IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 16, 2021, accepted December 16, 2021, date of publication December 20, 2021,
date of current version December 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3136896

Adaptive Iterative Learning Control for Robot Manipulators With Time-Varying Parameters and Arbitrary Initial Errors

QIUZHEN YAN[1], **JIANPING CAI**[2], **YUNTAO ZHANG**[1], **AND ZHI YANG**[1]
[1]College of Information Engineering, Zhejiang University of Water Resources and Electric Power, Hangzhou, Zhejiang 310018, China
[2]College of Electronic Engineering, Zhejiang University of Water Resources and Electric Power, Hangzhou, Zhejiang 310018, China

Corresponding author: Qiuzhen Yan (zjhzyqz@gamail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673050, in part by the Key Research and Development Plan of Zhejiang Province under Grant 2021C03019, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY22F030011 and Grant LZ22F030008, and in part by the Zhejiang Province Welfare Technology Applied Research Project under Grant LGF20F020007 and Grant LGF21F030001.

ABSTRACT In this paper, a novel error-tracking adaptive iterative learning control scheme is proposed to solve trajectory-tracking problem for a class of robot manipulators with time-varying parameters and arbitrary initial errors. Firstly, desired error trajectories are constructed for implementing error tracking strategy in the robotic systems, so as to relax the requirement of zero initial errors, which is usually assumed to be met in traditional iterative learning control algorithms. Secondly, with the help of reasonable parameterization to the robotic dynamics, the adaptive iterative control law is designed by using Lyapunov approach. Projection-free combined time-domain and iteration-domain adaptive learning strategy is adopted to estimate the unknown time-invariant parametric uncertainties, and difference learning strategy is adopted to estimate unknown time-varying parametric uncertainties. As the iteration number increases, the system error follows its desired error trajectory over the whole interval. As a result, system state can perfectly track the reference signal in the predetermined part interval. In the end, several numerical simulations are presented to demonstrate the effectiveness of the designed control scheme.

INDEX TERMS Adaptive iterative learning control, robot manipulator, initial position problem, time-varying systems, error-tracking strategy.

I. INTRODUCTION

Iterative learning control (ILC) technique is effective in dealing with repetitive processes [1]–[4]. The ILC system updates the control input by using the information of error and input in the previous trial. The special working principle of ILC systems brings two advantages for applications. The first advantage is that ILC algorithms may be used in the cases where it is very difficult to carry out system modeling. The second one is that, in ILC systems, higher control precision may be obtained by using gradual iterations. Specifically, the system output/state may follow the desired signal over the whole operation time interval, i.e., the tracking error converges to zero at each time point during the operation time interval [5]–[11]. Adaptive ILC may be regarded as a combination of adaptive control and adaptive ILC [12], [13].

Up to now, many control technologies have been adopted to the controller design for robotic systems, such as adaptive control [14], adaptive sliding mode control [15], neural network control [16]–[18], fuzzy control [20], [21] and state-constrained control [19]. Note that robot manipulators are widely used to perform repetitive tasks in assembly lines, rehabilitation processes, and so on, where the reference trajectories are repetitive over a fixed period of operation time. In such cases, ILC is a suitable technology for designing controllers so as to improve the tracking performance progressively. We will consider two aspects of adaptive ILC algorithm designs for robotic applications in this work. The first aspect is about time-varying payload of robot manipulators during operation. As reported in literature, there have existed some control schemes that work well for robotic systems with unknown constant parameters [14]. Nevertheless, in many situations, the parameters in robotic systems are unknown and time-varying, such as the mass of payloads or the mass

The associate editor coordinating the review of this manuscript and approving it for publication was Nishant Unnikrishnan.

of links [22]. For example, robotic pouring and tilling operations belong to this situation. In [24], Song *et al.* proposed a switching type control law for the robot manipulators with time-varying parameters. In [22] and [23], Pagilla *et al.* applied the time-scaling technique to deal with time-varying parameters in robotic systems. In [25], Hsu and Fu proposed a globally adaptive decentralized control scheme to solve the trajectory tracking problem for time-varying robot manipulators. The investigations in [22]–[25] focus the asymptotic convergence of system errors in time domain. Due to the existence of inherent time-varying characteristics, the accurate system modeling for robotic systems with time-varying parameters is a difficult job, which does harm to the accuracy and applicability of many existing control schemes. In order to obtain better control performance for the tracking control of robot manipulators with time-varying parameters, it is a useful attempt to design the controller by using new control technology. To date, the literature on how to develop an adaptive ILC scheme for robot manipulators with time-varying parameters uncertainties is still few.

The second aspect we will address in this work is about initial position problem of robotic adaptive ILC systems. In traditional ILC algorithms, there exist a common assumption that the initial value of system errors at each iteration must be zero. Otherwise, a slight nonzero initial error may lead to the divergence of tracking error [26]. Due to the limitation of physical resetting, the zero initial error condition cannot be met in real industrial applications. As a result, traditional ILC algorithms cannot be used in industrial systems, which is called the initial position problem of ILC [26], [27]. The past two decades have witnessed the rapid progress in adaptive ILC algorithms of robotic systems. Some adaptive ILC schemes for the tracking of robotic systems with time-invariant parameters have been proposed in the related literature [28]–[31]. In [28], Jia and Yuan proposed three simple ILC schemes to solve the position tracking problem for rigid robot manipulators. In [29], Chien and Tayebi designed a combined time-domain and iteration-domain adaptation law for uncertain robot manipulators. In [30], Cao and Liu designed a adaptive boundary ILC law for a two−link rigid−flexible manipulator, so as to drive the joints to follow desired trajectory and eliminate deformation of flexible beam simultaneously. In [31], Li *et al.* developed an iterative learning impedance controller for rehabilitation robots driven by series elastic actuators. The topics discussed in [28]–[31] belong to the ILC system design for robot manipulators with time-invariant parametric uncertainties under zero initial error condition. For the reason mentioned above, this condition is hard to be satisfied in real robotic applications, which brings about the need for further research to remove it in robotic ILC designs. Up to now, only a few results have reported the ILC algorithms for the tracking of robotic systems with time-invariant parameter and with arbitrary initial errors [32]–[34]. Specifically, in [32], initial rectification action [35], [36] is applied to solve the non-repetitive trajectory tracking of robot manipulators with joint position constraints and actuator faults. In [33], a neural network-based adaptive ILC scheme is developed to solve the trajectory tracking problem for rigid robot manipulators with arbitrary initial errors, where the technique of time-varying boundary layer [37] is used to remove the zero initial error condition. Besides, in [34], alignment condition is adopted as a solution to the initial position problem of time-variant robotic systems with smoothly closed desired trajectory, based on which, an position-constrained adaptive ILC controller is designed by using barrier Lyapunov function [39], [40]. By contrast, the literature result involving the adaptive learning control (including adaptive ILC and adaptive repetitive learning control) for robotic systems with time-varying parameters is very few. The repetitive learning control algorithm for robotic systems with time-varying parameters proposed in [38] can be used in the situation that two following assumptions are satisfied: (1) The reference trajectory is smoothly closed, i.e., the initial state of reference trajectory must be equal to the final state of reference trajectory; (2) The initial state of current cycle must be equal to the final state of previous cycle. Note that the above two assumptions means that the initial system error of current cycle must be equal to the final system error of previous cycle.

To the best of authors' knowledge, no literature has reported the adaptive ILC method for robot manipulator with time-varying parameters and arbitrary initial errors. How to develop an error-tracking adaptive ILC scheme for the robotic systems in this situation is still an issue to be further studied.

Based on the above discussion, this paper investigates the trajectory tracking problem for a class of robot manipulators with time-varying parameters and arbitrary initial errors. An robotic error-tracking adaptive ILC scheme is developed to obtain high-precision tracking performance. Compared with existing results, the main contributions of this work can be summarized as follows:

1) Error-tracking adaptive ILC is proposed for robot manipulators with time-varying parameters, which can guarantee the performance and overcome the initial position problem of ILC.

2) A reasonable parameterization strategy is put forward for the later uncertainty compensation. A projection-free combined adaptive iterative learning law is designed to compensate time-invariant parametric uncertainties, which can guarantee the boundeness of parameter estimations without saturation/projection strategy used.

3) With the proposed ILC scheme, all the signals of the closed-loop robotic system are proved to be bounded, the closed-loop system error during the preset operation time interval converge to zero as the iteration number increases.

The remainder of this paper is organized as follows. Problem formulation is presented in Section 2. In Section 3, the desired error trajectory is constructed for dealing with the initial position problem of robotic adaptive ILC systems. In Section 4, the adaptive iterative learning controller is designed based on Lyapunov approach. In Section 5, the global convergence of the proposed adaptive ILC law is given. To demonstrate the effectiveness of the proposed error-tracking adaptive ILC scheme, several numerical simulations are shown in Section 6, followed by Section 7 which concludes the work.

II. PROBLEM FORMULATION

The dynamics of an n-link rigid robot manipulator with time-varying parameters is described as [22]

$$M(\boldsymbol{q},\boldsymbol{\phi}(t))\ddot{\boldsymbol{q}}+C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi}(t))\dot{\boldsymbol{q}}+F(\boldsymbol{q},\dot{\boldsymbol{\phi}}(t))\dot{\boldsymbol{q}}+\boldsymbol{G}(\boldsymbol{q},\boldsymbol{\phi}(t)) \\ =\boldsymbol{\tau}, \quad (1)$$

where $\boldsymbol{q}\in\mathbb{R}^n$, $\dot{\boldsymbol{q}}\in\mathbb{R}^n$ and $\ddot{\boldsymbol{q}}\in\mathbb{R}^n$ are the joint position, joint velocity and joint acceleration vectors, respectively, $M(\boldsymbol{q},\boldsymbol{\phi}(t))$ is the inertia matrix, $C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi}(t))\in\mathbb{R}^{n\times n}$ is the matrix composed of Coriolis and centrifugal terms, $\boldsymbol{G}(\boldsymbol{q},\boldsymbol{\phi}(t))\in\mathbb{R}^n$ is the gravity vector, $F(\boldsymbol{q},\dot{\boldsymbol{\phi}}(t))\in\mathbb{R}^n$ is a symmetric matrix, which is a consequence of the symmetry of the inertia matrix, and $\boldsymbol{\tau}\in\mathbb{R}^n$ is the torque input vector. Here, $\boldsymbol{\phi}(t)\in\mathbb{R}^m$ denotes the combination of time-invariant parameters and time-varying ones, where the former contain the masses and inertias of links and motors, and the latter contain the payload mass and inertia. The reference trajectory $\boldsymbol{q}_d\in\mathbb{R}^n$ is twice differentiable. The properties of the dynamic model (1) are given as follows:

Property 1: $M(\boldsymbol{q},\boldsymbol{\phi}(t))\in\mathbb{R}^{n\times n}$ is a symmetric positive definite inertia matrix.

Property 2: The matrix $\dot{M}(\boldsymbol{q},\boldsymbol{\phi}(t))-2C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi}(t))-F(\boldsymbol{q},\dot{\boldsymbol{\phi}}(t))$ is skew symmetric. Note that the skew-symmetry property of robot manipulators with time-varying uncertainties is different from that of robot manipulators with time-invariant parameters [22].

Property 3: The dynamics of robot manipulators with time-varying uncertainties can be linearly parameterizable [22], i.e.

$$\begin{aligned}M(\boldsymbol{q},\boldsymbol{\phi}(t))\ddot{\boldsymbol{\beta}}+C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi}(t))\dot{\boldsymbol{\beta}}+F(\boldsymbol{q},\dot{\boldsymbol{\phi}}(t))(\frac{1}{2}\dot{\boldsymbol{\beta}}+\frac{1}{2}\dot{\boldsymbol{q}}) \\ +\boldsymbol{G}(\boldsymbol{q},\boldsymbol{\phi}(t)) \\ =\Phi(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{\beta}},\ddot{\boldsymbol{\beta}})\boldsymbol{\vartheta}+W(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{\beta}},\ddot{\boldsymbol{\beta}})\boldsymbol{\eta}(t), \quad (2)\end{aligned}$$

where the time-variant parameter vector $\boldsymbol{\vartheta}$ contains the masses and inertias of links and motors, the time-varying parameter vector $\boldsymbol{\eta}(t)$ contains the payload mass and inertia, and $\Phi(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{\beta}},\ddot{\boldsymbol{\beta}})$ and $W(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{\beta}},\ddot{\boldsymbol{\beta}})$ are the regressor matrices. More details on the linear parameterization of robotic systems with time-varying uncertainties, see [22], [23].

The robot manipulator considered in this work performs a repetitive task over the interval $t\in[0,T]$, whose dynamics at the kth iteration can be written as

$$M(\boldsymbol{q}_k,\boldsymbol{\phi}(t))\ddot{\boldsymbol{q}}_k+C(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\boldsymbol{\phi}(t))\dot{\boldsymbol{q}}_k+F(\boldsymbol{q}_k,\dot{\boldsymbol{\phi}}(t))\dot{\boldsymbol{q}}_k \\ +\boldsymbol{G}(\boldsymbol{q}_k,\boldsymbol{\phi}(t))=\boldsymbol{\tau}_k. \quad (3)$$

The control objective is to find a sequence of appropriate torque inputs $\boldsymbol{\tau}_k$ to make $\boldsymbol{q}_k(t)$ accurately track $\boldsymbol{q}_d(t)$ under the condition that $\boldsymbol{q}_k(0)\neq\boldsymbol{q}_d(0)$ and $\dot{\boldsymbol{q}}_k(0)\neq\dot{\boldsymbol{q}}_d(0)$, as the iteration number increases.

For brevity, in what follows, arguments are sometimes omitted when no confusion is likely to arise, and M_k, C_k, F_k and $\boldsymbol{G}_k$ denote $M(\boldsymbol{q}_k,\boldsymbol{\phi})$, $C(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\boldsymbol{\phi})$, $F(\boldsymbol{q}_k,\dot{\boldsymbol{\phi}})$ and $G(\boldsymbol{q}_k,\boldsymbol{\phi})$, respectively.

Remark 1: In traditional robotic ILC algorithms, $\boldsymbol{q}_k(0)=\boldsymbol{q}_d(0)$ and $\dot{\boldsymbol{q}}_k(0)=\dot{\boldsymbol{q}}_d(0)$ are assumed to be met at each iteration [28], [41]. In the repetitive learning control algorithm for robotic systems with time-varying parameters [38], the assumption conditions $\boldsymbol{q}_k(0)=\boldsymbol{q}_{k-1}(T)$, $\dot{\boldsymbol{q}}_k(0)=\dot{\boldsymbol{q}}_{k-1}(T)$, $\boldsymbol{q}_d(0)=\boldsymbol{q}_d(T)$ and $\dot{\boldsymbol{q}}_d(0)=\dot{\boldsymbol{q}}_d(T)$ must be observed. In this work, we want to deduce an adaptive ILC scheme so as to remove all the above-mentioned assumptions.

III. CONSTRUCTION OF DESIRED ERROR TRAJECTORY

Define $\tilde{\boldsymbol{q}}_k=[\tilde{q}_{1,k},\tilde{q}_{2,k},\cdots,\tilde{q}_{n,k}]^T=\boldsymbol{q}_k-\boldsymbol{q}_d$ and $d\tilde{\boldsymbol{q}}_k=[d\tilde{q}_{1,k},d\tilde{q}_{2,k},\cdots,d\tilde{q}_{n,k}]^T=\dot{\boldsymbol{q}}_k-\dot{\boldsymbol{q}}_d$. Due to $\boldsymbol{q}_k(0)\neq\boldsymbol{q}_d(0)$, it is impossible to obtain $\boldsymbol{q}_k(t)=\boldsymbol{q}_d(t)$, $\forall t\in[0,T]$. As an alternative, we want to find a sequence of appropriate torque inputs $\boldsymbol{\tau}_k$ such that $\lim_{k\to+\infty}(\boldsymbol{q}_k(t)-\boldsymbol{q}_d(t))=0$, $\forall t\in[t_\epsilon,T]$, where t_ϵ is a preset moment between 0 and T.

To remove the zero initial error condition, a common assumption in traditional ILC algorithms, let us construct the desired error trajectory $\tilde{\boldsymbol{q}}^d_k=[\tilde{q}^d_{1,k},\tilde{q}^d_{2,k},\cdots,\tilde{q}^d_{n,k}]^T$ and $d\tilde{\boldsymbol{q}}^d_k=[d\tilde{q}^d_{1,k},d\tilde{q}^d_{2,k},\cdots,d\tilde{q}^d_{n,k}]^T=\dot{\tilde{q}}^d_k$ as follows:

For $i=1,2,\cdots,n$, while $t_\epsilon\le t\le T$, let

$$\tilde{q}^d_{i,k}(t)=0,\quad d\tilde{q}^d_{i,k}(t)=0; \quad (4)$$

while $0\le t<t_\epsilon$, let

$$\begin{aligned}\tilde{q}^d_{i,k}(t)&=a_{i0,k}+a_{i1,k}t+a_{i2,k}t^2+a_{i3,k}t^3+a_{i4,k}t^4 \\ &\quad+a_{i5,k}t^5, \\ d\tilde{q}^d_{i,k}(t)&=a_{i1,k}+2a_{i2,k}t+3a_{i3,k}t^2+4a_{i4,k}t^3 \\ &\quad+5a_{i5,k}t^4, \quad (5)\end{aligned}$$

where $a_{i0,k}=\tilde{q}_{i,k}(0)$, $a_{i1,k}=d\tilde{q}_{i,k}(0)$, $a_{i2,k}=0$, $[a_{i3,k},a_{i4,k},a_{i5,k}]^T=$

$$\begin{bmatrix}t_\epsilon^3 & t_\epsilon^4 & t_\epsilon^5\\ 3t_\epsilon^2 & 4t_\epsilon^3 & 5t_\epsilon^4\\ 6t_\epsilon & 12t_\epsilon^2 & 20t_\epsilon^3\end{bmatrix}^{-1}\begin{bmatrix}-a_{i0,k}-a_{i1,k}t_\epsilon-a_{i2,k}t_\epsilon^2\\ -a_{i1,k}-2a_{i2,k}t_\epsilon\\ -2a_{i2,k}\end{bmatrix}. \quad (6)$$

Define $\boldsymbol{z}_k=[z_{1,k},z_{2,k},\cdots,z_{n,k}]^T=\tilde{\boldsymbol{q}}_k-\tilde{\boldsymbol{q}}^d_k$. It can be easily seen from the above construction that $z_{i,k}(0)=0$, $\dot{z}_{i,k}(0)=0$ and $\ddot{z}_{i,k}(0)=0$ hold for $i=1,2,\cdots,n$. Note that $z_{i,k}(t)=0,\dot{z}_{i,k}(t)=0$, $\forall t\in[0,T]$ is a sufficient condition for $\tilde{q}_{i,k}(t)=0,d\tilde{q}_{i,k}(t)=0$, $\forall t\in[t_\epsilon,T]$. In order to make $\tilde{\boldsymbol{q}}_k$ and $d\tilde{\boldsymbol{q}}_k$ follow $\tilde{\boldsymbol{q}}^d_k$ and $d\tilde{\boldsymbol{q}}^d_k$ for $t\in[t_\epsilon,T]$, respectively, our strategy is to let $\boldsymbol{z}_k(t)=0$ and $\dot{\boldsymbol{z}}_k(t)=0$ for $t\in[0,T]$.

Remark 2: (4) means that once $\boldsymbol{z}_k(t)=0,\dot{\boldsymbol{z}}_k(t)=0$, $\forall t\in[T_\epsilon,T]$ holds, then $\boldsymbol{q}_k(t)=\boldsymbol{q}_d(t),\dot{\boldsymbol{q}}_k(t)=\dot{\boldsymbol{q}}_d(t)$, $\forall t\in[t_\epsilon,T]$ will hold subsequently. From (5) and (6), we can see that $\boldsymbol{z}_k(0)=0,\dot{\boldsymbol{z}}_k(0)=0$, which is helpful to solve the initial problem of ILC. Moreover, $\tilde{\boldsymbol{q}}^d_k(t),\dot{\tilde{\boldsymbol{q}}}^d_k(t)$ and $\ddot{\tilde{\boldsymbol{q}}}^d_k(t)$ are all continuous over the whole interval (in particular, including $t=t_\epsilon$), which is helpful to obtain the continuity of control input to be designed.

IV. THE DESIGN OF ADAPTIVE LEARNING CONTROLLERS

Define a filtered tracking error-like variable $\boldsymbol{s}_k \in \mathbb{R}^n$ as follows:

$$\boldsymbol{s}_k = \dot{\boldsymbol{z}}_k + \lambda \boldsymbol{z}_k = \dot{\boldsymbol{q}}_k - \dot{\boldsymbol{q}}_d - \dot{\tilde{\boldsymbol{q}}}_k^d + \lambda \boldsymbol{z}_k \tag{7}$$

where λ is a positive constant. Taking the derivative of $V_k = \frac{1}{2}\boldsymbol{s}_k^T M_k \boldsymbol{s}_k$ with respect to time t, we have

$$\begin{aligned}
\dot{V}_k &= \frac{1}{2}\boldsymbol{s}_k^T \dot{M}_k \boldsymbol{s}_k + \boldsymbol{s}_k^T M_k(\ddot{\boldsymbol{q}}_k - \ddot{\boldsymbol{q}}_d - \ddot{\tilde{\boldsymbol{q}}}_k^d + \lambda \dot{\boldsymbol{z}}_k) \\
&= \frac{1}{2}\boldsymbol{s}_k^T \dot{M}_k \boldsymbol{s}_k + \boldsymbol{s}_k^T(\boldsymbol{\tau}_k - C_k\dot{\boldsymbol{q}}_k - F_k\dot{\boldsymbol{q}}_k - \boldsymbol{G}_k) \\
&\quad + \boldsymbol{s}_k^T M_k(-\ddot{\boldsymbol{q}}_d - \ddot{\tilde{\boldsymbol{q}}}_k^d + \lambda \dot{\boldsymbol{z}}_k) \\
&= \frac{1}{2}\boldsymbol{s}_k^T \dot{M}_k \boldsymbol{s}_k - \boldsymbol{s}_k^T C_k \boldsymbol{s}_k - \frac{1}{2}\boldsymbol{s}_k^T F_k \boldsymbol{s}_k + \boldsymbol{s}_k^T C_k \boldsymbol{s}_k \\
&\quad + \frac{1}{2}\boldsymbol{s}_k^T F_k \boldsymbol{s}_k + \boldsymbol{s}_k^T(\boldsymbol{\tau}_k - C_k\dot{\boldsymbol{q}}_k - F_k\dot{\boldsymbol{q}}_k - \boldsymbol{G}_k) \\
&\quad + \boldsymbol{s}_k^T M_k(-\ddot{\boldsymbol{q}}_d - \ddot{\tilde{\boldsymbol{q}}}_k^d + \lambda \dot{\boldsymbol{z}}_k)
\end{aligned} \tag{8}$$

By Property 2, $\frac{1}{2}\boldsymbol{s}_k^T \dot{M}_k \boldsymbol{s}_k - \boldsymbol{s}_k^T C_k \boldsymbol{s}_k - \frac{1}{2}\boldsymbol{s}_k^T F_k \boldsymbol{s}_k = 0$ holds. Thus, (8) may be rewritten as

$$\begin{aligned}
\dot{V}_k = \boldsymbol{s}_k^T C_k \boldsymbol{s}_k + \frac{1}{2}\boldsymbol{s}_k^T F_k \boldsymbol{s}_k + \boldsymbol{s}_k^T(\boldsymbol{\tau}_k - C_k\dot{\boldsymbol{q}}_k - F_k\dot{\boldsymbol{q}}_k - \boldsymbol{G}_k) \\
+ \boldsymbol{s}_k^T M_k(-\ddot{\boldsymbol{q}}_d - \ddot{\tilde{\boldsymbol{q}}}_k^d + \lambda \dot{\boldsymbol{z}}_k).
\end{aligned} \tag{9}$$

By defining $\boldsymbol{q}_{zk} = [q_{z1,k}, q_{z2,k}, \cdots, q_{zn,k}]^T = q_d + \tilde{q}_k^d - \lambda \int_0^t \boldsymbol{z}_k(\upsilon)d\upsilon$ and performing some simple algebraic operations, we can further simplify (9) as follows:

$$\begin{aligned}
\dot{V}_k &= \boldsymbol{s}_k^T C_k \boldsymbol{s}_k + \frac{1}{2}\boldsymbol{s}_k^T F_k \boldsymbol{s}_k + \boldsymbol{s}_k^T(\tau_k - C_k\dot{\boldsymbol{q}}_k - F_k\dot{\boldsymbol{q}}_k - \boldsymbol{G}_k) \\
&\quad - \boldsymbol{s}_k^T[M_k\ddot{\boldsymbol{q}}_{zk} + C_k\dot{\boldsymbol{q}}_{zk} + \frac{1}{2}F_k(\dot{\boldsymbol{q}}_{zk} + \dot{q}_k) + \boldsymbol{G}_k] \\
&\quad + \boldsymbol{s}_k^T[C_k\dot{\boldsymbol{q}}_{zk} + \frac{1}{2}F_k(\dot{\boldsymbol{q}}_{zk} + \dot{q}_k) + \boldsymbol{G}_k] \\
&= \boldsymbol{s}_k^T\tau_k - \boldsymbol{s}_k^T C_k(\dot{\boldsymbol{q}}_k - \dot{\boldsymbol{q}}_{zk} - \boldsymbol{s}_k) - \frac{1}{2}\boldsymbol{s}_k^T F_k(\dot{\boldsymbol{q}}_k - \dot{\boldsymbol{q}}_{zk} \\
&\quad - \boldsymbol{s}_k) - \boldsymbol{s}_k^T[M_k\ddot{\boldsymbol{q}}_{zk} + C_k\dot{\boldsymbol{q}}_{zk} + F_k(\frac{1}{2}\dot{\boldsymbol{q}}_{zk} + \frac{1}{2}\dot{\boldsymbol{q}}_k) \\
&\quad + \boldsymbol{G}_k].
\end{aligned} \tag{10}$$

According to the definition of $\boldsymbol{s}_k$, (10) now implies

$$\dot{V}_k = \boldsymbol{s}_k^T\tau_k - \boldsymbol{s}_k^T[M_k\ddot{\boldsymbol{q}}_{zk} + C_k\dot{\boldsymbol{q}}_{zk} + F_k(\frac{1}{2}\dot{\boldsymbol{q}}_{zk} + \frac{1}{2}\dot{\boldsymbol{q}}_k) + \boldsymbol{G}_k] \tag{11}$$

By Property 3, it follows from (11) that

$$\begin{aligned}
\dot{V}_k = \boldsymbol{s}_k^T\tau_k - \boldsymbol{s}_k^T[\Phi(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \dot{\boldsymbol{q}}_{zk}, \ddot{\boldsymbol{q}}_{zk})\boldsymbol{\vartheta} \\
+ W(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \dot{\boldsymbol{q}}_{zk}, \ddot{\boldsymbol{q}}_{zk})\boldsymbol{\eta}(t)]
\end{aligned} \tag{12}$$

On the basis of (12), we design an adaptive ILC law as

$$\boldsymbol{\tau}_k = -\gamma_1\boldsymbol{s}_k + \Phi_k\boldsymbol{\vartheta}_k + W_k\boldsymbol{\eta}_k, \tag{13}$$

where $\gamma_1 > 0$, $\boldsymbol{\vartheta}_k$ is determined by a combined time-domain and iteration-domain learning law (14), and $\boldsymbol{\eta}_k$ is determined by a difference learning law (15).

$$\begin{aligned}
(1-\mu)\dot{\boldsymbol{\vartheta}}_k &= -\gamma_2\Phi_k^T\boldsymbol{s}_k + \mu(\boldsymbol{\vartheta}_{k-1} - \boldsymbol{\vartheta}_k), \\
\boldsymbol{\vartheta}_k(0) &= \boldsymbol{\vartheta}_{k-1}(T), \quad \boldsymbol{\vartheta}_{-1}(t) = 0,
\end{aligned} \tag{14}$$

where $\gamma_2 > 0$, $1 > \mu > 0$, and $\boldsymbol{\vartheta}_k$ is used to estimate $\boldsymbol{\vartheta}$.

$$\boldsymbol{\eta}_k = \mathrm{sat}_{\bar{\eta}}(\boldsymbol{\eta}_{k-1}) - \gamma_3 W_k^T\boldsymbol{s}_k, \quad \boldsymbol{\eta}_{-1} = 0, \tag{15}$$

where $\gamma_3 > 0$, $\boldsymbol{\eta}_k$ is used to estimate $\boldsymbol{\eta}(t)$. In (15) and what follows, the saturation function $\mathrm{sat}_{\bar{\cdot}}(\hat{\cdot})$ is defined as follows: For a scalar $\hat{b}$ used to estimate unknown variable b,

$$\mathrm{sat}_{\bar{b}}(\hat{b}) = \mathrm{sign}(\hat{b})\min(|\hat{b}|, \bar{b}), \tag{16}$$

where $\bar{b}$ is the upper bound e of $|b|$, sign$(\cdot)$ represents a signum function. For a vector $\hat{b} = [\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_m]^T \in \mathbb{R}^m$,

$$\mathrm{sat}_{\bar{b}}(\hat{b}) := [\mathrm{sat}_{\bar{b}}(\hat{b}_1), \mathrm{sat}_{\bar{b}}(\hat{b}_2), \cdots, \mathrm{sat}_{\bar{b}}(\hat{b}_m)]^T.$$

It should be noted that, in the above robotic adaptive ILC algorithm, the initial systems error at each iteration is allowed to be any bounded error. Hence, the adaptive ILC algorithm may be used in the cases of any bounded initial state.

Remark 3: In fact, (14) may be regarded as the superposition of

$$\mu\boldsymbol{\vartheta}_k = \mu\boldsymbol{\vartheta}_{k-1} - \mu\gamma_2\Phi_k^T\boldsymbol{s}_k, \quad \boldsymbol{\vartheta}_{-1}(t) = 0 \tag{17}$$

and

$$\begin{aligned}
(1-\mu)\dot{\boldsymbol{\vartheta}}_k &= -(1-\mu)\gamma_2\Phi_k^T\boldsymbol{s}_k, \\
\boldsymbol{\vartheta}_k(0) &= \boldsymbol{\vartheta}_{k-1}(T), \quad \boldsymbol{\vartheta}_{-1}(T) = 0.
\end{aligned} \tag{18}$$

(17) is equivalent to the iteration-domain adaptive learning law

$$\boldsymbol{\vartheta}_k = \boldsymbol{\vartheta}_{k-1} - \gamma_2\Phi_k^T\boldsymbol{s}_k, \quad \boldsymbol{\vartheta}_{-1}(t) = 0, \tag{19}$$

(18) is equivalent to the time-domain adaptive learning law

$$\dot{\boldsymbol{\vartheta}}_k = -\gamma_2\Phi_k^T\boldsymbol{s}_k, \quad \boldsymbol{\vartheta}_k(0) = \boldsymbol{\vartheta}_{k-1}(T),\ \boldsymbol{\vartheta}_{-1}(T) = 0, \tag{20}$$

Hence, the adaptive learning law (14) is actually a combination of a time-domain adaptive learning law and an iteration-domain adaptive learning law. The combined adaptive leaning strategy has the following advantages:

(i) Compared with the time-domain adaptive learning law (18), the combined adaptive learning law (14) can obtain higher convergence speed of the tracking error.

(ii) Compared with the iteration-domain adaptive learning law (17), the combined adaptive learning law (14) can guarantee that $\boldsymbol{\vartheta}_k$ is bounded, while iteration-domain adaptive learning design (17) only guarantee $\boldsymbol{\vartheta}_k$ to be L_2 bounded.

More details on the combined adaptive learning strategy, see [29].

V. CONVERGENCE ANALYSIS

In this section, we will establish the following global convergence results on the proposed adaptive ILC scheme.

Theorem 1: For the closed-loop robotic systems composed of (3), adaptive ILC law (13) and learning laws (14)−(15), the tracking error converges in the sense that

$$\lim_{k\to+\infty}\tilde{\boldsymbol{q}}_k(t)=0, t\in[t_\epsilon,T]; \tag{21}$$

and all signals in the closed-loop robotic systems are guaranteed to be bounded.

Proof: Part A: Calculating difference of $L_k(t)$

First, substituting (13) into (12) yields

$$\dot{V}_k=-\gamma_1\boldsymbol{s}_k^T\boldsymbol{s}_k+\boldsymbol{s}_k^T\Phi_k\tilde{\boldsymbol{\vartheta}}_k+\boldsymbol{s}_k^TW_k\tilde{\boldsymbol{\eta}}_k, \tag{22}$$

where $\tilde{\boldsymbol{\vartheta}}_k=\boldsymbol{\vartheta}_k-\boldsymbol{\vartheta}, \tilde{\boldsymbol{\eta}}_k=\boldsymbol{\eta}_k-\boldsymbol{\eta}$, $\Phi_k\triangleq\Phi(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\boldsymbol{q}_{zk},\dot{\boldsymbol{q}}_{zk})$, $W_k\triangleq W(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\boldsymbol{q}_{zk},\dot{\boldsymbol{q}}_{zk})$. Calculating the definite integral to (22), we have

$$V_k=V_k(0)-\gamma_1\int_0^t\boldsymbol{s}_k^T\boldsymbol{s}_kd\tau+\int_0^t\boldsymbol{s}_k^T(\Phi_k\tilde{\boldsymbol{\vartheta}}_k+W_k\tilde{\boldsymbol{\eta}}_k)d\tau. \tag{23}$$

Then, we define a nonnegative functional as

$$L_k=V_k+\frac{\mu}{2\gamma_2}\int_0^t\tilde{\boldsymbol{\vartheta}}_k^T\tilde{\boldsymbol{\vartheta}}_kd\tau+\frac{(1-\mu)}{2\gamma_2}\tilde{\boldsymbol{\vartheta}}_k^T\tilde{\boldsymbol{\vartheta}}_k+\frac{1}{2\gamma_3}\int_0^t\tilde{\boldsymbol{\eta}}_k^T\tilde{\boldsymbol{\eta}}_kd\tau. \tag{24}$$

While $k>0$, according to (23) and (24), the difference of $L_k(t)$ between two adjacent iterations is

$$\begin{aligned}
&L_k-L_{k-1}\\
&=V_k-V_{k-1}+\frac{\mu}{2\gamma_2}\int_0^t(\tilde{\boldsymbol{\vartheta}}_k^T\tilde{\boldsymbol{\vartheta}}_k-\tilde{\boldsymbol{\vartheta}}_{k-1}^T\tilde{\boldsymbol{\vartheta}}_{k-1})d\tau\\
&\quad+\frac{(1-\mu)}{2\gamma_2}(\tilde{\boldsymbol{\vartheta}}_k^T\tilde{\boldsymbol{\vartheta}}_k-\tilde{\boldsymbol{\vartheta}}_{k-1}^T\tilde{\boldsymbol{\vartheta}}_{k-1})\\
&\quad+\frac{1}{2\gamma_3}\int_0^t(\tilde{\boldsymbol{\eta}}_k^T\tilde{\boldsymbol{\eta}}_k-\tilde{\boldsymbol{\eta}}_{k-1}^T\tilde{\boldsymbol{\eta}}_{k-1})d\tau\\
&=-\gamma_1\int_0^t\boldsymbol{s}_k^T\boldsymbol{s}_kd\tau+\int_0^t\boldsymbol{s}_k^T(\Phi_k\tilde{\boldsymbol{\vartheta}}_k+W_k\tilde{\boldsymbol{\eta}}_k)d\tau\\
&\quad+\frac{\mu}{2\gamma_2}\int_0^t(\tilde{\boldsymbol{\vartheta}}_k^T\tilde{\boldsymbol{\vartheta}}_k-\tilde{\boldsymbol{\vartheta}}_{k-1}^T\tilde{\boldsymbol{\vartheta}}_{k-1})d\tau\\
&\quad+\frac{(1-\mu)}{\gamma_2}\int_0^t\dot{\tilde{\boldsymbol{\vartheta}}}_k^T\tilde{\boldsymbol{\vartheta}}_kd\tau+\frac{(1-\mu)}{2\gamma_2}\tilde{\boldsymbol{\vartheta}}_k^T(0)\tilde{\boldsymbol{\vartheta}}_k(0)\\
&\quad-\frac{(1-\mu)}{2\gamma_2}\tilde{\boldsymbol{\vartheta}}_{k-1}^T\tilde{\boldsymbol{\vartheta}}_{k-1}+\frac{1}{2\gamma_3}\int_0^t(\tilde{\boldsymbol{\eta}}_k^T\tilde{\boldsymbol{\eta}}_k-\tilde{\boldsymbol{\eta}}_{k-1}^T\tilde{\boldsymbol{\eta}}_{k-1})d\tau\\
&\quad-V_{k-1}+V_k(0).
\end{aligned} \tag{25}$$

Through direct algebraic operations yields

$$\begin{aligned}
&\tilde{\boldsymbol{\vartheta}}_k^T\tilde{\boldsymbol{\vartheta}}_k-\tilde{\boldsymbol{\vartheta}}_{k-1}^T\tilde{\boldsymbol{\vartheta}}_{k-1}\\
&\quad=(2\boldsymbol{\vartheta}-\boldsymbol{\vartheta}_k-\boldsymbol{\vartheta}_{k-1})^T(\boldsymbol{\vartheta}_{k-1}-\boldsymbol{\vartheta}_k)\\
&\quad=(2\boldsymbol{\vartheta}-2\boldsymbol{\vartheta}_k+\boldsymbol{\vartheta}_k-\boldsymbol{\vartheta}_{k-1})^T(\boldsymbol{\vartheta}_{k-1}-\boldsymbol{\vartheta}_k)\\
&\quad\leq-2(\boldsymbol{\vartheta}_{k-1}-\boldsymbol{\vartheta}_k)^T\tilde{\boldsymbol{\vartheta}}_k.
\end{aligned} \tag{26}$$

Combining (14) with (26), we have

$$\begin{aligned}
&-\boldsymbol{s}_k^T\Phi_k\tilde{\boldsymbol{\vartheta}}_k+\frac{\mu}{2\gamma_2}(\tilde{\boldsymbol{\vartheta}}_k^T\tilde{\boldsymbol{\vartheta}}_k-\tilde{\boldsymbol{\vartheta}}_{k-1}^T\tilde{\boldsymbol{\vartheta}}_{k-1})+\frac{(1-\mu)}{\gamma_2}\dot{\tilde{\boldsymbol{\vartheta}}}_k^T\tilde{\boldsymbol{\vartheta}}_k\\
&=\frac{1}{\gamma_2}\left[\gamma_2\boldsymbol{s}_k^T\Phi_k-\mu(\boldsymbol{\vartheta}_{k-1}-\boldsymbol{\vartheta}_k)+(1-\mu)\dot{\boldsymbol{\vartheta}}_k\right]^T\tilde{\boldsymbol{\vartheta}}_k\\
&=0.
\end{aligned} \tag{27}$$

Substituting (27) into (25) leads to

$$\begin{aligned}
&L_k-L_{k-1}\\
&\quad=V_k(0)-\gamma_1\int_0^t\boldsymbol{s}_k^T\boldsymbol{s}_kd\tau+\int_0^t\boldsymbol{s}_k^TW_k\tilde{\boldsymbol{\eta}}_kd\tau\\
&\qquad+\frac{(1-\mu)}{2\gamma_2}\tilde{\boldsymbol{\vartheta}}_k^T(0)\tilde{\boldsymbol{\vartheta}}_k(0)-\frac{(1-\mu)}{2\gamma_2}\tilde{\boldsymbol{\vartheta}}_{k-1}^T\tilde{\boldsymbol{\vartheta}}_{k-1}-V_{k-1}\\
&\qquad+\frac{1}{2\gamma_3}\int_0^t(\tilde{\boldsymbol{\eta}}_k^T\tilde{\boldsymbol{\eta}}_k-\tilde{\boldsymbol{\eta}}_{k-1}^T\tilde{\boldsymbol{\eta}}_{k-1})d\tau.
\end{aligned} \tag{28}$$

According to the relationship $(a-b)^T(a-b)\leq(a-\text{sat}_{\bar{b}}(\hat{b}))^T(a-\text{sat}_{\bar{b}}(\hat{b}))$ and (15), the following inequality is satisfied:

$$\begin{aligned}
&\frac{1}{2\gamma_3}(\tilde{\boldsymbol{\eta}}_k^T\tilde{\boldsymbol{\eta}}_k-\tilde{\boldsymbol{\eta}}_{k-1}^T\tilde{\boldsymbol{\eta}}_{k-1})+\boldsymbol{s}_k^TW_k\tilde{\boldsymbol{\eta}}_k\\
&\quad=\frac{1}{2\gamma_3}(\boldsymbol{\eta}-\boldsymbol{\eta}_k)^T(\boldsymbol{\eta}-\boldsymbol{\eta}_k)-(\boldsymbol{\eta}-\boldsymbol{\eta}_{k-1})^T(\boldsymbol{\eta}-\boldsymbol{\eta}_{k-1})+\boldsymbol{s}_k^TW_k\tilde{\boldsymbol{\eta}}_k\\
&\quad\leq\frac{1}{2\gamma_3}(\boldsymbol{\eta}-\boldsymbol{\eta}_k)^T(\boldsymbol{\eta}-\boldsymbol{\eta}_k)-(\boldsymbol{\eta}-\text{sat}(\boldsymbol{\eta}_{k-1}))^T(\boldsymbol{\eta}-\text{sat}(\boldsymbol{\eta}_{k-1}))\\
&\qquad+\boldsymbol{s}_k^TW_k\tilde{\boldsymbol{\eta}}_k\\
&\quad=\frac{1}{2\gamma_3}(2\boldsymbol{\eta}-\boldsymbol{\eta}_k-\text{sat}_{\bar{\eta}}(\boldsymbol{\eta}_{k-1}))^T(\text{sat}_{\bar{\eta}}(\boldsymbol{\eta}_{k-1})-\boldsymbol{\eta}_k)+\boldsymbol{s}_k^TW_k\tilde{\boldsymbol{\eta}}_k\\
&\quad\leq-\frac{1}{\gamma_3}\tilde{\boldsymbol{\eta}}_k^T(\text{sat}_{\bar{\eta}}(\boldsymbol{\eta}_{k-1})-\boldsymbol{\eta}_k)+\boldsymbol{s}_k^TW_k\tilde{\boldsymbol{\eta}}_k=0.
\end{aligned} \tag{29}$$

After substituting (29) into (28), the following expression is obtained:

$$\begin{aligned}
L_k-L_{k-1}\leq{}&V_k(0)-\gamma_1\int_0^t\boldsymbol{s}_k^T\boldsymbol{s}_kd\tau+\frac{(1-\mu)}{2\gamma_2}\tilde{\boldsymbol{\vartheta}}_k^T(0)\tilde{\boldsymbol{\vartheta}}_k(0)\\
&-\frac{(1-\mu)}{2\gamma_2}\tilde{\boldsymbol{\vartheta}}_{k-1}^T\tilde{\boldsymbol{\vartheta}}_{k-1}-V_{k-1}.
\end{aligned} \tag{30}$$

According to (4) and (14), we can see $V_k(0)=0$ and $\boldsymbol{\vartheta}_k(T)=\boldsymbol{\vartheta}_{k-1}(0)$ hold, respectively. Therefore, it follows from (30) that

$$L_k(T)-L_{k-1}(T)\leq-\gamma_1\int_0^T\boldsymbol{s}_k^T\boldsymbol{s}_kd\tau. \tag{31}$$

Similarly, it may be concluded that

$$\begin{aligned}
L_{k-1}(T)-L_{k-2}(T)&\leq-\gamma_1\int_0^T\boldsymbol{s}_{k-1}^T\boldsymbol{s}_{k-1}d\tau,\\
L_{k-2}(T)-L_{k-3}(T)&\leq-\gamma_1\int_0^T\boldsymbol{s}_{k-2}^T\boldsymbol{s}_{k-2}\tau,\\
&\vdots\\
L_1(T)-L_0(T)&\leq-\gamma_1\int_0^T\boldsymbol{s}_1^T\boldsymbol{s}_1d\tau.
\end{aligned} \tag{32}$$

The sum of the two sides of above k inequalities in (31) and (32) is

$$L_k(T) - L_0(T) \leq -\gamma_1 \sum_{j=1}^{k} \int_0^T \boldsymbol{s}_j^T \boldsymbol{s}_j d\tau,$$

which implies

$$L_k(T) \leq L_0(T) - \gamma_1 \sum_{j=1}^{k} \int_0^T \boldsymbol{s}_j^T \boldsymbol{s}_j d\tau. \tag{33}$$

Part B: Boundedness of $L_0(t)$

Taking derivative of $L_0(t)$ with respect to t yields

$$\begin{aligned}\dot{L}_0 = -\gamma_1 \boldsymbol{s}_0^T \boldsymbol{s}_0 + \boldsymbol{s}_0^T \Phi_0 \tilde{\boldsymbol{\vartheta}}_0 + \boldsymbol{s}_0^T W_0 \tilde{\boldsymbol{\eta}}_0 + \frac{\mu}{2\gamma_2} \tilde{\boldsymbol{\vartheta}}_0^T \tilde{\boldsymbol{\vartheta}}_0 \\ + \frac{(1-\mu)}{\gamma_2} \dot{\tilde{\boldsymbol{\vartheta}}}_0^T \tilde{\boldsymbol{\vartheta}}_0 + \frac{1}{2\gamma_3} \tilde{\boldsymbol{\eta}}_0^T \tilde{\boldsymbol{\eta}}_0.\end{aligned} \tag{34}$$

By using learning law (14), we obtain

$$\begin{aligned}(1-\mu)\dot{\boldsymbol{\vartheta}}_0 &= -\gamma_2 W_0^T \boldsymbol{s}_0 + \mu(\boldsymbol{\vartheta}_{-1} - \boldsymbol{\vartheta}_0) \\ &= -\gamma_2 \Phi_0 \boldsymbol{s}_0 - \mu \boldsymbol{\vartheta}_0.\end{aligned} \tag{35}$$

Then, combining (35) with (34), we have

$$\begin{aligned}\dot{L}_0 &= -\gamma_1 \boldsymbol{s}_0^T \boldsymbol{s}_0 + \boldsymbol{s}_0^T W_0 \tilde{\boldsymbol{\eta}}_0 + \frac{1}{\gamma_2}(\gamma_2 \boldsymbol{s}_0^T \Phi_0 \tilde{\boldsymbol{\vartheta}}_0 + \frac{\mu}{2} \tilde{\boldsymbol{\vartheta}}_0^T \tilde{\boldsymbol{\vartheta}}_0 \\ &\quad + (1-\mu)\dot{\tilde{\boldsymbol{\vartheta}}}_0^T \tilde{\boldsymbol{\vartheta}}_0) + \frac{1}{2\gamma_3} \tilde{\boldsymbol{\eta}}_0^T \tilde{\boldsymbol{\eta}}_0. \\ &= -\gamma_1 \boldsymbol{s}_0^T \boldsymbol{s}_0 + \boldsymbol{s}_0^T W_0 \tilde{\boldsymbol{\eta}}_0 + \frac{1}{\gamma_2}(\frac{\mu}{2} \tilde{\boldsymbol{\vartheta}}_0^T \tilde{\boldsymbol{\vartheta}}_0 - \mu \boldsymbol{\vartheta}_0 \tilde{\boldsymbol{\vartheta}}_0) \\ &\quad + \frac{1}{2\gamma_3} \tilde{\boldsymbol{\eta}}_0^T \tilde{\boldsymbol{\eta}}_0.\end{aligned} \tag{36}$$

Due to

$$\begin{aligned}\frac{1}{2} \tilde{\boldsymbol{\vartheta}}_0^T \tilde{\boldsymbol{\vartheta}}_0 - \boldsymbol{\vartheta}_0 \tilde{\boldsymbol{\vartheta}}_0 &= \tfrac{1}{2}(\boldsymbol{\vartheta}_0 - \boldsymbol{\vartheta})^T (\boldsymbol{\vartheta}_0 - \boldsymbol{\vartheta}) - \boldsymbol{\vartheta}_0^T (\boldsymbol{\vartheta}_0 - \boldsymbol{\vartheta}) \\ &= -\tfrac{1}{2} \boldsymbol{\vartheta}_0^T \boldsymbol{\vartheta}_0 + \tfrac{1}{2} \boldsymbol{\vartheta}^T \boldsymbol{\vartheta},\end{aligned} \tag{37}$$

we rewrite (36) as follows:

$$\begin{aligned}\dot{L}_0 &= -\gamma_1 \boldsymbol{s}_0^T \boldsymbol{s}_0 + \boldsymbol{s}_0^T W_0 \tilde{\boldsymbol{\eta}}_0 + \frac{1}{\gamma_2}(\gamma_2 \boldsymbol{s}_0^T \Phi_0 \tilde{\boldsymbol{\vartheta}}_0 + \frac{\mu}{2} \tilde{\boldsymbol{\vartheta}}_0^T \tilde{\boldsymbol{\vartheta}}_0 \\ &\quad + (1-\mu)\dot{\tilde{\boldsymbol{\vartheta}}}_0^T \tilde{\boldsymbol{\vartheta}}_0) + \frac{1}{2\gamma_3} \tilde{\boldsymbol{\eta}}_0^T \tilde{\boldsymbol{\eta}}_0. \\ &= -\gamma_1 \boldsymbol{s}_0^T \boldsymbol{s}_0 - \frac{\mu}{2\gamma_2} \boldsymbol{\vartheta}_0^T \boldsymbol{\vartheta}_0 + \frac{\mu}{2\gamma_2} \boldsymbol{\vartheta}^T \boldsymbol{\vartheta} + \boldsymbol{s}_0^T W_0 \tilde{\boldsymbol{\eta}}_0 \\ &\quad + \frac{1}{2\gamma_3} \tilde{\boldsymbol{\eta}}_0^T \tilde{\boldsymbol{\eta}}_0.\end{aligned} \tag{38}$$

On the other hand, it follows from (14) that

$$\begin{aligned}&\boldsymbol{s}_0^T W_0 \tilde{\boldsymbol{\eta}}_0 + \frac{1}{2\gamma_3} \tilde{\boldsymbol{\eta}}_0^T \tilde{\boldsymbol{\eta}}_0 \\ &\quad = \frac{1}{\gamma_3}(-\boldsymbol{\eta}_0^T (\boldsymbol{\eta}_0 - \boldsymbol{\eta}) + \frac{1}{2}(\boldsymbol{\eta}_0 - \boldsymbol{\eta})^T (\boldsymbol{\eta}_0 - \boldsymbol{\eta})) \\ &\quad = \frac{1}{\gamma_3}(-\frac{1}{2} \boldsymbol{\eta}_0^T \boldsymbol{\eta}_0 + \frac{1}{2} \boldsymbol{\eta}^T \boldsymbol{\eta}).\end{aligned} \tag{39}$$

Substituting (39) into (38) yields

$$\begin{aligned}\dot{L}_0 = -\gamma_1 \boldsymbol{s}_0^T \boldsymbol{s}_0 - \frac{\mu}{2\gamma_2} \boldsymbol{\vartheta}_0^T \boldsymbol{\vartheta}_0 - \frac{1}{2\gamma_3} \boldsymbol{\eta}_0^T + \frac{1}{2\gamma_2} \boldsymbol{\vartheta}^T \boldsymbol{\vartheta} \boldsymbol{\eta}_0 \\ + \frac{1}{2\gamma_3} \boldsymbol{\eta}^T \boldsymbol{\eta}.\end{aligned} \tag{40}$$

Based on (40), it is clear that $L_0(t)$ is bounded for $t \in [0, T]$.

Part C: Convergence property of the closed-loop system

Since $L_0(T)$ is bounded, from (33), the following expression can be obtained:

$$\lim_{k \to +\infty} \int_0^T \boldsymbol{s}_k^T \boldsymbol{s}_k d\tau = 0. \tag{41}$$

On the other hand, from (30) and (31), we have

$$\begin{aligned}L_k(t) &= \frac{\mu}{2\gamma_2} \int_0^t \tilde{\boldsymbol{\vartheta}}_k^T \tilde{\boldsymbol{\vartheta}}_k d\tau + \frac{1}{2\gamma_3} \int_0^t \tilde{\boldsymbol{\eta}}_k^T \tilde{\boldsymbol{\eta}}_k d\tau - \gamma_1 \int_0^t \boldsymbol{s}_k^T \boldsymbol{s}_k d\tau \\ &\quad + \frac{(1-\mu)}{2\gamma_2} \tilde{\boldsymbol{\vartheta}}_k^T(0) \tilde{\boldsymbol{\vartheta}}_k(0) \\ &\leq L_{k-1}(T) - \gamma_1 \int_0^t \boldsymbol{s}_k^T \boldsymbol{s}_k d\tau \\ &\leq L_0(T) - \gamma_1 \int_0^t \boldsymbol{s}_k^T \boldsymbol{s}_k d\tau.\end{aligned} \tag{42}$$

Since $L_0(T)$ is bounded, from (42), we can easily obtain the boundedness of L_k, based on which, the boundedness of $\boldsymbol{s}_k$, $\tilde{\boldsymbol{\vartheta}}_k$ and $\boldsymbol{\vartheta}_k$ can be easily gotten. Then, it follows from the boundedness of $\boldsymbol{s}_k$, z_k and $\dot{z}_k$ are guaranteed to be bound. Applying the boundedness of z_k and $\dot{z}_k$, we can deduce that $\tilde{\boldsymbol{q}}_k, \dot{\tilde{\boldsymbol{q}}}_k, \boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \dot{\boldsymbol{q}}_{zk}$ and $\ddot{\boldsymbol{q}}_{zk}$ are all bounded. Furthermore, we can concluded that Φ_k and W_k are bounded. By the property of saturation function and the boundedness of W_k, we can draw a conclusion that $\boldsymbol{\eta}_k$ is bounded from (15). By using the above conclusions, we can get the boundedness of $\boldsymbol{\tau}_k$ from (13). Since $\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k$ and $\boldsymbol{\tau}_k$ are proved to be bounded, we can easily obtain the boudedness of $\ddot{q}_k$ from (1). From the definition of $\boldsymbol{s}_k$, the boundedness of $\dot{\boldsymbol{s}}_k$ can be deduced, which means $\boldsymbol{s}_k(t)$ is equicontinuous over $[0, T]$. In light of this conclusion and (41), we obtain

$$\lim_{k \to +\infty} \boldsymbol{s}_k(t) = 0, \quad t \in [0, T]. \tag{43}$$

Since $z_k(0) = 0$ and $\dot{z}_k(0) = 0$, from (43) we can further conclude

$$\lim_{k \to +\infty} z_k(t) = 0, \quad t \in [0, T], \tag{44}$$

which implies

$$\lim_{k \to +\infty} \tilde{\boldsymbol{q}}_k(t) = 0, \quad t \in [t_\epsilon, T].\blacksquare \tag{45}$$

■

Remark 4: In (30), if $V_k(0) \neq 0$. It follows that

$$L_k(T) - L_{k-1}(T) \leq V_k(0) - \gamma_1 \int_0^T \boldsymbol{s}_k^T \boldsymbol{s}_k d\tau. \tag{46}$$

A similar conclusion to (33) is

$$L_k(T) \leq L_0(T) + \sum_{j=1}^{k} V_j(0) - \gamma_1 \sum_{j=1}^{k} \int_0^T \boldsymbol{s}_j^T \boldsymbol{s}_j d\tau. \tag{47}$$

If $\lim_{k\to\infty}\sum_{j=1}^{k} V_j(0)$ is not bounded, the conclusion $\lim_{k\to+\infty}\boldsymbol{s}_k(t) = 0, t \in [0, T]$ can not be drawn even if $L_0(T)$ is bounded. Hence, it is really necessary to deal with the nonzero initial error during controller design. As far as error-tracking strategy is concerned, through the reasonable construction of desired error trajectory, $\lim_{k\to\infty}\sum_{j=1}^{k} V_j(0) = 0$ holds.

VI. ILLUSTRATIVE EXAMPLE

To verify the effectiveness of the proposed adaptive learning control scheme, the following two-link robot manipulator model is considered [22]:

$$M(\boldsymbol{q},\boldsymbol{\phi}(t))\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi}(t))\ddot{\boldsymbol{q}} + F(\boldsymbol{q},\dot{\boldsymbol{\phi}}(t))\dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{48}$$

where

$$\begin{aligned}
M(\boldsymbol{q},\boldsymbol{\phi}(t)) &= \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix},\\
C(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\phi}(t)) &= \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & 0 \end{bmatrix},\\
F(\boldsymbol{q},\dot{\boldsymbol{\phi}}(t)) &\\
&= \begin{bmatrix} l_1^2+l_2^2+2l_1l_2\cos(q_2) & l_2^2+l_1l_2\cos(q_2) \\ l_2^2+l_1l_2\cos(q_2) & l_2^2 \end{bmatrix}\dot{m}_p\\
&\quad + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\dot{I}_p,\\
m_{11} &= p_1 + 2p_2\cos(q_2) + v_1(q_2)m_p(t) + I_p(t),\\
m_{12} &= p_3 + p_2\cos(q_2) + v_2(q_2)m_p(t) + I_p(t),\\
m_{21} &= p_3 + p_2\cos(q_2) + v_2(q_2)m_p(t) + I_p(t),\\
m_{22} &= p_3 + v_3 m_p(t) + I_p(t),\\
c_{11} &= -\dot{q}_2(p_2 + l_1l_2m_p(t))\sin(q_2),\\
c_{12} &= -(\dot{q}_1 + \dot{q}_2)(p_2 + l_1l_2m_p(t))\sin(q_2),\\
c_{21} &= \dot{q}_1(p_2 + l_1l_2m_p(t))\sin(q_2).
\end{aligned}$$

Here, $m_p(t)$ and $I_p(t)$ are the payload mass and inertia, respectively, $I_p(t) = \frac{1}{2}m_p(t)R^2, m_p = k_1 t$, where k_1 represents the constant water flow rate in or out of the vessel, R is the radius of the cylindrical vessel. $\boldsymbol{\phi}(t) = [p_1, p_2, p_3, m_p(t), I_p(t)]^T$, where p_1, p_2, p_3 are the constant coupled parameters of the robot manipulator that contain masses and inertias of the links and the motors. l_1 and l_2 are the lengths of two links, $v_1 = l_1^2 + l_2^2 + 2l_1l_2\cos(q_2)$ and $v_2 = l_2^2 + l_1l_2\cos(q_2)$.

The desired trajectory $\boldsymbol{q}_d = [0.8\cos(t), 0.5\cos(\frac{\pi t}{2})], t \in [0, T], T = 5$. The initial state of the system is $\boldsymbol{q}_k(0) = [1 + 0.1\mathrm{r}_1, 0.8 + 0.1\mathrm{r}_2], d\boldsymbol{q}_k(0) = [0.05\mathrm{r}_3, 0.05\mathrm{r}_4]$, where r_1–r_4 are random numbers between 0 and 1. Some other parameters are set in the simulation as $T = 5, p_1 = 3.4, p_2 = 0.2, p_3 = 0.15, l_1 = 1, l_2 = 0.6, m_p = 0.3t, I_p = 0.01m_p/2, R = 0.1\ m, k_1 = 0.3\ kg/s$.

After constructing the desired error trajectory according to Section 3, we develop the control law and learning laws according to (13)-(15) with $t_\epsilon = 0.6, \lambda = 5, \gamma_1 = 5, \gamma_2 = 2, \gamma_3 = 2, \mu = 0.9, \bar{\eta} = 30$. $\boldsymbol{\vartheta}_k$ is used to estimate $\boldsymbol{\vartheta} = [p_1, p_2, p_3]^T$ according to combined adaptive learning law (14), $\boldsymbol{\eta}_k$ is used to estimate $\boldsymbol{\eta} = [m_p, I_p, \dot{m}_p, \dot{I}_p]^T$ according to difference learning law (15). Through proper parameter reorganization, we get

$$\begin{aligned}
\Phi(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\dot{\boldsymbol{q}}_{zk},\ddot{\boldsymbol{q}}_{zk}) &= \begin{bmatrix} \alpha_{z1} & \alpha_{z2} & \alpha_{z3} \\ 0 & \alpha_{z4} & \alpha_{z5} \end{bmatrix},\\
W(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\dot{\boldsymbol{q}}_{zk},\ddot{\boldsymbol{q}}_{zk}) &= \begin{bmatrix} \omega_{z1} & \omega_{z2} & \omega_{z3} & \omega_{z4} \\ \omega_{z5} & \omega_{z6} & \omega_{z7} & \omega_{z8} \end{bmatrix},
\end{aligned} \tag{49}$$

where

$$\begin{aligned}
\alpha_{z1} &= \ddot{q}_{z1,k},\\
\alpha_{z2} &= 2\ddot{q}_{z1,k}\cos(q_{2,k}) + \ddot{q}_{z2,k}\cos(q_{2,k}) - \dot{q}_{2,k}\sin(q_{2,k})\dot{q}_{z1,k}\\
&\quad -(\dot{q}_{1,k} + \dot{q}_{2,k})\sin(q_{2,k})\dot{q}_{z2,k},\\
\alpha_{z3} &= \ddot{q}_{z2,k},\\
\alpha_{z4} &= \ddot{q}_{z1,k}\cos(q_{2,k}) + \dot{q}_{1,k}\sin(q_{2,k})\dot{q}_{z1,k},\\
\alpha_{z5} &= \ddot{q}_{z1,k} + \ddot{q}_{z2,k},\\
\omega_{z1} &= (l_1^2 + l_2^2 + 2l_1l_2\cos(q_{2,k}))\ddot{q}_{z1,k} + (l_2^2 + l_1l_2\cos(q_{2,k}))\\
&\quad \times\ddot{q}_{z2,k} - l_1l_2[\dot{q}_{2,k}\sin(q_{2,k})\dot{q}_{z1,k} + (\dot{q}_{1,k} + \dot{q}_{2,k})\\
&\quad \times\sin(q_{2,k})\dot{q}_{z2,k}]\\
\omega_{z2} &= \ddot{q}_{z1,k} + \ddot{q}_{z2,k},\\
\omega_{z3} &= (l_1^2 + l_2^2)\xi_{z1} + 2l_1l_2\xi_{z1}\cos(q_{2,k}) + l_2^2\xi_{z2}\\
&\quad +l_1l_2\xi_{z2}\cos(q_{2,k}),\\
\omega_{z4} &= \xi_{z1} + \xi_{z2},\\
\omega_{z5} &= (l_2^2 + l_1l_2\cos(q_{2,k}))\ddot{q}_{z1,k} + l_2^2\ddot{q}_{z2,k}\\
&\quad +\dot{q}_{1,k}l_1l_2\sin(q_{2,k})\dot{q}_{z1,k},\\
\omega_{z6} &= \ddot{q}_{z1,k} + \ddot{q}_{z2,k},\\
\omega_{z7} &= l_2^2\xi_{z1} + l_1l_2\xi_{z1}\cos(q_{2,k}) + l_2^2\xi_{z2},\\
\omega_{z8} &= \omega_{z6},\\
\xi_{z1} &= \frac{1}{2}\dot{q}_{z1,k} + \frac{1}{2}\dot{q}_{1,k},\\
\xi_{z2} &= \frac{1}{2}\dot{q}_{z2,k} + \frac{1}{2}\dot{q}_{2,k}.
\end{aligned}$$

The simulation results are shown in Figs. 1-15. From Figs. 1-4, we can see that after 20 iterations, $q_1, \dot{q}_1, q_2$ and $\dot{q}_2$ converge to $q_{1,d}, \dot{q}_{1,d}, q_{2,d}$ and $\dot{q}_{2,d}$ for $t \in [t_\epsilon, T]$, respectively. The profiles of position/velocity errors and corresponding desired error trajectories in the 20th iteration are shown in Figs. 5-8. Figs. 9-12 show the profile of differences between robotic system error and the desired error trajectories at the 20th iteration. Figs. 5-12 indicate that $\tilde{q}_1, \dot{\tilde{q}}_1, \tilde{q}_2, \dot{\tilde{q}}_2$ follow $\tilde{q}_1^d, \dot{\tilde{q}}_1^d, \tilde{q}_2^d, \dot{\tilde{q}}_2^d$ over the interval $[t_\epsilon, T]$, respectively. Figs. 13-14 give the torque input for link 1 and link 2 at the 20th iteration, respectively. Fig. 15 illustrates the learning converge profile of output error in the robotic closed loop system, where $J_k \triangleq \max_{t\in[t_\epsilon,T]}(|\tilde{q}_{1,k}(t)| + |\tilde{q}_{2,k}(t)|)$.

For comparison, we present two simulation examples as below:

Comparison A: Traditional D-type learning law [41] is adopted to simulation as follows:

$$\boldsymbol{\tau}_k = \boldsymbol{\tau}_{k-1} + \Gamma[\boldsymbol{q}_d - \boldsymbol{q}_k]^T, \tag{50}$$

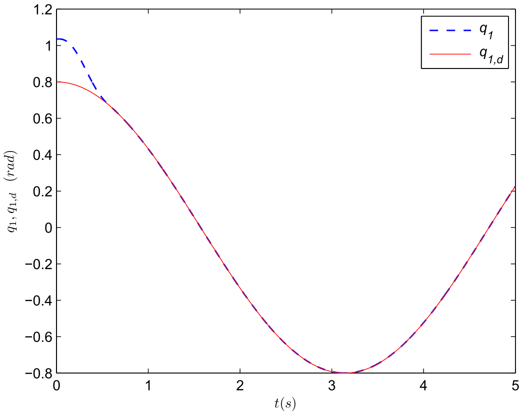


FIGURE 1. Position trajectory of joint 1 (error-tracking ILC).

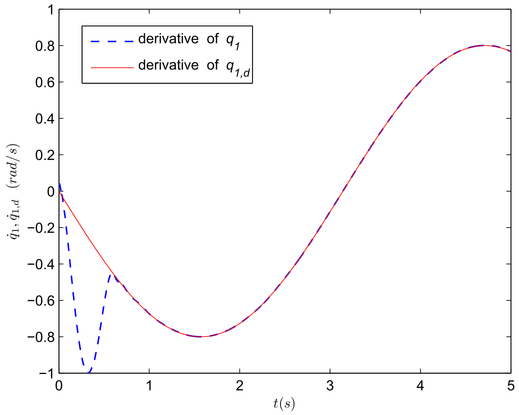


FIGURE 2. Velocity trajectory of joint 1 (error-tracking ILC).

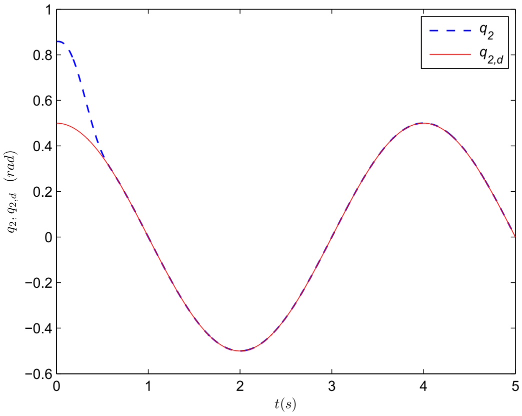


FIGURE 3. Position trajectories of joint 2 (error-tracking ILC).

where $\Gamma = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.9 \end{bmatrix}$ is the learning gain. The position trajectories at the 20th iteration are shown in Figs. 16-17. The maximum position error of 100 iterations is illustrated in Fig. 18, where the definition of J_k is the same as above, we can see the tracking error can not converge to zero or the small neighborhood of zero even if after so many iterations. From Figs. 16-18, we can see that traditional D-type ILC algorithms are not suitable for the robotic control system with arbitrary initial errors.

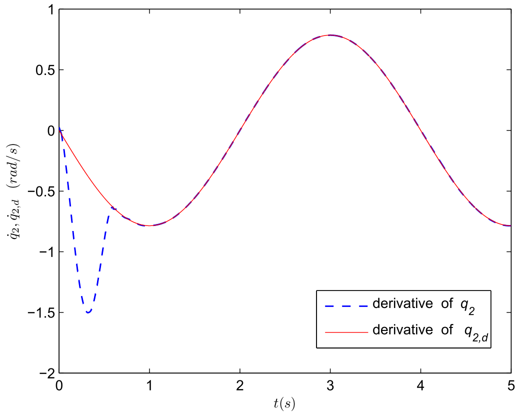


FIGURE 4. Velocity trajectory of joint 2 (error-tracking ILC).

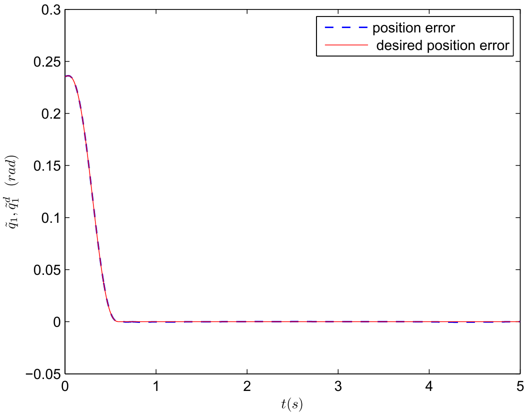


FIGURE 5. Position error $\tilde{q}_1$ and its desired $\tilde{q}_1^d$ (error-tracking ILC).

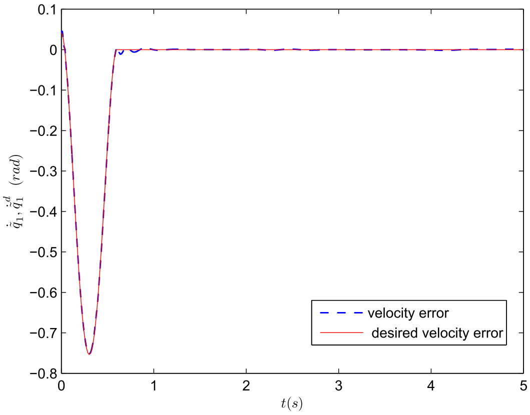


FIGURE 6. Velocity error $\dot{\tilde{q}}_1$ and its desired $\dot{\tilde{q}}_1^d$ (error-tracking ILC).

Comparison B: An initial-rectification adaptive ILC algorithm is adopted to the following simulation. This algorithm has been verified to be effective for robotic manipulators with time-invariant parameters in [42].

$$\boldsymbol{\tau}_k = -\gamma_4 \boldsymbol{s}_{rk} + W_r(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \dot{\boldsymbol{q}}_{k,r}, \ddot{\boldsymbol{q}}_{k,r})\boldsymbol{\theta}_k, \tag{51}$$

$$\boldsymbol{\theta}_k = \mathrm{sat}_{\bar{\theta}}(\hat{\boldsymbol{\theta}}_{k-1}) - \gamma_5 W_r(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \dot{\boldsymbol{q}}_{k,r}, \ddot{\boldsymbol{q}}_{k,r})\boldsymbol{s}_{rk}, \boldsymbol{\theta}_{-1} = 0, \tag{52}$$

where $\boldsymbol{s}_{rk} = (\dot{\boldsymbol{q}}_k - \dot{\boldsymbol{q}}_{k,r}) + \lambda(\boldsymbol{q}_k - \boldsymbol{q}_{k,r}), \boldsymbol{q}_{k,r} = [q_{1,k,r}, q_{2,k,r}]^T$. $q_{1,k,r}(t)$ and $q_{2,k,r}(t)$ are formed according to (53).

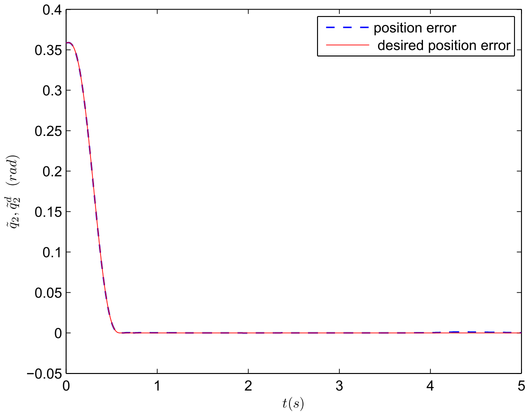


FIGURE 7. Position error $\tilde{q}_2$ and its desired $\tilde{q}_2^d$ (error-tracking ILC).

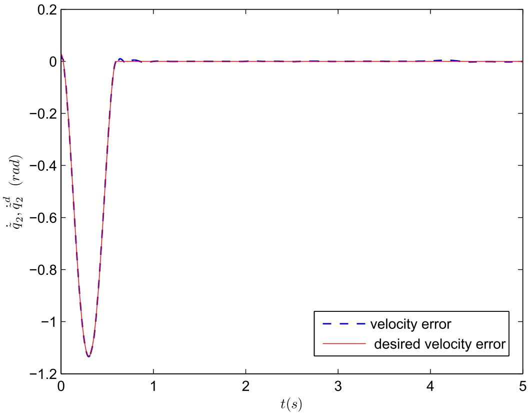


FIGURE 8. Velocity error $\dot{\tilde{q}}_2$ and its desired $\dot{\tilde{q}}_2^d$ (error-tracking ILC).

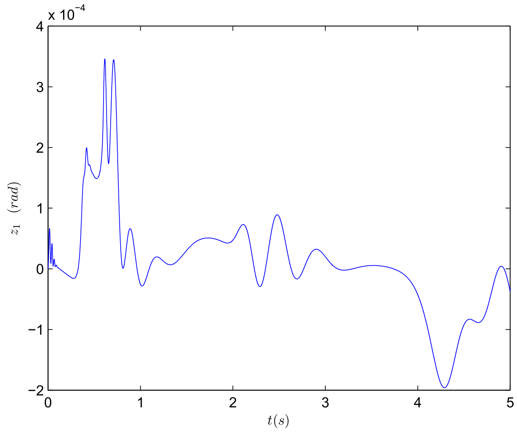


FIGURE 9. The difference between $\tilde{q}_1$ and $\tilde{q}_1^d$ (error-tracking ILC).

$$q_{i,k,r}(t) = \xi_i \cdot q^r_{i,k}(t) + (1-\xi_i)\cdot q_{i,d}(t), \quad i = 1, 2, \tag{53}$$

in which

$$\xi_i = \begin{cases} 1, & \text{if } t \in [0, t_\epsilon), \\ 0, & \text{if } t \in [t_\epsilon, T], \end{cases} \tag{54}$$

and $q^r_{i,k}(t) = A_{i,5}t^5 + A_{i,4}t^4 + A_{i,3}t^3 + A_{i,2}t^2 + A_{i,1}t + A_{i,0}$, with $A_{i,0} = q_{i,k}(0)$, $A_{i,1} = \dot{q}_{i,k}(0)$, $A_{i,2} = 0$, $A_{i,3} = \frac{10}{t_\epsilon^3}\varrho_1 - \frac{4}{t_\epsilon^2}\varrho_2 + \frac{1}{2t_\epsilon}\varrho_3$, $A_{i,4} = -\frac{15}{t_\epsilon^4}\varrho_1 + \frac{7}{t_\epsilon^3}\varrho_2 - \frac{1}{t_\epsilon^2}\varrho_3$,

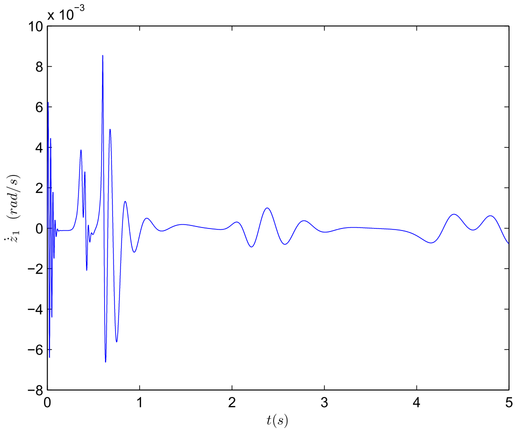


FIGURE 10. The difference between $\dot{\tilde{q}}_1$ and $\dot{\tilde{q}}_1^d$ (error-tracking ILC).

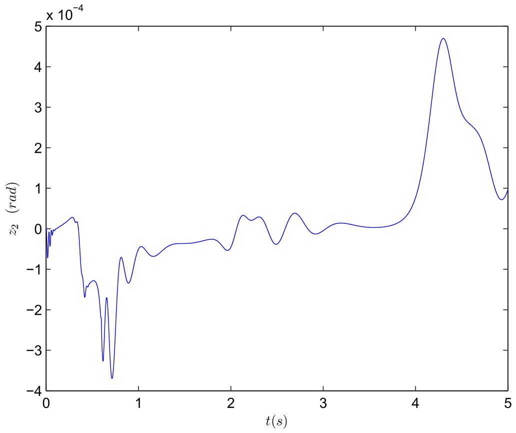


FIGURE 11. The difference between $\tilde{q}_2$ and $\tilde{q}_2^d$ (error-tracking ILC).

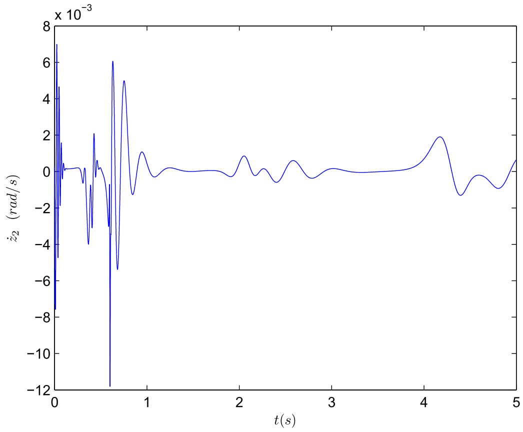


FIGURE 12. The difference between $\dot{\tilde{q}}_2$ and $\dot{\tilde{q}}_2^d$ (error-tracking ILC).

$A_{i,5} = \frac{6}{t_\epsilon^5}\varrho_1 - \frac{3}{t_\epsilon^4}\varrho_2 + \frac{1}{2t_\epsilon^3}\varrho_3$, $\varrho_1 = q_{i,d}(t_\epsilon) - \dot{q}_{i,k}(0)t_\epsilon - q_{i,k}(0)$, $\varrho_2 = \dot{q}_{i,d}(t_\epsilon) - \dot{q}_{i,k}(0)$, $\varrho_3 = \ddot{q}_{i,d}(t_\epsilon)$. The definition of $W(q_k, \dot{q}_k, \dot{q}_{k,r}, \ddot{q}_{k,r})$ in (51) and (52) is given as follows:

$$W_r(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \dot{\boldsymbol{q}}_{k,r}, \ddot{\boldsymbol{q}}_{k,r}) = \begin{bmatrix} w_{r1} & w_{r2} & w_{r3} \\ w_{r4} & w_{r5} & w_{r6} \end{bmatrix}, \tag{55}$$

with

$$w_{r1} = \ddot{q}_{1,k,r} + g/l_1 \cos q_{2,k},$$
$$w_{r2} = \ddot{q}_{1,k,r} + \ddot{q}_{2,k,r},$$

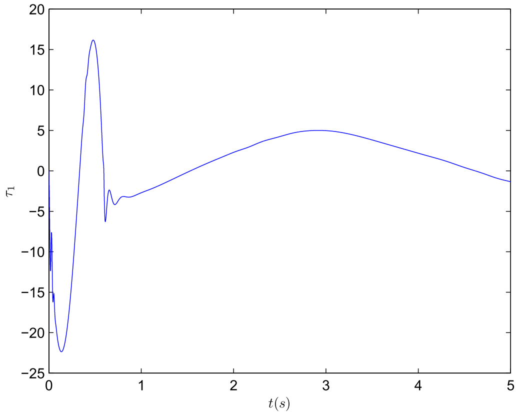


FIGURE 13. Torque input τ_1 (error-tracking ILC).

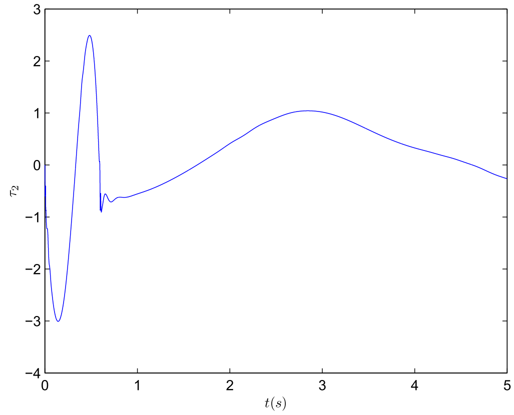


FIGURE 14. Torque input τ_2 (error-tracking ILC).

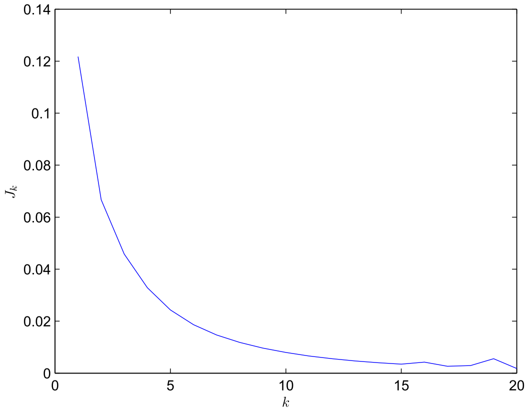


FIGURE 15. J_k (error-tracking ILC).

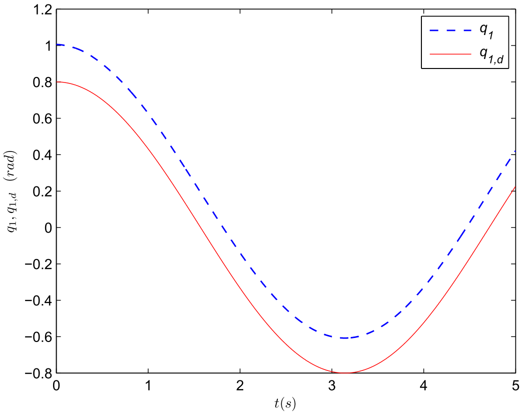


FIGURE 16. Position trajectory of joint 1 (Comparison A).

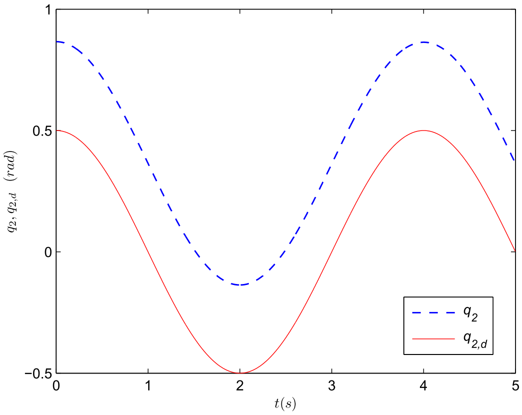


FIGURE 17. Position trajectory of joint 2 (Comparison A).

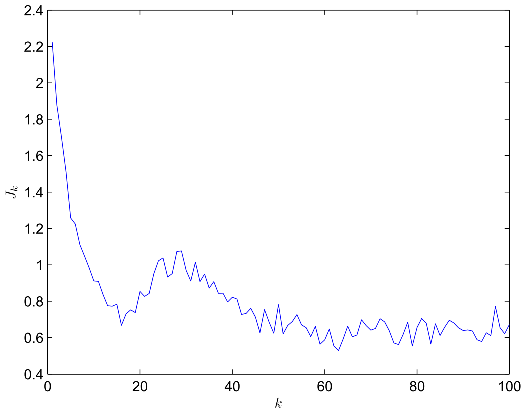


FIGURE 18. J_k (Comparison A).

$$\begin{aligned}
w_{r3} &= 2\ddot{q}_{1,k,r}\cos q_{2,k} + \ddot{q}_{2,k,r}\cos q_{2,k} - \dot{q}_{2,k}\dot{q}_{1,k,r}\sin q_{2,k}\\
&\quad -(\dot{q}_{1,k} + \dot{q}_{2,k})\dot{q}_{2,k,r}\sin q_{2,k} + g/l_1\cos(q_{1,k} + q_{2,k}),\\
w_{r4} &= 0,\\
w_{r5} &= w_{r2},\\
w_{r6} &= \dot{q}_{1,k}\dot{q}_{1,k,r}\sin q_{2,k} + \ddot{q}_{1,k,r}\cos q_{2,k} + g/l_1\cos(q_{1,k}\\
&\quad + q_{2,k}).
\end{aligned}$$

The control parameter and learning gains are set as $\gamma_4 = 5, \gamma_5 = 2$. The initial system state, t_ϵ, λ and l_1 are set as the same as above. The position trajectories at the 20th iteration are shown in Figs. 19-20. As shown in Fig.21, the tracking error does not converge to zero as the iteration number increases. From Figs. 19-21, we can see that the initial-rectification based adaptive ILC algorithm (51) is not suitable for the robotic control system with time-varying parameters.

Combining Fig. 15, Fig. 18 with Fig. 21, we can see that the proposed adaptive ILC scheme is effective in dealing with time-varying parameters and arbitrary initial errors for the robotic systems. The above simulation results verify the effectiveness of the proposed error-tracking adaptive robotic ILC scheme.

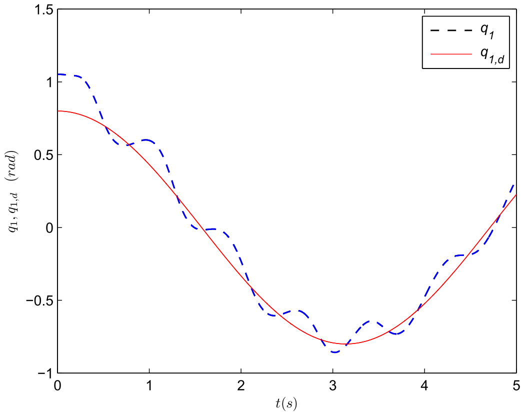


FIGURE 19. Position trajectory of joint 1 (Comparison B).

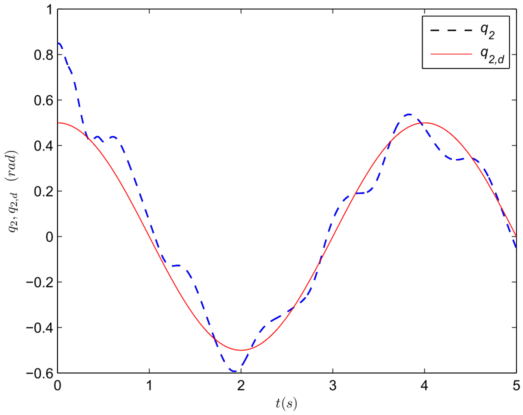


FIGURE 20. Position trajectory of joint 2 (Comparison B).

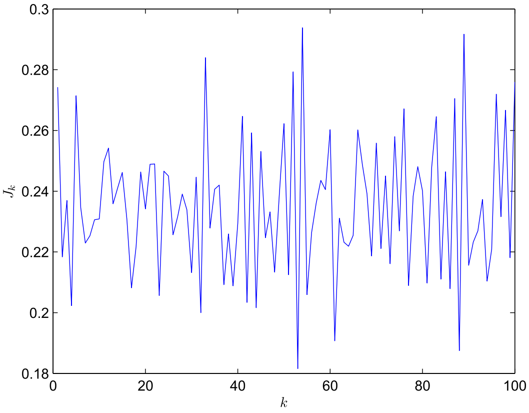


FIGURE 21. J_k (Comparison B).

Remark 5: If the external disturbances and unmodeled dynamics are considered, the above-discussed robotic systems may be described as

$$M(\boldsymbol{q}_k, \boldsymbol{\phi}(t))\ddot{\boldsymbol{q}}_k + C(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \boldsymbol{\phi}(t))\dot{\boldsymbol{q}}_k + F(\boldsymbol{q}_k, \dot{\boldsymbol{\phi}}(t))\dot{\boldsymbol{q}}_k + \boldsymbol{G}(\boldsymbol{q}_k, \boldsymbol{\phi}(t)) = \boldsymbol{\tau}_k + \boldsymbol{d}_k, \quad (56)$$

where $\boldsymbol{d}_k$ represents the sum of external disturbances and other unmodeled dynamics. By slightly modifying the origin control law and learning laws(13)-(15), the control algorithm for (56) may be achieved as

$$\boldsymbol{\tau}_k = -\gamma_1 \boldsymbol{s}_k + \Phi_k \boldsymbol{\vartheta}_k + W_k \boldsymbol{\eta}_k - \hat{\boldsymbol{d}}_k \mathrm{sat}_1(\frac{s_k}{\upsilon}), \quad (57)$$

$$(1-\mu)\dot{\boldsymbol{\vartheta}}_k = -\gamma_2 \Phi_k^T s_k + \mu(\boldsymbol{\vartheta}_{k-1} - \boldsymbol{\vartheta}_k), \quad (58)$$

$$\boldsymbol{\eta}_k = \mathrm{sat}_{\bar{\eta}}(\hat{\boldsymbol{\eta}}_{k-1}) - \gamma_3 W_k^T \boldsymbol{s}_k, \hat{\boldsymbol{\eta}}_{-1} = 0, \quad (59)$$

$$\hat{\boldsymbol{d}}_k = \mathrm{sat}_{\bar{d}}(\hat{\boldsymbol{d}}_{k-1}) + \gamma_4 |\boldsymbol{s}_k|, \hat{\boldsymbol{d}}_{-1} = 0, \quad (60)$$

where $\hat{\boldsymbol{d}}_k$ is used to estimate the upper bound of $\boldsymbol{d}_k$, υ is a small positive number, $\gamma_4 > 0$, and $\gamma_1, \gamma_2, \gamma_3$ and $\boldsymbol{\vartheta}_k$ are the same as that in (13), (14) and (15), respectively.

VII. CONCLUSION

In this paper, an error-tracking adaptive ILC scheme is proposed to solve the trajectory-tracking problem for robotic systems with time-varying parameters and arbitrary initial errors. Error-tracking strategy is applied to deal with the initial position problem of adaptive ILC. After proper parameterization to the dynamic model of robot manipulators, unknown time-invariant parameters and time-varying parameters are estimated according to combined learning law and difference learning law, respectively. It is rigorously proved that the proposed adaptive ILC law can achieve perfect tracking performance under arbitrary initial error condition. Numerical simulations are carried out to verify the effectiveness of our proposed robotic error-tracking adaptive ILC scheme.

REFERENCES

[1] D. Huang, W. Yang, T. Huang, N. Qin, Y. Chen, and Y. Tan, "Iterative learning operation control of high-speed trains with adhesion dynamics," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 6, pp. 1–11, Jan. 2021.

[2] T. Hu, K. H. Low, L. Shen, and X. Xu, "Effective phase tracking for bioinspired undulations of robotic fish models: A learning control approach," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 1, pp. 191–200, Feb. 2014.

[3] Y. Wang, D. Zhou, and F. Gao, "Iterative learning model predictive control for multi-phase batch processes," *J. Process Control*, vol. 18, no. 6, pp. 543–577, Jul. 2008.

[4] D. Shen, "Iterative learning control with incomplete information: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 5, pp. 885–901, Jul. 2018.

[5] D. Meng, "Convergence conditions for solving robust iterative learning control problems under nonrepetitive model uncertainties," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1908–1919, Jun. 2019.

[6] X. Dai, S. Tian, Y. Peng, and W. Luo, "Closed-loop P-type iterative learning control of uncertain linear distributed parameter systems," *IEEE/CAA J. Automatica Sinica*, vol. 1, no. 3, pp. 267–273, Jul. 2014.

[7] R. Chi, X. Liu, R. Zhang, Z. Hou, and B. Huang, "Constrained data-driven optimal iterative learning control," *J. Process Control*, vol. 55, pp. 10–29, Jul. 2017.

[8] Y.-Li, Y.-Q. Chen, and H.-S. Ahn, "Convergence analysis of fractional-order iterative learning control," *Control Theory Appl.*, vol. 29, no. 8, pp. 1027–1031, Aug. 2012.

[9] X. Bu, Q. Yu, Z. Hou, and W. Qian, "Model free adaptive iterative learning consensus tracking control for a class of nonlinear multiagent systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 4, pp. 677–686, Apr. 2019.

[10] L. Zhang and S. Liu, "Basis function based adaptive iterative learning control for non-minimum phase systems," *Acta Automatica Sinica*, vol. 40, no. 12, pp. 2716–2725, Dec. 2014.

[11] J. Liu, X. Ruan, and Y. Zheng, "Iterative learning control for discrete-time systems with full learnability," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 21, 2020, doi: 10.1109/TNNLS.2020.3028388.

[12] J. Li and J. Li, "Distributed adaptive fuzzy iterative learning control of coordination problems for higher order multi-agent systems," *Int. J. Syst. Sci.*, vol. 47, no. 10, pp. 2318–2329, Feb. 2015.

[13] Q. Z. Yan, X. B. Liu, S. Zhu, and J. P. Cai, "Suboptimal learning control for nonparametric systems with uncertain input gains," *Acta Automatica Sinica*, vol. 46, no. 5, pp. 1051–1060, Sep. 2020.

[14] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 3, pp. 49–59, 1987.

[15] M. Van, M. Mavrovouniotis, and S. Ge, "An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 7, pp. 1448–1458, Jul. 2019.

[16] Y.-J. Liu, S. Lu, and S. Tong, "Neural network controller design for an uncertain robot with time-varying output constraint," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2060–2068, Aug. 2017.

[17] W. He and Y. Dong, "Adaptive fuzzy neural network control for a constrained robot using impedance learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1174–1186, Apr. 2018.

[18] C. Sun, W. He, and J. Hong, "Neural network control of a flexible robotic manipulator using the lumped spring-mass model," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 1863–1874, Aug. 2016.

[19] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 620–629, 2016.

[20] Z. Liu, C. Chen, and Y. Zhang, "Decentralized robust fuzzy adaptive control of humanoid robot manipulation with unknown actuator backlash," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 3, pp. 605–616, Jun. 2015.

[21] M. R. Soltanpour, P. Otadolajam, and M. H. Khooban, "Robust control strategy for electrically driven robot manipulators: Adaptive fuzzy sliding mode," *IET Sci., Meas. Technol.*, vol. 9, no. 3, pp. 322–334, May 2015.

[22] P. R. Pagilla, B. Yu, and K. L. Pau, "Adaptive control of time-varying mechanical systems: Analysis and experiments," *IEEE/ASME Trans. Mechatronics*, vol. 5, no. 4, pp. 410–418, Dec. 2000.

[23] P. R. Pagilla and Y. Zhu, "Adaptive control of mechanical systems with time-varying parameters and disturbances," *J. Dynamic Syst., Meas., Control*, vol. 126, no. 3, pp. 520–530, 2004.

[24] Y. D. Song and R. H. Middleton, "Dealing with time-varying parameter problem of robot manipulators performing path tracking tasks," *IEEE Trans. Autom. Control*, vol. 37, no. 10, pp. 1597–1601, Oct. 1992.

[25] S. H. Hsu and L. C. Fu, "Globally adaptive decentralized control of time-varying robot manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, Nov. 2003, pp. 1458–1463.

[26] J.-X. Xu and R. Yan, "On initial conditions in iterative learning control," *IEEE Trans. Autom. Control*, vol. 50, no. 9, pp. 1349–1354, Sep. 2005.

[27] J.-X. Xu, X. Jin, and D. Huang, "Composite energy function-based iterative learning control for systems with nonparametric uncertainties," *Int. J. Adapt. Control Signal Process.*, vol. 28, no. 1, pp. 1–13, 2014.

[28] X. Jia and Z. Yuan, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 120–1195, Jul. 2004.

[29] C.-J. Chien and A. Tayebi, "Further results on adaptive iterative learning control of robot manipulators," *Automatica*, vol. 44, no. 3, pp. 830–837, Mar. 2008.

[30] F. Cao and J. Liu, "An adaptive iterative learning algorithm for boundary control of a coupled ODE–PDE two-link rigid–flexible manipulator," *J. Franklin Inst.*, vol. 354, no. 1, pp. 277–297, Jan. 2017.

[31] X. Li, Y.-H. Liu, and H. Yu, "Iterative learning impedance control for rehabilitation robots driven by series elastic actuators," *Automatica*, vol. 90, pp. 1–7, Apr. 2018.

[32] X. Jin, "Iterative learning control for non-repetitive trajectory tracking of robot manipulators with joint position constraints and actuator faults," *Int. J. Adapt. Control Signal Process.*, vol. 31, no. 6, pp. 859–875, Jun. 2017.

[33] L. Wu, Q. Yan, and J. Cai, "Neural network-based adaptive learning control for robot manipulators with arbitrary initial errors," *IEEE Access*, vol. 7, pp. 180194–180204, 2019, doi: 10.1109/TFUZZ.2004.834806.

[34] X. Jin and J.-X. Xu, "A barrier composite energy function approach for robot manipulators under alignment condition with position constraints," *Int. J. Robust Nonlinear Control*, vol. 24, no. 17, pp. 2840–2851, Nov. 2014.

[35] X.-D. Li, M.-M. Lv, and J. K. L. Ho, "Adaptive ILC algorithms of nonlinear continuous systems with non-parametric uncertainties for non-repetitive trajectory tracking," *Int. J. Syst. Sci.*, vol. 47, no. 10, pp. 2279–2289, Jul. 2016.

[36] Q. Z. Yan, M. X. Sun, and J. P. Cai, "Reference-signal rectifying method of iterative learning control," *Acta Autom. Sinica*, vol. 43, no. 8, pp. 1470–1477, Aug. 2017.

[37] C.-J. Chien, C.-T. Hsu, and C.-Y. Yao, "Fuzzy system-based adaptive iterative learning control for nonlinear plants with initial state errors," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 5, pp. 724–732, Oct. 2004.

[38] M.-X. Sun, X.-X. He, and B.-Y. Chen, "Repetitive learning control for time-varying robotic systems: A hybrid learning scheme," *Acta Automatica Sinica*, vol. 33, no. 11, pp. 1189–1195, Nov. 2007.

[39] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, Apr. 2009.

[40] L. Liu, Y. J. Liu, S. C. Tong, and C. L. P. Chen, "Integral barrier Lyapunov function-based adaptive control for switched nonlinear systems," *Sci. China Inf. Sci.*, vol. 63, no. 3, pp. 1–14, 2020.

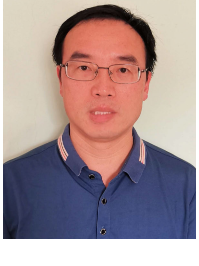
[41] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, no. 2, pp. 123–140, 1984.

[42] Q. Yan, J. Cai, Y. Ma, and Y. Yu, "Robust learning control for robot manipulators with random initial errors and iteration-varying reference trajectories," *IEEE Access*, vol. 7, pp. 32628–32643, 2019.

QIUZHEN YAN received the M.S. degree in computer science and the Ph.D. degree in control science and engineering from the Zhejiang University of Technology, Hangzhou, China, in 2005 and 2015, respectively. Since 2005, he has been with the College of Information Engineering, Zhejiang University of Water Resources and Electric Power, where he is currently a Lecturer. His current research interests include iterative learning control and repetitive control. He is a Senior Member of the Chinese Association of Automation.

JIANPING CAI was born in 1975. He received the Ph.D. degree from Zhejiang University, in 2014. He is currently an Associate Professor with the Zhejiang University of Water Resources and Electric Power. His main research interests include nonlinear systems and adaptive control.

YUNTAO ZHANG received the B.S. and M.S. degrees in computer science from the Zhejiang University of Technology, Hangzhou, in 2002 and 2005, respectively. Since 2005, he has been with the College of Information Engineering, Zhejiang University of Water Resources and Electric Power, where he is currently an Associate Professor. His current research interests include adaptive control and computer control algorithms.



ZHI YANG received the B.S. degree in computer application technology from Zhejiang Sci-Tech University, in 2000, and the M.S. degree in software engineering from Hangzhou Dianzi University, in 2017. He is currently an Associate Professor with the College of Information Engineering, Zhejiang University of Water Resources and Electric Power. His current research interests include computer assisted design (CAD) and iterative learning control.