. 3, pp. 96–114.
3. Ahn, H-S., Chen, Y.Q., and Moore, K.L., Iterative Learning Control: Survey and Categorization, *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.*, 2007, vol. 37, no. 6, pp. 1099–1121.
4. Rogers, E., Chu, B., Freeman, C., and Lewin, P., *Iterative Learning Control Algorithms and Experimental Benchmarking*, Chichester: John Wiley & Sons, 2023.
5. Lim, I., Hoelzle, D.J., and Barton, K.L., A multi-objective iterative learning control approach for additive manufacturing applications, *Control Engineer. Practice*, 2017, vol. 64, pp. 74–87.
6. Sammons, P.M., Gegel, M.L., Bristow, D.A., and Landers, R.G., Repetitive Process Control of Additive Manufacturing with Application to Laser Metal Deposition, *IEEE Transact. Control Syst. Technol.*, 2019, vol. 27, no. 2, pp. 566–577.
7. Freeman, C.T., Rogers, E., Hughes, A.-M., Burridge, J.H., and Meadmore, K.L., Iterative learning control in health care: electrical stimulation and robotic-assisted upper-limb stroke rehabilitation, *IEEE Control Syst. Magaz.*, 2012, vol. 47, pp. 70–80.
8. Meadmore, K.L., Exell, T.A., Hallewell, E., Hughes, A.-M., Freeman, C.T., Kutlu, M., Benson, V., Rogers, E., and Burridge, J.H., The application of precisely controlled functional electrical stimulation to the shoulder, elbow and wrist for upper limb stroke rehabilitation: a feasibility study, *J. NeuroEngin. Rehabilitat.*, 2014, pp. 11–105.
9. Ketelhut, M., Stemmler, S., Gesenhues, J., Hein, M., and Abel, D., Iterative learning control of ventricular assist devices with variable cycle durations, *Control Engin. Practice*, 2019, vol. 83, pp. 33–44.
10. Wang, Y., Gao, F., and Doyle, F.J., Survey on iterative learning control, repetitive control, and run-to-run control, *J. Proc. Control*, 2009, vol. 19, pp. 1589–1600.
11. Polyak, B.T., Some methods of speeding up the convergence of iteration methods, *USSR Computational Mathematics and Mathematical Physics*, 1964, vol. 4, no. 5, pp. 1–17.
12. Polyak, B.T., *Introduction to Optimization, Optimization Software*, New York, 1987.
13. d'Aspremont, A., Scieur, D., and Taylor, A., Acceleration Methods. arXiv:2101.09545v3. 2021.
14. Ahn, H.S., Moore, K.L., and Chen, Y.Q., Iterative Learning Control. Robustness and Monotonic Convergence for Interval Systems, *Lecture Notes in Control and Information Sciences*, London: Springer-Verlag, 2007.
15. Bien, Z. and Huh, K.M., Higher-order iterative learning control algorithm, *IEE Proc. D-Control Theory Appl.*, 1989, vol. 136, pp. 105–112.
16. Chen, Y., Gong, Z., and Wen, C., Analysis of a high-order iterative learning control algorithm for uncertain nonlinear systems with state delays, *Automatica*, 1998, vol. 34, pp. 345–353.
17. Norrlöf, M. and Gunnarsson, S., A frequency domain analysis of a second order iterative learning control algorithm, *Proc. 38th IEEE Conf. Decis. Control*, 1999, vol. 2, pp. 1587–1592.
18. Bu, X., Yu, F., Fu, Z., and Wang, F., Stability analysis of high-order iterative learning control for a class of nonlinear switched systems, *Abstract Appl. Anal.*, 2013, pp. 1–13.
19. Wei, Y.-S. and Li, X.-D., Robust higher-order ILC for non-linear discrete-time systems with varying trail lengths and random initial state shifts, *IET Control Theory Appl.*, 2017, vol. 11, pp. 2440–2447.
20. Wang, X., Chu, B., and Rogers, E., Higher-order Iterative Learning Control Law Design using Linear Repetitive Process Theory: Convergence and Robustness, *IFAC PapersOnLine*, 2017, vol. 50-1, pp. 3123–3128.
21. Gu, P., Tian, S., and Chen, Y., Iterative learning control based on Nesterov accelerated gradient method, *IEEE Access*, 2019, vol. 7, pp. 115836–115842.
22. Nesterov, Y., A method of solving a convex programming problem with convergence rate $O(\frac{1}{k^2})$, *Sov. Math. Doklady*, 1983, vol. 269, no. 3, pp. 372–376.

23. Paszke, W., Rogers, E., and Galkowski, K., Experimentally verified generalized KYP lemma based iterative learning control design, *Control Engin. Practice*, 2016, vol. 33, pp. 57–67.

24. Rogers, E., Gałkowski, K., and Owens, D.H., Control Systems Theory and Applications for Linear Repetitive Processes, *Lecture Notes in Control and Information Sciences*, vol. 349, Berlin: Springer-Verlag, 2007.

25. Hladowski, L., Galkowski, K., Cai, Z., Rogers, E., Freeman, C.T., and Lewin, P.L., Experimentally supported 2D systems based iterative learning control law design for error convergence and performance, *Control Engin. Practice*, 2010, vol. 18, pp. 339–348.

26. Pakshin, P., Emelianova, J., Emelianov, M., Gałkowski, K., and Rogers, E., Dissipativity and stabilization of nonlinear repetitive processes, *Syst. Control Lett.*, 2016, vol. 91, pp. 14–20.

27. Pakshin, P. and Emelianova, J., Iterative learning control design for discrete-time stochastic switched systems, *Automat. Remote Control*, 2020, vol. 81, no. 11, pp. 2011–2025.

This paper was recommended for publication by P.S. Shcherbakov, a member of the Editorial Board