Repetitive Learning Control for Time-varying Robotic Systems: A Hybrid Learning Scheme

SUN Ming-Xuan[1] HE Xiong-Xiong[1] CHEN Bing-Yu[1]

Abstract Repetitive learning control is presented for finite-time-trajectory tracking of uncertain time-varying robotic systems. A hybrid learning scheme is given to cope with the constant and time-varying unknowns in system dynamics, where the time functions are learned in an iterative learning way, without the aid of Taylor expression, while the conventional differential learning method is suggested for estimating the constant ones. It is distinct that the presented repetitive learning control avoids the requirement for initial repositioning at the beginning of each cycle, and the time-varying unknowns are not necessary to be periodic. It is shown that with the adoption of hybrid learning, the boundedness of state variables of the closed-loop system is guaranteed and the tracking error is ensured to converge to zero as iteration increases. The effectiveness of the proposed scheme is demonstrated through numerical simulation.

Key words Adaptive control, iterative learning control, repetitive control, robotic systems, time-varying systems

1 Introduction

Iterative learning control (ILC) and repetitive control (RC) are two parallel research areas that have been developed for more than two decades. ILC copes with the repeated-tracking-control problem, where the same tasks are performed repetitively over a finite time interval. For execution, the system is first set to an initial position. Then, it starts, runs, stops, and resets to the same initial position for each cycle. Complete tracking, together with complete rejection of repetitive disturbances, is eventually achieved through repetition. RC aims to achieve tracking/rejection of periodic references/disturbances. According to the internal model principle, the dynamic structure of the reference/disturbance signals has to be incorporated in the controller. Initial repositioning is not required as the system undertaken starts where it has left. The interested reader may refer to the published references [1, 2] for the original ILC and RC formations, and the recent survey [3] for more references.

The restrictive assumption on initial repositioning would be destroyed as repositioning errors are inevitable in practical implementation. The presence of repositioning errors may eventually result in instability owing to ILC′s iterative nature, and the lack of robustness of the P-type learning was well understood. It is therefore necessary to ensure that learning algorithms are technically sound even in the presence of repositioning errors. The introduction of a forgetting factor was shown to be helpful to guarantee the robustness[1]. Attempts have been made to solve the initial shift problem, where the system does not reset to the desired initial reposition at each cycle. Instead, there exist initial shifts. Techniques include the application of the initial impulsive action[4], the initial rectifying action[5], and the average operator[6].

Early studies on RC were presented in [7, 8]. The stability of the controlled systems was established in the continuous-time and discrete-time domains, respectively. In [7], a modified repetitive control system was presented, which sacrifices tracking performance at high frequencies for system stability. In [8], the discrete-time formulation makes repetitive control theory more accessible. The zero phase-error compensation was included in the designed controller, which greatly simplified the stability analysis. A finite-dimensional (approximate) repetitive controller was presented in [9], which showed asymptotic tracking of the periodic reference signal by the proposed repetitive controller in closed-loop up to the Nth harmonic frequency. However, a periodic signal may have an infinite number of harmonics. An infinite-dimensional controller would be needed to have steady-state tracking of all those signals.

Over the years, a number of research efforts have been made for developing learning schemes without the requirement for initial repositioning, *e.g.* [10]. Cyclic learning control is introduced in [11], which cyclically steers the state of the system along a finite sequence of equilibrium points. Like RC, no initial repositioning at each cycle is required when using cyclic learning control.

RC is regarded as a simple ILC because both exploit the repetitive nature for control design. Looking into the difference between repetivity and periodicity, RC is unlike ILC. For example, in chemical industry, batch process spans over a finite interval, during which the reference temperature profile is not of periodicity. Repetitive learning control (RLC) blends RC and ILC, which was recently formulated as follows[12].

F1. Every operation ends in a finite time of duration, *i.e.*, $t \in [0,\, T]$;

F2. The desired trajectory is given a priori over $[0,\, T]$, and is closed, *i.e.*, $\boldsymbol{x}_d(T) = \boldsymbol{x}_d(0)$, where $\boldsymbol{x}_d(t), t \in [0,\, T]$, is the desired trajectory;

F3. The initial position of the system at the beginning of each cycle is aligned with that of the preceding cycle such that $\boldsymbol{x}_k(0) = \boldsymbol{x}_{k-1}(T)$, where k is the iteration index and $\boldsymbol{x}_k(t), t \in [0,\, T]$, is the iterative trajectory at the kth cycle;

F4. The time functions to be learned are iteration-independent;

F5. The system dynamics are invariant throughout all the cycles.

In RLC, it assumes that the system dynamics are invariant throughout all the cycles and the time functions to be learnt are k-independent, which is similar to ILC. However, unlike ILC, the tracking tasks are repetitively carried out in RLC without initial repositioning. This is similar to RC because the system undertaken by RLC starts where it has left at each cycle. In comparison with RC, RLC operates under more relaxed condition of repetivity, whereas the variables to be learned by RC should be of periodicity. An idea of repetitive learning was presented in [13] and numerical simulation was conducted to sustain and verify it. In [12], a RLC learning scheme was presented, and it has been shown to work well for robotic systems with constant unknown parameters and repetitive disturbances.

In this paper, we shall address the problem of repetitive learning control for trajectory tracking of time-varying

Received July 21, 2006; in revised form March 30, 2007
Supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, and National Natural Science Foundation of China (60474005)
1. College of Information Engineering, Zhejiang University of Technology, Hangzhou 310032, P. R. China
DOI: 10.1360/aas-007-1189

robotic systems. A hybrid learning scheme is presented, by which the iterative learning law is used to estimate the time-varying unknown parameters, and the differential learning law is suggested for estimating constant unknowns. Different from the work in [14], our proposed control scheme allows us to conduct the control design without expressing the unknown time-varying parameters to be a finite-length polynomial in time and a residue based on Taylor′s formula.

2 Robot model and problem statement

For an n-link rigid robot manipulator with time-varying parameters, additional terms will be present in the dynamic equation[14], which can be described by

$$D(\boldsymbol{q},\boldsymbol{\phi})\ddot{\boldsymbol{q}}+C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi})\dot{\boldsymbol{q}}+F(\boldsymbol{q},\dot{\boldsymbol{\phi}})\dot{\boldsymbol{q}}+G(\boldsymbol{q},\boldsymbol{\phi})=\boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{\phi}\in\mathbf{R}^{\boldsymbol{\phi}}$ is the vector of parameters, assumed to be continuously differentiable, $\boldsymbol{q}\in\mathbf{R}^n$ is the vector of generalized coordinates, $\boldsymbol{\tau}\in\mathbf{R}^n$ is the vector of input torques, $D(\boldsymbol{q},\boldsymbol{\phi})\in\mathbf{R}^{n\times n}$ is the inertia matrix, $C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi})$ is the Coriolis matrix, $F(\boldsymbol{q},\dot{\boldsymbol{\phi}})$ is the additional term due to the presence of time-varying parameters, and $G(\boldsymbol{q},\boldsymbol{\phi})\in\mathbf{R}^n$ is the gravity vector.

To facilitate control development, two properties of the dynamic model (1) are given:

P1. $D(\boldsymbol{q},\boldsymbol{\phi})$ is symmetric and positive definite for all $\boldsymbol{\phi}\in\mathbf{R}^{\boldsymbol{\phi}}$;

P2. The dynamics described by (1) is linearly parameterizable, *i.e.*,

$$\begin{aligned}&D(\boldsymbol{q},\boldsymbol{\phi})\ddot{\boldsymbol{q}}+C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi})\dot{\boldsymbol{q}}+F(\boldsymbol{q},\dot{\boldsymbol{\phi}})\dot{\boldsymbol{q}}+G(\boldsymbol{q},\boldsymbol{\phi})=\\&Y_0(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\theta}+Y_1(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{p}(t)\end{aligned} \tag{2}$$

where we introduce new parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{p}(t)$ as follows

$$[\boldsymbol{\theta}^{\mathrm{T}},\boldsymbol{p}^{\mathrm{T}}(t)]^{\mathrm{T}}=[\boldsymbol{\phi}^{\mathrm{T}},\dot{\boldsymbol{\phi}}^{\mathrm{T}}]^{\mathrm{T}} \tag{3}$$

$\boldsymbol{\theta}$ is the vector of unknown constant parameters, and $\boldsymbol{p}(t)$ is the vector of unknown time-varying parameters. $Y_0(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ and $Y_1(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ are the corresponding regression matrices.

Let $[0,\ T]$ be a bounded interval (*i.e.*, T is finite). The label $B[0,\ T]$ denotes the set consisting of all bounded functions on $[0,\ T]$. The robotic system under study is assumed to perform the same tasks over the interval $[0,\ T]$ repeatedly. Let $k(=0,\ 1,\cdots)$ denote the iteration index. Given a desired trajectory $\boldsymbol{q}_d(t)$, $t\in[0,\ T]$, satisfying that $\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d,\ddot{\boldsymbol{q}}_d\in B[0,\ T]$, the control objective of this paper is to find a torque profile $\boldsymbol{\tau}_k(t), t\in[0,\ T]$ that eventually realizes $\boldsymbol{q}_k(t)\to\boldsymbol{q}_d(t)$ and $\dot{\boldsymbol{q}}_k(t)\to\dot{\boldsymbol{q}}_d(t)$ on $[0,\ T]$ as $k\to\infty$.

The following two assumptions are made to achieve this objective.

Assumption 1. The given trajectory $\boldsymbol{q}_d(t)$ is twice differentiable and satisfies

$$\boldsymbol{q}_d(0)=\boldsymbol{q}_d(T),\dot{\boldsymbol{q}}_d(0)=\dot{\boldsymbol{q}}_d(T) \tag{4}$$

Assumption 2. For $k=0,\ 1,\cdots$, the initial values of $\boldsymbol{q}_k(0)$ and $\dot{\boldsymbol{q}}_k(0)$ are set to satisfy

$$\boldsymbol{q}_k(0)=\boldsymbol{q}_{k-1}(T),\dot{\boldsymbol{q}}_k(0)=\dot{\boldsymbol{q}}_{k-1}(T) \tag{5}$$

Define the tracking error at the kth cycle as $\boldsymbol{e}_k=\boldsymbol{q}_k-\boldsymbol{q}_d$. By Assumptions 1 and 2, $\boldsymbol{e}_k(0)=\boldsymbol{e}_{k-1}(T),\dot{\boldsymbol{e}}_k(0)=\dot{\boldsymbol{e}}_{k-1}(T)$.

3 RLC with hybrid learning

Throughout this paper, we use the notations $\hat{(\cdot)}_k$ to indicate the estimate for parameter $(\cdot)$ at the kth cycle, and $\tilde{(\cdot)}_k=\hat{(\cdot)}_k-(\cdot)$ to indicate the error between the estimate and the actual parameters.

In the following, we shall consider the form of torque input similar to those adopted for the control of rigid manipulators[15, 16]. Applying the torque input,

$$\begin{aligned}\boldsymbol{\tau}_k\ &=\ D(\boldsymbol{q}_k,\hat{\boldsymbol{\phi}}_k)a_k+C(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\hat{\boldsymbol{\phi}}_k)\dot{\boldsymbol{q}}_k+F(\boldsymbol{q}_k,\dot{\hat{\boldsymbol{\phi}}}_k)\dot{\boldsymbol{q}}_k+\\&\quad\ G(\boldsymbol{q}_k,\hat{\boldsymbol{\phi}}_k)+\boldsymbol{u}_k\end{aligned} \tag{6}$$

$$\boldsymbol{a}_k\ =\ \ddot{\boldsymbol{q}}_d-K_v\dot{\boldsymbol{e}}_k-K_p\boldsymbol{e}_k \tag{7}$$

Equation (1) can be written as

$$\begin{aligned}&D(\boldsymbol{q}_k,\hat{\boldsymbol{\phi}}_k)[\ddot{\boldsymbol{q}}_k-\boldsymbol{a}_k]=\\&[D(\boldsymbol{q}_k,\hat{\boldsymbol{\phi}}_k)-D(\boldsymbol{q}_k,\boldsymbol{\phi})]\ddot{\boldsymbol{q}}_k+\\&[C(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\hat{\boldsymbol{\phi}}_k)-C(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\boldsymbol{\phi})]\dot{\boldsymbol{q}}_k+\\&[F(\boldsymbol{q}_k,\dot{\hat{\boldsymbol{\phi}}}_k)-F(\boldsymbol{q}_k,\dot{\boldsymbol{\phi}})]\dot{\boldsymbol{q}}_k+\\&[G(\boldsymbol{q}_k,\hat{\boldsymbol{\phi}}_k)-G(\boldsymbol{q}_k,\boldsymbol{\phi})]+u_k=\\&Y_0(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\ddot{\boldsymbol{q}}_k)\tilde{\boldsymbol{\theta}}_k+Y_1(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\ddot{\boldsymbol{q}}_k)\tilde{\boldsymbol{p}}_k(t)+\boldsymbol{u}_k\end{aligned}$$

Define $\Phi_{0,k}=D^{-1}(\boldsymbol{q}_k,\hat{\boldsymbol{\phi}}_k)Y_0(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\ddot{\boldsymbol{q}}_k)$ and $\Phi_{1,k}=D^{-1}(\boldsymbol{q}_k,\hat{\boldsymbol{\phi}}_k)Y_1(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\ddot{\boldsymbol{q}}_k)$. It follows that

$$\begin{aligned}&\ddot{\boldsymbol{e}}_k+K_v\dot{\boldsymbol{e}}_k+K_p\boldsymbol{e}_k=\\&\Phi_{0,k}\tilde{\boldsymbol{\theta}}_k+\Phi_{1,k}\tilde{\boldsymbol{p}}_k(t)+D^{-1}(\boldsymbol{q}_k,\hat{\boldsymbol{\phi}}_k)\boldsymbol{u}_k\end{aligned} \tag{8}$$

Equation (8) can then be written in the state space form

$$\dot{\boldsymbol{x}}_k=A\boldsymbol{x}_k+B[\Phi_{0,k}\tilde{\boldsymbol{\theta}}_k+\Phi_{1,k}\tilde{\boldsymbol{p}}_k(t)+D^{-1}(\boldsymbol{q}_k,\hat{\boldsymbol{\phi}}_k)\boldsymbol{u}_k] \tag{9}$$

where $\boldsymbol{x}_k=[\boldsymbol{e}_k^{\mathrm{T}},\dot{\boldsymbol{e}}_k^{\mathrm{T}}]^{\mathrm{T}}$, and

$$A=\begin{bmatrix}0 & I\\ -K_p & -K_v\end{bmatrix},B=\begin{bmatrix}0\\ I\end{bmatrix}$$

By Assumptions 1 and 2, it is seen that $\boldsymbol{x}_k(0)=\boldsymbol{x}_{k-1}(T)$.

The feedback control law is given as

$$\boldsymbol{u}_k\ =\ -\frac{1}{2}\beta Y_{1,k}\Gamma_1^{-1}\Phi_{1,k}^{\mathrm{T}}B^{\mathrm{T}}P\boldsymbol{x}_k \tag{10}$$

with $\beta\geq 1$, and the update laws for $\hat{\boldsymbol{p}}_k$ and $\hat{\boldsymbol{\theta}}_k$ are given as

$$\hat{\boldsymbol{p}}_k(t)\ =\ \mathrm{sat}(\bar{\boldsymbol{p}}_k(t)) \tag{11}$$

$$\begin{aligned}\bar{\boldsymbol{p}}_{k+1}(t)\ &=\ \mathrm{sat}(\bar{\boldsymbol{p}}_k(t))-\Gamma_1^{-1}\Phi_{1,k}^{\mathrm{T}}(t)B^{\mathrm{T}}P\boldsymbol{x}_k(t),\\&\quad\ t\in[0,T]\\ \dot{\hat{\boldsymbol{\theta}}}_k\ &=\ -\Gamma_2^{-1}\Phi_{0,k}^{\mathrm{T}}B^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}_k,\ \hat{\boldsymbol{\theta}}_k(0)=\hat{\boldsymbol{\theta}}_{k-1}(T)\end{aligned} \tag{12}$$

where both Γ_1 and Γ_2 are diagonal positive definite matrices, P is the unique symmetric positive definite solution to the Lyapunov equation

$$A^{\mathrm{T}}P+PA=-Q \tag{13}$$

for a given $Q(=Q^{\mathrm{T}}>0)$. Initial conditions $\bar{\boldsymbol{p}}_0(t), t\in[0,\ T]$ and $\hat{\boldsymbol{\theta}}_0(0)$ can be simply set to zero if a priori information for both $\bar{\boldsymbol{p}}_0(t), t\in[0,\ T]$, and $\hat{\boldsymbol{\theta}}_0(0)$ are unavailable. To assure the boundedness of $\hat{\boldsymbol{p}}_k$, the saturation function sat : $\mathbf{R} \to \mathbf{R}$ is adopted, defined as that for a scalar a,

$$\mathrm{sat}(a) = \begin{cases} \bar{a}^1, & a < \bar{a}^1 \\ a, & \bar{a}^1 \leq a \leq \bar{a}^2 \\ \bar{a}^2, & a > \bar{a}^2 \end{cases} \tag{14}$$

where $\bar{a} = \{\bar{a}^1, \bar{a}^2\}$ represent the lower and upper bounds, satisfying that $\bar{a}^1 < \bar{a}^2$. For a vector $\boldsymbol{a} \in \mathbf{R}^m$, the vector-valued saturation function is defined as

$$\mathrm{sat}(\boldsymbol{a}) = [\mathrm{sat}(a_1), \mathrm{sat}(a_2), \cdots, \mathrm{sat}(a_m)]^{\mathrm{T}}$$

with $\bar{a} = \{\bar{a}_1, \bar{a}_2, \cdots, \bar{a}_m\}$ and $\bar{a}_i = \{\bar{a}_i^1, \bar{a}_i^2\}, i = 1, 2, \cdots, m$.

The learning control mechanism is a mixture of two different kinds of learning laws, 1) iterative learning and 2) differential learning laws. The iterative learning law (11) is used to estimate the time-varying parameter $\boldsymbol{p}(t)$. The differential learning law (12) is suggested for estimating $\boldsymbol{\theta}$ because it is constant.

We need Lemma 1 to aid the performance analysis of the proposed RLC scheme.

Lemma 1. For $\boldsymbol{a}, \boldsymbol{b} \in \mathbf{R}^m$, if a satisfies that $\bar{b}_i^1 \leq a_i \leq \bar{b}_i^2, i = 1, 2, \cdots, m$, then

$$[(\gamma+1)\boldsymbol{a} - (\gamma \boldsymbol{b} + \mathrm{sat}_{\bar{\boldsymbol{b}}}(\boldsymbol{b}))]^{\mathrm{T}} \Lambda [\boldsymbol{b} - \mathrm{sat}_{\bar{b}}(\boldsymbol{b})] \leq 0$$

where $\gamma \geq 0$ is a scalar and $\Lambda > 0$ is a diagonal matrix with appropriate dimension.

Proof. See the proof for Lemma 1 in [17]. $\square$

4 Performance analysis

In this section, we shall conduct performance analysis for the presented RLC design. Define the Lyapunov-like function as

$$\begin{aligned} L_{k+1}(t) &= \int_0^T \tilde{\boldsymbol{p}}_{k+1}^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_{k+1} \mathrm{d}s + V_k(t) \\ V_k &= \tilde{\boldsymbol{\theta}}_k^{\mathrm{T}} \Gamma_2 \tilde{\boldsymbol{\theta}}_k + \boldsymbol{x}_k^{\mathrm{T}} P \boldsymbol{x}_k \end{aligned}$$

Lemma 2. Given the desired trajectory $\boldsymbol{q}_d(t)$ for the robotic system (1), satisfying Assumptions 1 and 2, the torque input (6) together with feedback control law (10) and parameter update laws (11) and (12) results in

$$L_{k+1}(t) \leq -\int_0^T \boldsymbol{x}_k^{\mathrm{T}} Q \boldsymbol{x}_k \mathrm{d}s + \int_0^T \tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_k \mathrm{d}s + V_k(0) \tag{15}$$

and, as $t = T$

$$L_{k+1}(T) - L_k(T) \leq -\int_0^T \boldsymbol{x}_k^{\mathrm{T}} Q \boldsymbol{x}_k \mathrm{d}s \tag{16}$$

Proof. The difference between L_{k+1} and L_k can be written as

$$\begin{aligned} L_{k+1}(t) - L_k(t) = &\int_0^T [\tilde{\boldsymbol{p}}_{k+1}^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_{k+1} - \tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_k] \mathrm{d}s + \\ &V_k(t) - V_{k-1}(t) \end{aligned} \tag{17}$$

Through algebra manipulations, we have

$$\begin{aligned} &\tilde{\boldsymbol{p}}_{k+1}^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_{k+1} - \tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_k = \\ &[\hat{\boldsymbol{p}}_{k+1} - \boldsymbol{p}]^{\mathrm{T}} \Gamma_1 [\hat{\boldsymbol{p}}_{k+1} - \boldsymbol{p}] - \\ &[\bar{\boldsymbol{p}}_{k+1} - \boldsymbol{p}]^{\mathrm{T}} \Gamma_1 [\bar{\boldsymbol{p}}_{k+1} - \boldsymbol{p}] + [\bar{\boldsymbol{p}}_{k+1} - \boldsymbol{p}]^{\mathrm{T}} \Gamma_1 [\bar{\boldsymbol{p}}_{k+1} - \boldsymbol{p}] - \\ &[\hat{\boldsymbol{p}}_k - \boldsymbol{p}]^{\mathrm{T}} \Gamma_1 [\hat{\boldsymbol{p}}_k - \boldsymbol{p}] = \\ &[\bar{\boldsymbol{p}}_{k+1} - \hat{\boldsymbol{p}}_{k+1}]^{\mathrm{T}} \Gamma_1 [2\boldsymbol{p} - \hat{\boldsymbol{p}}_{k+1} - \bar{\boldsymbol{p}}_{k+1}] + \\ &[\bar{\boldsymbol{p}}_{k+1} - \boldsymbol{p}]^{\mathrm{T}} \Gamma_1 [\bar{\boldsymbol{p}}_{k+1} - \boldsymbol{p}] - \\ &[\hat{\boldsymbol{p}}_k - \boldsymbol{p}]^{\mathrm{T}} \Gamma_1 [\hat{\boldsymbol{p}}_k - \boldsymbol{p}] \end{aligned}$$

By Lemma 1, we have

$$[\bar{\boldsymbol{p}}_{k+1} - \hat{\boldsymbol{p}}_{k+1}]^{\mathrm{T}} \Gamma_1 [2\boldsymbol{p} - \hat{\boldsymbol{p}}_{k+1} - \bar{\boldsymbol{p}}_{k+1}] \leq 0$$

which implies

$$\begin{aligned} &\tilde{\boldsymbol{p}}_{k+1}^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_{k+1} - \tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_k = \\ &[\bar{\boldsymbol{p}}_{k+1} - \boldsymbol{p}]^{\mathrm{T}} \Gamma_1 [\bar{\boldsymbol{p}}_{k+1} - \boldsymbol{p}] - [\hat{\boldsymbol{p}}_k - \boldsymbol{p}]^{\mathrm{T}} \Gamma_1 [\hat{\boldsymbol{p}}_k - \boldsymbol{p}] \end{aligned}$$

and in turn results in

$$\begin{aligned} &\tilde{\boldsymbol{p}}_{k+1}^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_{k+1} - \tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Gamma_1 \tilde{\boldsymbol{p}}_k \leq \\ &-2\tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Gamma_1 [\hat{\boldsymbol{p}}_k - \bar{\boldsymbol{p}}_{k+1}] + [\hat{\boldsymbol{p}}_k - \bar{\boldsymbol{p}}_{k+1}]^{\mathrm{T}} \Gamma_1 [\hat{\boldsymbol{p}}_k - \bar{\boldsymbol{p}}_{k+1}] \end{aligned}$$

Accordingly, it follows from (17) that

$$\begin{aligned} L_k(t) - L_{k-1}(t) \leq &\int_0^T \{-2\tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Gamma_1 [\hat{\boldsymbol{p}}_k - \bar{\boldsymbol{p}}_{k+1}] + \\ &[\hat{\boldsymbol{p}}_k - \bar{\boldsymbol{p}}_{k+1}]^{\mathrm{T}} \Gamma_1 [\hat{\boldsymbol{p}}_k - \bar{\boldsymbol{p}}_{k+1}]\} \mathrm{d}s + \\ &V_k(t) - V_{k-1}(t) \end{aligned} \tag{18}$$

To get a useful expression for V_k, let us calculate the time derivative of V_k along (9), which can be given as

$$\begin{aligned} \dot{V}_k = &2\tilde{\boldsymbol{\theta}}_k^{\mathrm{T}} \Gamma_2 \dot{\hat{\boldsymbol{\theta}}}_k - \boldsymbol{x}^{\mathrm{T}} Q \boldsymbol{x}_k + 2\tilde{\boldsymbol{\theta}}_k^{\mathrm{T}} \Phi_{0,k}^{\mathrm{T}} B^{\mathrm{T}} P x_k + \\ &2\tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Phi_{1,k}^{\mathrm{T}} B^{\mathrm{T}} P \boldsymbol{x}_k + 2\boldsymbol{x}_k^{\mathrm{T}} PBD^{-1}(\boldsymbol{q}_k, \hat{\boldsymbol{\phi}}_k) \boldsymbol{u}_k \end{aligned}$$

Using update law (12) leads to

$$\begin{aligned} \dot{V}_k = &-\boldsymbol{x}_k^{\mathrm{T}} Q \boldsymbol{x}_k + 2\tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Phi_{1,k}^{\mathrm{T}} B^{\mathrm{T}} P \boldsymbol{x}_k + \\ &2\boldsymbol{x}_k^{\mathrm{T}} PBD^{-1}(\boldsymbol{q}_k, \hat{\boldsymbol{\phi}}_k) \boldsymbol{u}_k \end{aligned}$$

Integrating from $t = 0$ to T gives

$$\begin{aligned} V_k(t) = &V_k(0) + \int_0^T [-\boldsymbol{x}_k^{\mathrm{T}} Q \boldsymbol{x}_k + 2\tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Phi_{1,k}^{\mathrm{T}} B^{\mathrm{T}} P \boldsymbol{x}_k + \\ &2\boldsymbol{x}_k^{\mathrm{T}} PBD^{-1}(\boldsymbol{q}_k, \hat{\boldsymbol{\phi}}_k) \boldsymbol{u}_k] \mathrm{d}s \end{aligned} \tag{19}$$

Substituting (19) into (18) produces

$$\begin{aligned} &L_{k+1}(t) - L_k(t) \leq \\ &-\int_0^T \boldsymbol{x}_k^{\mathrm{T}} Q \boldsymbol{x}_k \mathrm{d}s + \\ &2\int_0^T [-\tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Gamma_1 (\hat{\boldsymbol{p}}_k - \bar{\boldsymbol{p}}_{k+1}) + \tilde{\boldsymbol{p}}_k^{\mathrm{T}} \Phi_{1,k}^{\mathrm{T}} B^{\mathrm{T}} P \boldsymbol{x}_k] \mathrm{d}s + \\ &\int_0^T \{[\hat{\boldsymbol{p}}_k - \bar{\boldsymbol{p}}_{k+1}]^{\mathrm{T}} \Gamma_1 [\hat{\boldsymbol{p}}_k - \bar{\boldsymbol{p}}_{k+1}] + \\ &2\boldsymbol{x}_k^{\mathrm{T}} PBD^{-1}(\boldsymbol{q}_k, \hat{\boldsymbol{\phi}}_k) \boldsymbol{u}_k]\} \mathrm{d}s + \\ &V_k(0) - V_{k-1}(t) \end{aligned} \tag{20}$$

Applying (10) and (11), we have

$$\begin{aligned}&L_{k+1}(t)-L_k(t)\leq\\&-\int_0^T[\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k+(\beta-1)\boldsymbol{x}_k^{\mathrm{T}}P^{\mathrm{T}}B\Phi_{1,k}\\&\Gamma_1^{-1}\Phi_{1,k}^{\mathrm{T}}(t)B^{\mathrm{T}}P\boldsymbol{x}_k]\mathrm{d}s+V_k(0)-V_{k-1}(t)\end{aligned}$$

As $\beta\geq 1$,

$$L_{k+1}(t)-L_k(t)\leq-\int_0^T\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k\mathrm{d}s+V_k(0)-V_{k-1}(t)\quad(21)$$

which implies

$$L_{k+1}(t)\leq-\int_0^T\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k\mathrm{d}s+L_k(t)-V_{k-1}(t)+V_k(0)\quad(22)$$

By the definition of $L_k(t)$,

$$L_k(t)-V_{k-1}(t)=\int_0^T\tilde{\boldsymbol{p}}_k^{\mathrm{T}}\Gamma_1\tilde{\boldsymbol{p}}_k\mathrm{d}s\quad(23)$$

Substituting (23) into (22) gives rise to inequality (15).

Setting $t=T$ in (21) yields

$$L_{k+1}(T)-L_k(T)\leq-\int_0^T\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k\mathrm{d}s+V_k(0)-V_{k-1}(T)\quad(24)$$

From the initial condition $\hat{\boldsymbol{\theta}}_k(0)=\hat{\boldsymbol{\theta}}_{k-1}(T)$ and by Assumptions 1 and 2, we have

$$V_k(0)=V_{k-1}(T)$$

Thus, inequality (16) holds. □

Lemma 3. $L_1(T)$ is bounded.

Proof. The time derivative of L_1 can be written as

$$\begin{aligned}\dot{L}_1&=\tilde{\boldsymbol{p}}_1^{\mathrm{T}}\Gamma_1\tilde{\boldsymbol{p}}_1+\dot{V}_0=-2\tilde{\boldsymbol{p}}_0^{\mathrm{T}}\Phi_{1,0}^{\mathrm{T}}B^{\mathrm{T}}P\boldsymbol{x}_0+\\&\boldsymbol{x}_0^{\mathrm{T}}PB\Phi_{1,0}\Gamma_1^{-1}\Phi_{1,0}^{\mathrm{T}}B^{\mathrm{T}}P\boldsymbol{x}_0+\\&\tilde{\boldsymbol{p}}_0^{\mathrm{T}}\Gamma_1\tilde{\boldsymbol{p}}_0+\dot{V}_0\end{aligned}\quad(25)$$

Note that

$$\begin{aligned}\dot{V}_0&=-\boldsymbol{x}_0^{\mathrm{T}}Q\boldsymbol{x}_0+2\tilde{\boldsymbol{p}}_0^{\mathrm{T}}\Phi_{1,0}^{\mathrm{T}}B^{\mathrm{T}}P\boldsymbol{x}_0+\\&2\boldsymbol{x}_0^{\mathrm{T}}PBD^{-1}(\boldsymbol{q}_0,\hat{\boldsymbol{\phi}}_0)\boldsymbol{u}_0\end{aligned}\quad(26)$$

Substituting (26) into (25) and using (26) yield

$$\begin{aligned}\dot{L}_1&=-\boldsymbol{x}_0^{\mathrm{T}}Q\boldsymbol{x}_0+(1-\beta)\boldsymbol{x}_0^{\mathrm{T}}PB\Phi_{1,0}^{\mathrm{T}}\Gamma_1^{-1}\Phi_{1,0}^{\mathrm{T}}B^{\mathrm{T}}P\boldsymbol{x}_0+\\&\tilde{\boldsymbol{p}}_0^{\mathrm{T}}\Gamma_1\tilde{\boldsymbol{p}}_0\end{aligned}$$

As $\beta\geq 1$,

$$\dot{L}_1\leq\tilde{\boldsymbol{p}}_0^{\mathrm{T}}\Gamma_1\tilde{\boldsymbol{p}}_0$$

Integrating both sides from 0 to T results in

$$L_1(T)\leq L_1(0)+\int_0^T\tilde{\boldsymbol{p}}_0^{\mathrm{T}}\Gamma_1\tilde{\boldsymbol{p}}_0\mathrm{d}s\quad(27)$$

Since $L_1(0)=V_0(0)=\tilde{\boldsymbol{\theta}}_0^{\mathrm{T}}(0)\Gamma_2\tilde{\boldsymbol{\theta}}_0(0)+\boldsymbol{x}_0^{\mathrm{T}}(0)P\boldsymbol{x}_0(0)$, and $\boldsymbol{p},\hat{\boldsymbol{p}}_0,\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_0(0),\boldsymbol{q}_0(0)$, and $\dot{\boldsymbol{q}}_0(0)$ are all bounded, $L_1(T)$ is thus bounded. □

We are now at a position to present the major result of our paper.

Theorem 1. Given the desired trajectory $\boldsymbol{q}_d(t)$ for the robotic system (1), satisfying Assumptions 1 and 2, the repetitive learning controller composed of (6), (10), (11), and (12) achieves the complete tracking in the sense that

1) $\boldsymbol{e}_k,\dot{\boldsymbol{e}}_k,\hat{\boldsymbol{\theta}}_k,\hat{\boldsymbol{p}}_k\in B[0,\ T]$;

2) Both $\boldsymbol{e}_k$ and $\dot{\boldsymbol{e}}_k$ converge to zero in the sense of $L^2[0,T]$, *i.e.*,

$$\begin{aligned}\lim_{k\to\infty}\int_0^T\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{e}_k\mathrm{d}s&=0\\\lim_{k\to\infty}\int_0^T\dot{\boldsymbol{e}}_k^{\mathrm{T}}\dot{\boldsymbol{e}}_k\mathrm{d}s&=0\end{aligned}$$

Proof. 1) It follows from Lemma 2 that

$$L_{k+1}(T)\leq L_k(T)\cdots\leq L_1(T)\quad(28)$$

By Lemma 3, $L_1(T)$ is bounded. We can conclude that $L_k(T)$ is bounded for all k. The boundedness of $L_k(T)$ implies the boundedness of $V_k(T)$ and $\int_0^T\tilde{\boldsymbol{p}}_k^{\mathrm{T}}\Gamma_1^{-1}\tilde{\boldsymbol{p}}_k\mathrm{d}s$. From (16), we have, for all $t\in[0,\ T]$,

$$\begin{aligned}\int_0^T\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k\mathrm{d}s\quad&\leq\quad\int_0^T\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k\mathrm{d}s\leq\\&L_k(T)-L_{k+1}(T)\leq L_k(T)\end{aligned}$$

which implies $\int_0^T\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k\mathrm{d}\boldsymbol{\tau}\in B[0,\ T]$. Also by Lemma 2, we have

$$L_{k+1}(t)\leq-\int_0^T\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k\mathrm{d}s+\int_0^T\tilde{\boldsymbol{p}}_k^{\mathrm{T}}\Gamma_1^{-1}\tilde{\boldsymbol{p}}_k\mathrm{d}s+V_{k-1}(T)$$

Therefore, $L_k(t)\in B[0,\ T]$. From the definition of $L_k(t)$, $V_k(t)\in B[0,\ T]$, which implies that $\hat{\boldsymbol{\theta}}_k(t)\in B[0,\ T]$ and $\boldsymbol{x}_k(t)\in B[0,\ T]$. It follows from (11) that $\hat{\boldsymbol{p}}_k(t)\in B[0,\ T]$. From the definition of $\boldsymbol{x}_k$ and the boundedness of $\boldsymbol{q}_d$ and $\dot{\boldsymbol{q}}_d$, we can conclude that both $\boldsymbol{q}_k(t)\in B[0,\ T]$ and $\dot{\boldsymbol{q}}_k(t)\in B[0,\ T]$.

2) From (28), $L_k(T)$ is monotone decreasing. The limit of $L_k(T)$ exists due to its positive definiteness. Therefore,

$$\lim_{k\to\infty}\{L_k(T)-L_{k+1}(T)\}=0$$

By Lemma 2,

$$\int_0^T\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k\mathrm{d}s\leq L_k(T)-L_{k+1}(T)$$

which implies

$$\lim_{k\to\infty}\int_0^T\boldsymbol{x}_k^{\mathrm{T}}Q\boldsymbol{x}_k\mathrm{d}s=0$$

Thus, $\boldsymbol{x}_k$ converges to zero in the sense of $L^2[0,\ T]$. It follows from the definition of $\boldsymbol{x}_k$ that both $\boldsymbol{e}_k$ and $\dot{\boldsymbol{e}}_k$ converge to zero in the sense of $L^2[0,\ T]$. □

The scheme presented in this paper is motivated by the adaptive control law for rigid manipulators derived by Craig[15]. Similarly, the scheme assumes that the joint acceleration is measurable, and the presence of ideal acceleration sensors is assumed in the practical implementation. The scheme should be applied with care because the joint acceleration may lead to an overlarge input torque. One condition is that an upper bound on the desired joint acceleration $\ddot{\boldsymbol{q}}_d$ is set in the implementation. The numerical result from the simulation given in the next section verifies that this condition is feasible. The introduction of a filter in the designed controller is one method to overcome the need for acceleration measurements. This deserves further investigation and will be part of our future research. For simplicity of presentation, in this paper, we focus on the controller design and performance analysis of the acceleration-required scheme.

The parameter update laws (11) and (12) do not ensure that the estimates remain within pre-specified regions, which may lead to that the inverse of the estimated inertia matrix is not bounded. To guarantee that the estimated parameters remain within known regions, Craig suggested to reset the estimates[15]. If parameter estimates move outside their bounds of the known regions, they are reset to their bounds. There are effective ways to overcome this restriction. One can make a projection-modification for update law (12), and apply a fully saturated learning law for modifying (11)[17]. It is interesting to note that Spong and Ortega modified the control law to remove the need to modify the parameter update law[16].

5 Case study

To verify the effectiveness of the designed learning controller for robotic systems with time-varying parameters, the model of the two-link robot manipulator given in [14] is used, in which the length of the links is assumed to be fixed and the payload is time-varying. The model of the manipulator, in the form of (1), is given by

$$D(\boldsymbol{q},\boldsymbol{\phi})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi})\dot{\boldsymbol{q}} + F(\boldsymbol{q},\dot{\boldsymbol{\phi}})\dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{29}$$

where $D(\boldsymbol{q},\boldsymbol{\phi}) = [d_{ij}]$, with

$$\begin{aligned}
d_{11} &= p_1 + 2p_2\cos q_2 + (v_1(q_2) + R^2/2)m_p(t)\\
d_{12} &= p_3 + p_2\cos q_2 + (v_2(q_2) + R^2/2)m_p(t)\\
d_{21} &= d_{1,2}\\
d_{22} &= p_3 + (v_3 + R^2/2)m_p(t)\\
C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi}) &= \begin{bmatrix} -v_4(q_2)\dot{q}_2 & -v_4(q_2)(\dot{q}_1+\dot{q}_2)\\ v_4(q_2)\dot{q}_1 & 0 \end{bmatrix}\\
F(\boldsymbol{q},\dot{\boldsymbol{\phi}}) &= \begin{bmatrix} v_1(q_2)+R^2/2 & v_2(q_2)+R^2/2\\ v_2(q_2)+R^2/2 & v_3+R^2/2 \end{bmatrix}\dot{m}_p
\end{aligned}$$

R is the radius of the cylindrical vessel, which is mounted on the end of the second link, $m_p(t) = k_1 t$ is the payload mass and k_1 is the constant water flow rate, p_1, p_2, and p_3 are the constant parameters that contain the masses and the inertia parameters of the links and the motors, $v_4(q_2) = (p_2 + l_1 l_2 m_p(t))\sin q_2$, $v_1(q_2) = l_1^2 + l_2^2 + 2l_1 l_2\cos q_2$, $v_2(q_2) = l_2^2 + l_1 l_2 \cos q_2$ and $v_3(q_2) = l_2^2$. For the manipulator dynamics, $\boldsymbol{p} = [p_1,\, p_2,\, p_3]^{\mathrm{T}}$ and $\boldsymbol{\theta}(t) = [m_p(t),\, \dot{m}_p(t)]^{\mathrm{T}}$, and the regression matrices Y_p and $Y_{\boldsymbol{\theta}}$ are given by

$$\begin{aligned}
Y_{p,11} &= \ddot{q}_1\\
Y_{p,12} &= 2\ddot{q}_1\cos q_2 + \ddot{q}_2\cos q_2 - \dot{q}_2\dot{q}_1\sin q_2 -\\
&\quad (\dot{q}_1+\dot{q}_2)\dot{q}_2\sin q_2\\
Y_{p,13} &= \ddot{q}_2\\
Y_{p,21} &= 0\\
Y_{p,22} &= \ddot{q}_1\cos q_2 + \dot{q}_1^2\sin q_2\\
Y_{p,23} &= \ddot{q}_1 + \ddot{q}_2\\
Y_{\boldsymbol{\theta},11} &= (v_1(q_2)+R^2/2)\ddot{q}_1 + (v_2(q_2)+R^2/2)\ddot{q}_2 -\\
&\quad l_1 l_2\dot{q}_2\dot{q}_1\sin q_2 - l_1 l_2(\dot{q}_1+\dot{q}_2)\dot{q}_2\sin q_2\\
Y_{\boldsymbol{\theta},12} &= (v_1(q_2)+R^2/2)\dot{q}_1 + (v_2(q_2)+R^2/2)\dot{q}_2\\
Y_{\boldsymbol{\theta},21} &= (v_2(q_2)+R^2/2)\ddot{q}_1 + (v_3+R^2/2)\ddot{q}_2 +\\
&\quad l_1 l_2\dot{q}_1^2\sin q_2\\
Y_{\boldsymbol{\theta},22} &= (v_2(q_2)+R^2/2)\dot{q}_1 + (v_3+R^2/2)\dot{q}_2
\end{aligned}$$

The general expression for the desired position trajectories, with continuous bounded position, velocity, and acceleration, is given by

$$\boldsymbol{q}_d(t) = \begin{cases} c_d(t, 0, t_1, \boldsymbol{q}_d^0, \boldsymbol{q}_d^1), & 0 \le t < t_1\\ \boldsymbol{q}_d^1, & t_1 \le t \le t_2\\ c_d(t, t_2, T, \boldsymbol{q}_d^1, \boldsymbol{q}_d^0), & t_2 < t \le T \end{cases} \tag{30}$$

with

$$c_d(t, t_0, t_f, \boldsymbol{q}^0, \boldsymbol{q}^f) = \boldsymbol{q}^0 + \left(10\frac{(t-t_0)^3}{(t_f-t_0)^3} - 15\frac{(t-t_0)^4}{(t_f-t_0)^4} + 6\frac{(t-t_0)^5}{(t_f-t_0)^5}\right)(\boldsymbol{q}^f - \boldsymbol{q}^0) \tag{31}$$

In (30), $\boldsymbol{q}_d^0$ and $\boldsymbol{q}_d^1$ represent the desired initial and intermediate positions, respectively, and the desired final position coincides with the desired initial position. It is easy to check that the desired position trajectory (30) and its derivative satisfy Assumption 1. Over the operation cycle $[0,\, T]$, the manipulator moves from the desired initial position to the desired intermediate position over the time interval $[0,\, t_1)$. Upon reaching the desired intermediate position at t_1, the manipulator will stay at this position for the time interval $[t_1,\, t_2]$. Immediately after t_2, the manipulator starts to move towards the desired final position, which is coincidentally the desired initial position, reaching it at T.

The data that we use in the simulation are

$$\begin{aligned}
&T = 5\,\mathrm{s},\ t_1 = 1\,\mathrm{s},\ t_2 = 4\,\mathrm{s}\\
&\boldsymbol{q}_d^0 = [0,\, 0]^{\mathrm{T}}\,\mathrm{rad},\ \boldsymbol{q}_d^1 = [1,\, 1]^{\mathrm{T}}\,\mathrm{rad}\\
&\boldsymbol{p} = [4.8,\, 0.9,\, 0.7]^{\mathrm{T}},\ l_1 = 1\,\mathrm{m},\ l_2 = 0.6\,\mathrm{m},\ R = 0.1\,\mathrm{m}\\
&k_1 = 0.3\,\mathrm{kg/s}
\end{aligned}$$

The initial value of parameter vector $\boldsymbol{p}$ differs from the true values by 200%. The performance indices are defined as $J_{p,k} = \sup_{t\in[0,5]}\max\{|q_{d,1}(t) - q_{k,1}(t)|, |q_{d,2}(t) - q_{k,2}(t)|\}$ and $J_{v,k} = \sup_{t\in[0,5]}\max\{|\dot{q}_{d,1}(t) - \dot{q}_{k,1}(t)|, |\dot{q}_{d,2}(t) - \dot{q}_{k,2}(t)|\}$. The torque input (6), together with parameter update laws (11) and (12), are used with the parameters settings:

$$\begin{aligned}
&\beta = 10,\ Q = \mathrm{diag}[30,\, 60]\\
&\Gamma_1 = \mathrm{diag}[1],\ \Gamma_2 = \mathrm{diag}[5]\\
&K_p = \mathrm{diag}[40],\ K_v = \mathrm{diag}[10]\\
&\boldsymbol{q}_0(0) = [0.2,\, 0.2]^{\mathrm{T}}\,\mathrm{rad},\ \dot{\boldsymbol{q}}_0(0) = [0,\, 0]^{\mathrm{T}}\,\mathrm{rad}
\end{aligned}$$

Simulation results are shown in Figs $1\sim 6$. It is observed that the joint position and velocity trajectories converge to the desired ones over the entire finite interval $[0,\, 5]$ through 25 cycles. Figure 5 shows the resulted torque input profiles at the cycle $k = 25$.

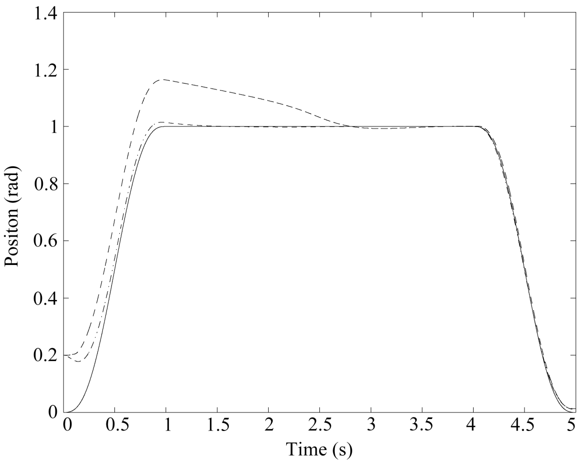

Fig. 1 Desired and actual position trajectories at cycle $k = 0$ (Desired: solid line for joints 1 and 2; actual: dash-dot line for joint 1 and dash line for joint 2)

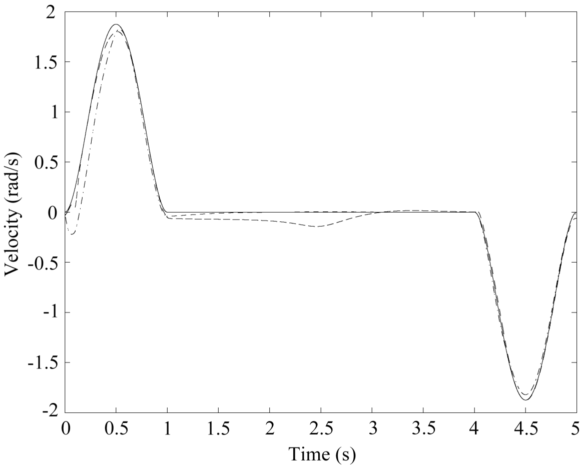

Fig. 2 Desired and actual velocity trajectories at cycle $k = 0$ (Desired: solid line for joints 1 and 2; actual: dash-dot line for joint 1 and dash line for joint 2)

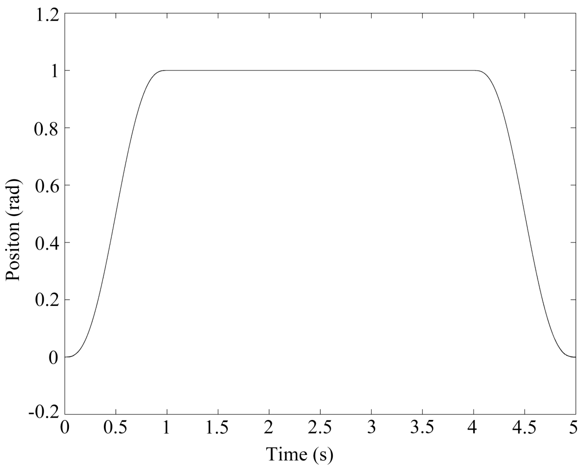

Fig. 3 Desired and actual position trajectories at cycle $k = 25$ (Desired: solid line for joints 1 and 2; actual: dash-dot line for joint 1 and dash line for joint2)

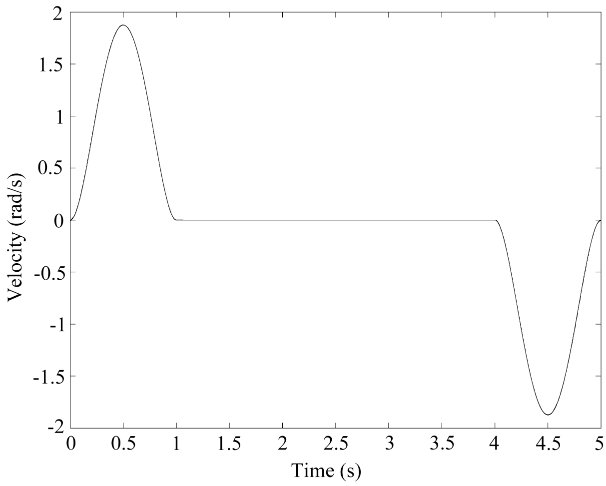

Fig. 4 Desired and actual velocity trajectories at cycle $k = 25$ (Desired: solid line for joints 1 and 2; actual: dash-dot line for joint 1 and dash line for joint2)

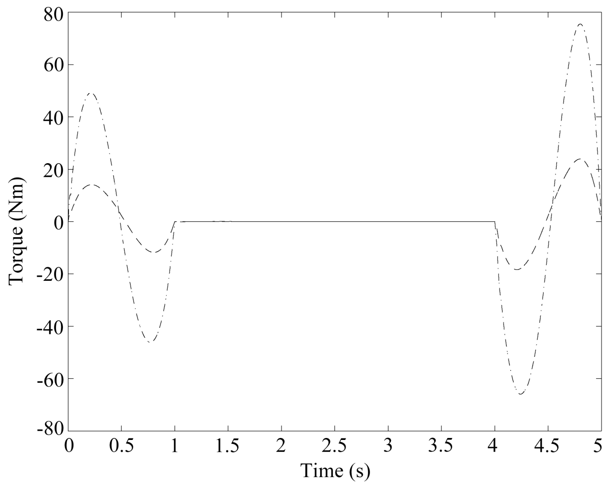

Fig. 5 Torque input profiles for joints 1 and 2 at cycle $k = 25$ (Dash-dot line for joint 1 and dash line for joint2)

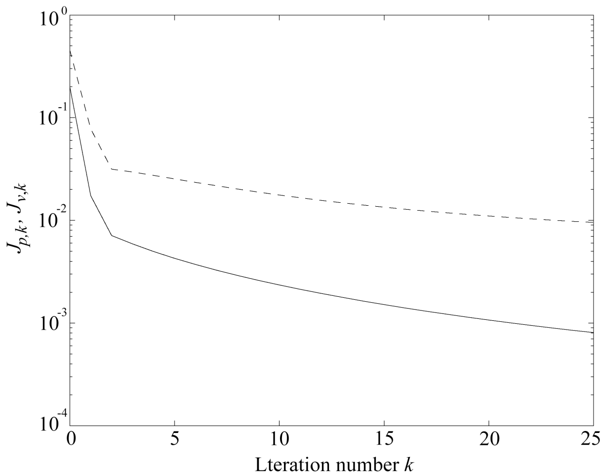

Fig. 6 Learning convergence: position error $J_{p,k}$ (Solid line) and velocity error $J_{v,k}$(Dash line)

6 Conclusion

In this paper, repetitive learning control has been presented for finite-time trajectory tracking of time-varying robotic systems. It has been shown that state variables in the closed-loop system are bounded and zero-error tracking is achieved over the finite time interval as the number of iterations increases, without the strict requirement of identical initial repositioning and without assuming the time-varying unknowns to be periodic. The hybrid learning has been shown applicable to robots with both constant and time-varying parameters, where the iterative learning law is used to estimate the time-varying unknowns whereas the differential learning law is suggested for estimating the constant ones.

References

1 Arimoto S. *Control Theory of Non-linear Mechanical Systems*. UK: Oxford University Press, 1996

2 Nakano M, Inoue T, Yamamoto Y, Hara S. *Repetitive Control*. Tokyo: SICE Publications, 1989

3 Bristow D A, Tharayil M, Alleyne A G. A survey of iterative learning control. *IEEE Control Systems Magzine*, 2006, **26**(3): 96∼114

4 Porter B, Mohamed S S. Iterative learning control of partially irregular multivariable plants with initial impulsive action. *International Journal of Systems Science*, 1991, **22**(3): 447∼454

5 Sun M, Wang D. Initial condition issues on iterative learning control for nonlinear systems time delay. *International Journal of Systems Science*, 2001, **32**(11): 1365∼1375

6 Park K H. An average-based PD-type iterative learning control for variable initial state error. *IEEE Transactions on Automatic Control*, 2005, **50**(6): 865∼869

7 Hara S, Yamamoto Y, Omata T, Nakano M. Repetitive control system: a new type servo system for periodic exogenous signals. *IEEE Transactions on Automatic Control*, 1988, **33**(7): 659∼668

8 Tomizuka M, Tsao T C, Chew K K. Analysis and synthesis of discrete-time repetitive controllers. *ASME Journal of Dynamic Systems, Measurement and Control*, 1989, **111**: 353∼358

9 Ghosh J, Paden B. Nonlonear repetitive control. *IEEE Transactions on Automatic Control*, 2000, **45**(5): 949∼954

10 Xu J X, Yan R. On repetitive learning control for periodic tracking tasks. *IEEE Transactions on Automatic Control*, 2006, **51**(11): 1842∼1848

11 Lucibello P, Panzieri S. Cyclic control of linear systems with application to a flexible arm. *IEE Proceedings, Part D, Control Theory Applications*, 1998, **145**(1): 19∼24

12 Sun M, Ge S S, Mareels I M Y. Adaptive repetitive learning control of robotic manipulators without the requirement for initial repositioning. *IEEE Transactions on Robotics*, 2006, **22**(3): 563∼568

13 Sun Ming-Xuan, Chen Yang-Quan, Huang Bao-Jian. Robust higher order iterative learning algorithm for tracking control of delayed repeated systems. *Acta Automatica Sinica*, 1994, **20**(3): 360∼365 (in Chinese)

14 Pagilla P R, Zhu Y. Adaptive control of mechanical systems with time-varying parameters and disturbances. *ASME Journal of Dynamic Systems, Measurement and Control*, 2004, **126**(3): 520∼530

15 Craig J J. *Adaptive Control of Machanical Manipulators*. New York: Addison-Wesley Publishing Company, 1988

16 Spong M W, Ortega R. On adaptive inverse dynamics control of rigid robots. *IEEE Transactions on Automatic Control*, 1990, **35**(1): 92∼95

17 Sun M, Ge S S. Adaptive repetitive control for a class of nonlinearly parametrized systems. *IEEE Transactions on Robotics*, 2006, **51**(10): 1684∼1688

SUN Ming-Xuan Professor in College of Information Engineering at Zhejiang University of Technology. His research interest covers iterative learning control and repetitive control, as well as their applications. Corresponding author of this paper. E-mail: mxsun@zjut.edu.cn

HE Xiong-Xiong Associate professor in College of Information Engineering at Zhejiang University of Technology. His research interest covers repetitive control and robotic control.

CHEN Bing-Yu Master student of Department of Automation at Zhejiang University of Technology. His research interest covers iterative learning control of nonlinear systems.