Research Article

Trajectory Tracking Control for Uncertain Robot Manipulators with Repetitive Motions in Task Space

Jiutai Liu, Xiucheng Dong, Yong Yang, and Hongyu Chen

School of Electrical Engineering and Electronic Information, Xihua University, Chengdu 610039, China

Correspondence should be addressed to Xiucheng Dong; dxc136@163.com

Received 23 September 2020; Revised 12 December 2020; Accepted 17 December 2020; Published 7 January 2021

Academic Editor: Luigi Rodino

Copyright © 2021 Jiutai Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper aims at the trajectory tracking problem of robot manipulators performing repetitive tasks in task space. Two control schemes are presented to conduct trajectory tracking tasks under uncertain conditions including unmodeled dynamics of robot and additional disturbances. The first controller, pure adaptive iterative learning control (AILC), is based upon the use of a proportional-derivative-like (PD-like) feedback structure, and its design seems very simple in the sense that the only requirement on the learning gain and control parameters is the positive definiteness condition. The second controller is designed with a combination of AILC and neural networks (NNs) where the AILC is adopted to learn the periodic uncertainties that attribute to the repetitive motion of robot manipulators while the add-on NNs are used to approximate and compensate all nonperiodic ones. Moreover, a combined error factor (CEF), which is composed of the weighted sum of tracking error and its derivative, is designed for network updating law to improve the learning speed as well as tracking accuracy of the system. Stabilities of the controllers and convergence are proved rigorously by a Lyapunov-like composite energy function. The simulations performed on two-link manipulator are provided to verify the effectiveness of the proposed controllers. The results of compared simulations illustrate that our proposed control schemes can significantly conduct trajectory tracking tasks.

1. Introduction

Iterative learning control (ILC) is a well-established control scheme for systems with repetitive tasks during a finite time interval. The fundamental idea of this method is to iteratively construct a control input based on errors from previous control information so that the performance of the system such as trajectory tracking and disturbance attenuation can be optimized. ILC has been widely adopted in industries and manufacturing for control of repetitive tasks, such as train operation systems [1], subway trains [2], robotic fish systems [3], multiagent systems [4], and so forth. On the one hand, it is clear to see that plants are becoming various and complex. On the other hand, the requirement for ILC algorithms to deal with high productivity and high accurate tasks is increasing. Thus, the nonlinear dynamics which include vibration of flexible links, inertia effect of machine bodies, viscous, static friction forces, and so forth, can no longer be overlooked. Besides, there are disturbances and various uncertainties existing when modeling a complex nonlinear system, which makes our control problem complex and challenging. The implementation of ILC generally has a requirement of strict repetitiveness of the control system, but in practice, this is not always the case due to unavoidable noise/unmodeled disturbances and uncertainties. In recent years, with the great efforts of researchers in ILC filed, one of the significant achievements is the so-called adaptive ILC (AILC). Some great efforts in the field of AILC have been reported in [5–10]. The main idea behind the AILC approach is to approximate the disturbances and various uncertainties of the system to generate current input that can improve the output trajectory tracking performance from trial to trial.

AILC is a kind of control scheme which has great advantages in handling large parameter uncertainties of systems and sometimes no prior parameter information is required. There are many types of AILC algorithms that have been widely studied in recent years. For highly unknown uncertainties of the system, various methods of
approximation are proposed. It is worthy of noticing that the neural networks (NNs) and fuzzy system controllers have become very popular and efficient methods to deal with unknown system nonlinearities and approximate various uncertainties as well as disturbances. The researchers have devoted great efforts to improve the AILC schemes by combining neural networks with AILC schemes, and a variety of control algorithms have been proposed for nonlinear systems [11–15]. The authors employ the fuzzy system-based AILC schemes for nonlinear systems due to good approximation of fuzzy neural networks [16]. The works [16–18] take good use of fuzzy neural systems to compensate the unknown uncertainties of system. Wavelet networks are employed to deal with the nonlinear uncertainties, disturbances, and unlearnable dynamics of system [19–21]. These studies illustrate the superiority and feasibility of neural networks or fuzzy systems designed for AILC schemes in addressing unknown system nonlinearities as well as various uncertainties. Compared with the majority of AILC schemes, NN-based AILC algorithms have a remarkable improvement and much better performance in the speed of learning as well as error convergence.

Robot manipulators [5, 22–25], in recent years, have become increasingly vital in industrial manufacture, which draw much attention of the researchers to make great efforts in their controller design. Varieties of schemes have been designed to achieve accurate trajectory tracking and great control performance. The conventional techniques, such as impedance control, sliding mode control, adaptive control, and robust control, are employed to control the robot manipulators. However, these classical approaches are subject to an appropriate mathematical model, priori knowledge of system dynamics, and known system parameters. For the highly coupled nonlinear system with various uncertainties and disturbances, the current trend of control is to focus on making enormous efforts to combine intelligent schemes, such as neural networks and fuzzy systems [26], with classical control approaches to overcome the various defects of conventional techniques. There are some great improvements reported in recent years [22–24, 27–32]. The authors present a NN controller with full-state feedback to control the uncertain constrained robot manipulator with flexible joints (RMFJs) and successfully make varieties of experiments on Baxter [23]. A neural network-based adaptive iterative learning control scheme is proposed for robot manipulators with arbitrary initial errors, where the AILC scheme is adopted to address the initial problem for robot ILC schemes and adaptive learning neural networks are used to estimate uncertainties in the system [24]. Moreover, many robot pose trajectory tracking controllers require robot full-state measurements, such as position, velocity, and acceleration in the joint and task space. However, it is very difficult to obtain the real-time position, velocity, and acceleration measurements of robot manipulators in practical situation for various disturbances, unmodeled dynamics, and unknown uncertainties. Motivated by this situation, the authors provide a new controller for the pose trajectory tracking control problem which is based on a primary joint loop of joint velocity and a secondary loop of operational space position control [33]. Recent study [34] presents a task space finite-time control of robotic manipulators subject to uncertain kinematic and dynamic equations, unbounded disturbances, and unknown payload. To estimate various disturbances and uncertainties of systems, the work [35] presents a new nonsingular fast terminal sliding-mode control approach for the finite-time control based on disturbance observer.

In this paper, the proposed control scheme is further extended to the task space or the so-called Cartesian space [36]. The purpose of this paper is to address the tracking control of uncertain robot manipulators with repetitive tasks, where two controllers are considered in consequence. Concerning the characteristics of repetitive tasks when the robot manipulator is working, it is rational to consider the uncertainties as two parts, periodic part and nonperiodic part. The periodic part mainly attributes to the repetitive motions of robot manipulator, while the nonperiodic part is mainly caused by unmodeled dynamics, unknown parameters, and additional disturbances, such as the static friction forces with joints. The first controller is pure adaptive iterative learning control (AILC) built around a classical PD-like feedback structure, for which an iterative term is added to address the unknown parameters and additional disturbances, while the second controller is a combination of AILC and neural networks (NNs) where the AILC is adopted to iteratively learn those involved periodic uncertainties purposely and the NNs are used to approximate and compensate all nonperiodic ones. It is worth noticing that a combined error factor (CEF), which consists of the weighted sum of tracking error and its derivative, is designed for NN updating law to improve the learning speed as well as tracking accuracy of the system. Stabilities of two proposed controllers are proved rigorously by a Lyapunov-like composite energy function. With the help of NNs and CEF, it is clear to see that the proposed control approach has achieved a significant control effect with remarkable transient performance. These have been verified by compared simulations.

The main contributions of this paper can be summarized as follows:

1. A novel control scheme which combines AILC and NNs is proposed for robot manipulators performing repetitive tasks in the task space, where the repetitiveness of system motion is fully considered.
2. The stability and convergence of proposed control scheme are provided rigorously by a Lyapunov-like composite energy function, where the CEF, which is composed of the weighted sum of tracking error and its derivative, is employed to be designed for network updating law to improve the learning speed as well as tracking accuracy of the system.
3. The compared simulation results illustrate that the combination of AILC and NNs is superior to AILC in learning speed and tracking accuracy of robot manipulators.

The remainder of this paper is organized as follows. In Section 2, preliminaries and problem formulation including useful properties and assumptions are given. In Section 3,
two controllers are designed for the problem formulation. The verification process of convergence and stability analysis is provided by a Lyapunov-like composite energy function in Section 4. In Section 5, the simulation of the proposed two control schemes is achieved on a 2-degree-of-freedom (2DOF) manipulator and there are some important results. The conclusions and future recommendations are illustrated in the last section.

2. Preliminaries and Problem Formulation

2.1. Kinematics and Dynamics. The forward kinematics of the robot is described as

\[ x(t) = \phi(q), \]

where \( x(t), q \in \mathbb{R}^n \), and \( n \) are the positions of robot and end effector in Cartesian space, joint coordinates, and the degree of the robotic freedom, respectively, differentiating the forward kinematics relation \( x = \phi(q) \) with respect to time.

\[ \dot{x} = J(q)\dot{q}, \quad \ddot{x} = J(q)\ddot{q} + \dot{J}(q)q, \quad J(q) = \frac{\partial \phi}{\partial q}, \]

where \( J(q) \in \mathbb{R}^{n \times n} \) is the Jacobian matrix denoting the relation of velocity between the Cartesian space and joint coordinate.

The dynamics of the \( n \)-link robot can be given by

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + D(t) + d, \]

where \( q, \dot{q}, \text{and} \ddot{q} \) belong to \( n \)-dimensional vectors and are the robot joint angle, velocity, and acceleration; \( M(q) \in \mathbb{R}^{n \times n} \) is the (symmetric, positive definite) inertia matrix; \( C(q, \dot{q}) \) is the Coriolis/centrifugal matrix; \( G(q) \in \mathbb{R}^{n \times n} \) represents the gravitational torque; \( \tau \in \mathbb{R}^n \) is the vector of control torque applied on the robot joint; \( D(t) \in \mathbb{R}^n \) is the vector containing nonperiodic uncertainties and friction in the robot; and \( d \in \mathbb{R}^n \) is external disturbance in the environment.

In this paper, the trajectory tracking tasks are conducted in the Cartesian space; we have to transfer the dynamics (3) to the task space. To do so, by considering the kinematics (1) and (2) as well as dynamics (3), the manipulator dynamics in the Cartesian space can be given as follows:

\[ M_k(q)\ddot{x}(t) + C_k(q, \dot{q})\dot{x}(t) + G_k(q) = \tau_x + D_k(t) + d_k, \]

where

\[ M_k(q) = J^{-T}(q)M(q)J^{-1}(q), \]
\[ C_k(q, \dot{q}) = J^{-T}(q)(C(q, \dot{q}) - M(q)\dot{J}(q))J^{-1}(q), \]
\[ G_k(q) = J^{-T}(q)G(q), \]
\[ D_k(t) = J^{-T}(q)D(t), \]
\[ d_k = J^{-T}(q)d, \]
\[ \tau_x = J^{-T}(q)\tau. \]

We define \( J^{-T}(q) \) as the inverse matrix of the transposed \( J(q) \).

\[ J^{-T}(q) = (J^T(q))^{-1}. \] (6)

2.2. Problem Formulation. Considering the inaccurate modeling of robot models, the dynamics (4) will be described as follows:

\[ (\bar{M}_k(q) + \Delta M_k(q))\ddot{x}_k(t) + (\bar{C}_k(q, \dot{q}) + \Delta C_k(q, \dot{q}))\dot{x}_k(t) + (\bar{G}_k(q) + \Delta G_k(q)) = \tau_x + D_k(t) + d_k, \]

where \( \Delta M_k(q), \Delta C_k(q, \dot{q}), \Delta G_k(q) \) represent the unmodeled parts of \( \bar{M}_k(q), \bar{C}_k(q, \dot{q}), \) and \( \bar{G}_k(q) \). When considering the controller design for (7), \( \Delta M_k(q), \Delta C_k(q, \dot{q}), \Delta G_k(q) \), \( D_k(t) \), and \( d_k \) will be regarded as uncertainties of the robot manipulator system. In this paper, \( \tau_x \) is the control input to be designed later. We make the following reasonable assumptions [25].

Assumption 1. The reference trajectory and its first and second time derivative, namely, \( x_k(0), \dot{x}_k(0), \ddot{x}_k(0) \), are given.

Assumption 2. The resetting initial condition is satisfied:

\[ x_k(0) - x_d(0) = \dot{x}_k(0) - \dot{x}_d(0) = 0, \quad \forall k \in \mathbb{Z}. \]

We will also need the following properties, which are common to robot manipulators [36].

Property 1. The matrix \( \bar{M}_k(q) - 2\bar{C}_k(q, \dot{q}) \) is skew-symmetric [37]:

\[ x^T[\bar{M}_k(q) - 2\bar{C}_k(q, \dot{q})]x = 0, \quad \forall q, \dot{q}, x \in \mathbb{R}^n. \] (9)

Property 2. The inertia matrix \( \bar{M}_k(q) \) is symmetric and positive-definite, which is further bounded:

\[ \lambda_2\|x\|^2 < x^T\bar{M}_k(q)x < \lambda_1\|x\|^2. \] (10)

Property 3

\[ \|D_k(t) - f\| < \beta\|d_k\| < k_2\|\dot{x}_k(t)\| < \delta, \] (11)

where \( k_2 > 0 \) is a known positive constant, \( f \) represents the dynamics uncertainties of robot manipulator model and will be introduced in (18), \( \beta \) is a known positive parameter, and \( \delta > 0 \).
2.3. Preliminaries. To design an appropriate estimation and compensation algorithm, Gaussian radial basis function neural networks (RBFNNs) are an alternative method; by virtue of the simple topological structure and universal approximation ability, RBFNNs have been widely applied for nonlinear system modeling and control [38–46]. In this paper, RBFNNs are used to estimate and compensate for all the nonperiodic uncertainties of the robot manipulator system. A RBFNN is a particular network architecture which uses \( p \) numbers of Gaussian function of the form expressed as follows [40]:

\[
\phi_j(u) = \exp\left(-\frac{\|u - c_j\|^2}{2b_j^2}\right), \quad j = 1, 2, \ldots, p,
\]

\[
y_i = \sum_{j=1}^{p} w_{ij}\phi_j, \quad i = 1, 2, \ldots, k,
\]

where \( c_j \in \mathbb{R}^n \) is the center vector and \( b_j \in \mathbb{R} \) is the width of the \( j \)-th kernel. \( u \in \mathbb{R}^n \) is the input of NN, \( \phi_j \in \mathbb{R} \) is the output of the hidden layer, and \( y_i \in \mathbb{R} \) is the \( i \)-th output of the NN. Besides, \( w_{ij} \in \mathbb{R} \) is the weight for the \( j \)-th hidden layer to the \( i \)-th output of the NN. Let

\[
\begin{bmatrix}
w_{11} & w_{21} & \cdots & w_{k1} \\
 w_{12} & w_{22} & \cdots & w_{k2} \\
 \vdots & \vdots & \ddots & \vdots \\
 w_{1p} & w_{2p} & \cdots & w_{kp}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\phi_1(u) \\
\phi_2(u) \\
\vdots \\
\phi_p(u)
\end{bmatrix}, \quad
\begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_p
\end{bmatrix}
\]

The approximation of the NN can be described as follows:

\[
Y = \hat{W}^T\varphi(u).
\]

When the uncertainties of the system to be approximated, denoted by \( \Lambda(u) : \mathbb{R}^n \rightarrow \mathbb{R}^k \), are piecewise continuous, the following is assumed to be true unconditionally.

Assumption 3 (see [47]). Assume that there is a parametric matrix \( W^* \in \mathbb{R}^{p\times k} \), known as the optimal approximation parameter; therefore, \( W^*\varphi(u) \) can estimate the system uncertainties as close as possible, that is, given an arbitrary small positive constant \( \varepsilon_N \), there exists a ideal weight matrix \( W^* \); as a result, the approximation error \( \varepsilon(u) \triangleq W^*\varphi(u) - \Lambda(u) : \mathbb{R}^n \rightarrow \mathbb{R}^k \) satisfies

\[
\|\varepsilon(u)\| = \|W^*\varphi(u) - \Lambda(u)\| < \varepsilon_N.
\]

3. Controller Design

In this section, two control schemes will be given to address the control problem of the robot manipulator system. The first controller, pure AILC, is based upon the use a PD-like feedback structure, for which an iterative term is added to deal with the unknown parameters and additional disturbances. The second controller is designed with a combination of AILC and NNs where the AILC is adopted to iteratively learn the periodic uncertainties while the NNs are used to approximate and compensate all of the system nonperiodic uncertainties.

3.1. AILC Design. Now, we are in the position of deriving the tracking error dynamics of the robot. Take the control input \( \tau_x \) as follows:

\[
\tau_x = \dot{M}_s(q)\ddot{x}_d(t) + \ddot{C}_s(q, \dot{q})\dot{x}_d(t) + \ddot{G}_s(q) + \tau_k,
\]

where \( \tau_k \) is the virtual input to be designed later. Substituting (16) into (7), we can get the following form:

\[
\dot{M}_s(q)\ddot{e}_k(t) + \ddot{C}_s(q, \dot{q})\dot{e}_k(t) + f = D_x(t) + d_x + \tau_k.
\]

where

\[
f = \Delta M_s\ddot{x}_k(t) + \Delta C_s\dot{x}_k(t) + \Delta G_s,
\]

\[
e_k(t) = x_k(t) - x_d(t),
\]

\[
\dot{e}_k(t) = \dot{x}_k(t) - \dot{x}_d(t),
\]

where \( f \) represents the dynamic uncertainties of the robot model.

Considering system (17) with Properties 1–3, the AILC controller is designed as follows:

\[
\tau_k = \frac{k_1}{k_2}\dot{M}_s(q)e_k(t) - \frac{k_1}{k_2}\dot{C}_s(q, \dot{q})e_k(t) - k_3\delta(t)\text{sgn}(e_k(t)),
\]

with

\[
\delta(t) = \delta_{k-1}(t) + \Gamma\varepsilon^T(t)\text{sgn}(e_k(t)),
\]

where \( \delta_{k-1}(t) = 0, k_1, k_3 > 0 \) from (21) are the weight factors of the combined error, \( \Gamma \) is a positive scalar, \( \text{sgn}(e_k(t)) \) is the vector obtained by applying the signum function to all elements of \( e_k \), \( \delta \) is used to describe the uncertainties of robot manipulators, and \( \delta \) is used to approximate \( \delta \) adaptively.

3.2. AILC Design with NN. In this part, we will use RBFNNs to approximate and compensate all of the system uncertainties, external disturbances, and nonperiodic ones in controller design.

In order to improve the learning speed as well as tracking accuracy of the system, a CEF is applied. Moreover, it will be used to design for the updating law of NNs and stability analysis of the system later.

\[
E_k(t) = k_1e_k(t) + k_2\dot{e}_k(t),
\]

where \( k_1, k_2 > 0 \) are the weight factors of the combined error.

Firstly, we have to put the external disturbances \( d_x \), nonperiodic uncertainties and friction \( D_x(t) \), and the dynamic uncertainties of robot model \( f \) together in order to be
compensated and approximated by RBFNNs. According to (17), we have
\[ \mathbf{M}_x(q)\ddot{e}_k(t) + \mathbf{C}_x(q, \dot{q})\dot{e}_k(t) = \mathbf{F} + \mathbf{r}_k, \]
(22)
\[ \mathbf{F} = -\mathbf{f} + \mathbf{D}_x(t) + \mathbf{d}_x. \]
(23)

The new proposed controller that combines AILC and NNs is given as follows:
\[ \mathbf{r}_k = \frac{k_1}{k_2}\mathbf{M}_x(q)\dot{e}_k(t) - \frac{k_1}{k_2}\mathbf{C}_x(q, \dot{q})e_k(t) \]
\[ - \hat{\mathbf{W}}^T \varphi(u) - \hat{\delta}_k(t)\text{sgn}(\dot{e}_k(t)), \]
(24)
with
\[ \hat{\delta}_k(t) = \tilde{\delta}_k(t) + \Gamma \dot{e}_k^T(t)\text{sgn}(\dot{e}_k(t)), \]
(25)
where \( u = [e^T, \dot{e}^T]^T \) is the input signal for RBF neural network, while \( \mathbf{F} \) is the output of the network. In (24), \( \hat{\mathbf{W}}^T \varphi(u) \) is employed to estimate all nonperiodic uncertainties and external disturbances in the environment and \( \tilde{\delta}_k(t) \) is employed to adaptively approximate the periodic uncertainties of the robot manipulator system. Besides, \( \delta > \varepsilon_N \).

\[ \varepsilon(u) \triangleq \mathbf{F}(u) - \hat{\mathbf{F}}(u, \mathbf{W}^*), \]
(26)
\[ \hat{\mathbf{F}}(u, \mathbf{W}^*) = \mathbf{W}^{*T}\varphi(u), \]
(27)
where \( \mathbf{W}^{*T} \) represents the ideal RBFNN weights and \( \varepsilon(u) \) is the approximation error of the RBFNN. The network updating law is given by
\[ \hat{\mathbf{W}} = \varphi(u)E_k^T(t), \]
(28)
where \( \varphi \) is a prespecified positive-definite diagonal matrix.

4. Convergence Analysis

In this part, there are mainly two parts for the convergence and stability analysis of the system. First of all, we have to prove that the initial state has bounded convergence, and Lyapunov-like composite energy function is designed according to the system mathematical model. We need to prove the continuity and boundedness of Lyapunov-like composite energy function.

**Theorem 1.** Let Assumptions 1 and 2 be satisfied. If matrices \( K_p = (k_1/k_2)\mathbf{C}_x(q, \dot{q}) \in \mathbb{R}^{n \times n} \), \( K_p = (k_1/k_2)\mathbf{M}_x(q) \in \mathbb{R}^{n \times n} \) are symmetric positive definite and \( \Gamma \) is a positive scalar, then \( e_k(t), \dot{e}_k(t), r_k, \forall t \in [0, T] \), and \( e_k \in \mathbb{Z} \) are bounded.

\[ \lim_{k \to -\infty} e_k(t) = \lim_{k \to -\infty} \dot{e}_k(t) = 0. \]
(29)

**Proof.** The proof consists of three parts. Part 1 derives the difference of the Lyapunov functional \( N_k(t) \). Part 2 proves that the Lyapunov functional \( N_k(t) \) is bounded for \( k = 0 \). There are tracking performance and the boundedness of all signals in the closed-loop system shown in Part 3. \( \square \)

**Part 1. Difference of \( N_k(t) \).**

Consider the Lyapunov-like composite energy function as follows:
\[ N_k(t) = V_k(t) + \frac{1}{2} \int_0^t \Gamma^{-1} \delta_k^2(\eta)d\eta, \]
(30)
\[ V_k(t) = V_1(t) + V_2(t), \]
(31)
\[ V_1(t) = \frac{1}{2} \dot{e}_k^T(t)\mathbf{M}_x(q)\dot{e}_k(t), \]
(32)
\[ V_2(t) = \frac{1}{2} \mathbf{e}_k^T(t)K_p\mathbf{e}_k(t), \]
(33)
where \( \delta_k(\eta) = \delta - \tilde{\delta}_k(\eta), \delta = \beta + k_d \). Then, the time derivative of \( V_k(t) \) is
\[ \dot{V}_k(t) = \dot{V}_1(t) + \dot{V}_2(t), \]
(34)
\[ \dot{V}_1(t) = \dot{e}_k^T(t)\mathbf{M}_x(q)\dot{e}_k(t) + \frac{1}{2} \dot{e}_k^T(t)\mathbf{M}_x(q)\dot{e}_k(t), \]
\[ \dot{V}_2(t) = \dot{e}_k^T(t)K_p\mathbf{e}_k(t), \]
(35)

According to (30), we have
\[ \Delta N_k(t) = N_k(t) - N_{k-1}(t) \]
\[ = V_k(t) - V_{k-1}(t) + \frac{1}{2} \int_0^t \Gamma^{-1} \left( \dot{\delta}_k^2(\eta) - \tilde{\delta}_{k-1}^2(\eta) \right)d\eta, \]
\[ = V_k(t) - V_{k-1}(t) - \frac{1}{2} \int_0^t \Gamma^{-1} \left( \dot{\delta}_k^2(\eta) + 2\tilde{\delta}_k(\eta)\tilde{\delta}_k(\eta) \right)d\eta, \]
(36)

Finding the first derivative of \( V_k(t) \) and then integrating the two sides, we have
\[ V_k(t) = V_k(0) + \int_0^t \dot{V}_1(\eta) + \dot{V}_2(\eta)d\eta. \]
(37)

According to Assumption 2, \( V_k(0) = 0 \), and it is worthy of noticing (17); note that
\[ \mathbf{M}_x(q)\dot{e}_k(t) = \mathbf{D}_x(t) + \mathbf{r}_k - \mathbf{C}_x(q, \dot{q})\dot{e}_k(t) - \mathbf{f}, \]
(38)
\[ \dot{V}_k(t) = \dot{e}_k^T(t)\mathbf{M}_x(q)\dot{e}_k(t) + \frac{1}{2} \dot{e}_k^T(t)\mathbf{M}_x(q)\dot{e}_k(t) \]
\[ + \dot{e}_k^T(t)K_p\mathbf{e}_k(t). \]
(39)

Substituting (38) into (39) yields
\[ \dot{V}_k(t) = \frac{1}{2} \dot{e}_k^T(t) \left[ \mathbf{M}_x(q) - 2\mathbf{C}_x(q, \dot{q}) \right] \dot{e}_k(t) \]
\[ + \dot{e}_k^T(t) \left[ \mathbf{r}_k + \mathbf{D}_x(t) - \mathbf{f} + \mathbf{d}_x \right] + \dot{e}_k^T(t)K_p\mathbf{e}_k(t). \]
(40)
Then, according to Property 1, (40) can be simplified:
\[ \dot{V}_k(t) = \dot{\epsilon}_k^T(t) \left[ \mathbf{D}_k(t) - \mathbf{f} + \mathbf{d}_k \right] + \dot{\epsilon}_k^T(t) \mathbf{K}_p \mathbf{e}_k(t). \]  

(41)

Noticing the control law (19) and substituting (19) into (41), we have
\[ \dot{V}_k(t) = \dot{\epsilon}_k^T(t) \left[ \mathbf{D}_k(t) - \mathbf{f} + \mathbf{d}_k \right] - \dot{\epsilon}_k^T(t) \mathbf{K}_p \mathbf{\hat{e}}_k(t) \]
\[ - \dot{\epsilon}_k^T(t) \mathbf{\hat{d}}_k(t) \text{sgn}(\mathbf{\hat{e}}_k(t)). \]

Combining (42), (43), and (44) yields
\[ \dot{V}_k(t) \leq \dot{\epsilon}_k^T(t) \left( \beta + k_0 \right) \text{sgn}(\mathbf{\hat{e}}_k(t)) - \dot{\epsilon}_k^T(t) \mathbf{K}_p \mathbf{\hat{e}}_k(t) \]
\[ - \dot{\epsilon}_k^T(t) \mathbf{\hat{d}}_k(t) \text{sgn}(\mathbf{\hat{e}}_k(t)) \leq - \dot{\epsilon}_k^T(t) \mathbf{K}_p \mathbf{\hat{e}}_k(t) + \dot{\epsilon}_k^T(t) \mathbf{\hat{d}}_k(t) \text{sgn}(\mathbf{\hat{e}}_k(t)). \]

Integrating the two sides of (45), we have
\[ V_k(t) \leq - \int_0^t \dot{\epsilon}_k^T(\eta) \mathbf{K}_p \mathbf{\hat{e}}_k(\eta) d\eta + \int_0^t \dot{\epsilon}_k^T(\eta) \mathbf{\hat{d}}_k(\eta) \text{sgn}(\mathbf{\hat{e}}_k(\eta)) d\eta. \]

(46)

Subsequently, combination of (35), (36), and (46) leads to
\[ \Delta N_k(t) \leq - V_{k-1}(t) - \int_0^t \dot{\epsilon}_k^T(\eta) \mathbf{K}_p \mathbf{\hat{e}}_k(\eta) d\eta - \frac{1}{2} \int_0^t \Gamma^{-1} \dot{\delta}_k(\eta) d\eta. \]

(47)

According to (31), (32), and (33), we can easily get that \( V_{k-1}(t) \geq 0. \) Thus, \( \Delta N_k(t) \leq 0 \) and the Lyapunov function \( N_k(t) \) is a nonincreasing sequence.

Part 2. Finiteness of \( N_0(t) \).

The next setup is to prove the boundedness of \( N_0(t) \). Considering (28) and (46), we have
\[ N_0(t) \leq \int_0^t \dot{\epsilon}_0^T(\eta) \left[ \mathbf{\bar{\delta}}_0(\eta) \text{sgn}(\mathbf{\hat{e}}_0(\eta)) - \mathbf{K}_p \mathbf{\hat{e}}_0(\eta) \right] d\eta \]
\[ + \frac{1}{2} \int_0^t \Gamma^{-1} \dot{\delta}_0^2(\eta) d\eta. \]

(48)

Derivation of \( t \) on both sides of (48) yields
\[ \dot{N}_0(t) \leq \frac{1}{2} \Gamma^{-1} \dot{\delta}_0^2(t) + \dot{\epsilon}_0^T(t) \left[ \mathbf{\bar{\delta}}_0(t) \text{sgn}(\mathbf{\hat{e}}_0(t)) - \mathbf{K}_p \mathbf{\hat{e}}_0(t) \right]. \]

(49)

Noticing that \( \mathbf{\bar{\delta}}_0(t) = 0, \mathbf{\bar{\delta}}_0(t) - \bar{\delta}_0(t) = \Gamma \mathbf{\hat{e}}_0(t) \text{sgn}(\mathbf{\hat{e}}_0(t)) \), we have
\[ \dot{\epsilon}_0^T(t) \text{sgn}(\mathbf{\hat{e}}_0(t)) = \Gamma^{-1} \mathbf{\hat{\delta}}_0(t). \]

Substituting (50) into (49) yields
\[ \dot{N}_0(t) \leq - \dot{\epsilon}_0^T(t) \mathbf{K}_p \mathbf{\hat{e}}_0(t) + \left[ \mathbf{\bar{\delta}}_0(t) + \frac{1}{2} \mathbf{\hat{\delta}}_0(t) \right] \Gamma^{-1} \mathbf{\hat{\delta}}_0(t). \]

(51)

Moreover, \( \mathbf{\bar{\delta}}_0(t) = \delta - \bar{\delta}_0(t) \), and substituting it into (51) leads to
\[ \dot{N}_0(t) \leq - \dot{\epsilon}_0^T(t) \mathbf{K}_p \mathbf{\hat{e}}_0(t) + \delta \Gamma^{-1} \mathbf{\hat{\delta}}_0(t) - \frac{1}{2} \mathbf{\hat{\delta}}_0(t) \Gamma^{-1} \mathbf{\hat{\delta}}_0(t). \]

(52)

When \( \lambda > 0 \), using the Cauchy inequality, we have
\[ \delta \Gamma^{-1} \mathbf{\hat{\delta}}_0(t) \leq \lambda \left[ \Gamma^{-1} \mathbf{\hat{\delta}}_0(t) \right]^2 + \frac{1}{4\lambda} \delta^2. \]

(53)

Thus, (52) can be further simplified as follows:
\[ \dot{N}_0(t) \leq - \mathbf{\bar{\delta}}_0(t) \mathbf{\hat{e}}_0(t) + \lambda \left[ \Gamma^{-1} \mathbf{\hat{\delta}}_0(t) \right]^2 + \frac{1}{4\lambda} \delta^2 - \frac{1}{2} \Gamma^{-1} \mathbf{\hat{\delta}}_0^2(t). \]

(54)

Since the initial set value of the system is bounded, we can get that \( \mathbf{\bar{\delta}}_0(t) \) is bounded. (54) can be further simplified as follows:
\[ \dot{N}_0(t) \leq \lambda \left[ \Gamma^{-1} \mathbf{\hat{\delta}}_{0_{\text{max}}}(t) \right]^2 + \frac{1}{4\lambda} \delta_{\text{max}}^2, \]

(55)

where
\[ \delta_{\text{max}} = \sup_{t \in [0,T]} \mathbf{\hat{\delta}}_{0_{\text{max}}}(t) = \sup_{t \in [0,T]} \mathbf{\bar{\delta}}_0(t). \]

(56)

Therefore, \( N_0(t) \) is bounded and \( N_0(t), \forall t \in [0,T] \) is consistently continuous and bounded.

Part 3. Convergence of \( \mathbf{e}_k(t) \) and \( \mathbf{\hat{e}}_k(t) \).

In order to prove the convergence of error, we have
\[ N_k(t) = N_0(t) + \sum_{i=1}^{k} \Delta N_i(t). \]

(57)

According to (30), (31), (32), and (33), we have
\[ N_k(t) \leq N_0(t) + \sum_{i=1}^{k} V_{i-1}(t) \]
\[ \leq N_0(t) - \frac{1}{2} \sum_{i=1}^{k} \mathbf{e}_i^T(t) \mathbf{M}_k(q) \mathbf{e}_i(t) \]
\[ - \frac{1}{2} \sum_{i=1}^{k} \mathbf{e}_{i-1}^T(t) \mathbf{K}_p \mathbf{e}_{i-1}(t). \]

(58)

Thus, we have
δ
C
C
Ek

\[ \sum_{i=1}^{k} e_i^T(t)M_k(q)e_{i-1}(t) + \sum_{i=1}^{k} e_i^T(t)K_pe_{i-1}(t), \]  
\leq 2(N_0(t) - N_k(t)) \leq 2N_0(t). \]

Since \( N_k(t) \) is continuous and bounded, we have
\[ \lim_{k \to \infty} e_k(t) = \lim_{k \to \infty} \dot{e}_k(t) = 0, \quad t \in [0, T]. \]

Therefore, \( e_k(t) \) and \( \dot{e}_k(t) \) are bounded, and we can see from (19) that control law \( \tau_k \) is bounded. The proof of Theorem 1 has been completed.

**Theorem 2.** For the robot system dynamics (22) under Properties 1–3 and Assumptions 1–3, the CEF \( E_k(t) \) and \( W_k(t) \) will converge to zero asymptotically \( \forall t \in [0, T] \) and \( \forall k \in \mathbb{Z}_+ \), and all the signals of the closed-loop system are bounded.

\[ \lim_{k \to \infty} W_k(t) = \lim_{k \to \infty} E_k(t) = 0, \quad t \in [0, T]. \]

**Proof.** The proof consists of three parts. Part 4 derives the difference of the Lyapunov functional \( W_k(t) \). Part 5 proves that the Lyapunov functional \( W_k(t) \) is bounded for \( k = 0 \). There are tracking performance and the boundedness of all signals in the closed-loop system shown in Part 6.

**Part 4.** Difference of \( W_k(t) \).

Consider the Lyapunov-like composite energy function as follows:
\[ W_k(t) = V_k(t) + k_2^2 \int_0^T \Gamma^{-1}\delta_k^2(\eta)d\eta, \]
\[ V_k(t) = V_3(t) + V_4(t), \]
\[ V_3(t) = E_k^T(t)M_k(q)E_k(t), \]
\[ V_4(t) = k_2tr \left[ W_k^T(t)\psi^{-1}W_k(t) \right]. \]

where \( \delta_k(t) = \delta - \delta_k(t), \quad \tilde{W} = W^* - \tilde{W} \). Then, the time derivative of \( V_k(t) \) is
\[ \dot{V}_k(t) = \dot{V}_3(t) + \dot{V}_4(t), \]
\[ \dot{V}_3(t) = 2E_k^T(t)M_k(q)\dot{E}_k(t) + E_k^T(t)\dot{M}_k(q)E_k(t), \]
\[ \dot{V}_4(t) = 2k_2tr \left[ \tilde{W}_k^T\psi^{-1}\tilde{W}_k \right]. \]

According to (63), we have
\[ \Delta W_k(t) = W_k(t) - W_{k-1}(t) = V_k(t) - V_{k-1}(t) \]
\[ + k_2^2 \int_0^T \Gamma^{-1}(\delta_k^2(\eta) - \delta_k^2(\eta))d\eta \]
\[ = V_k(t) - V_{k-1}(t) - k_2^2 \int_0^T \Gamma^{-1}(\delta_k^2(\eta) + 2\delta_k(\eta)\delta_k(\eta))d\eta, \]
\[ \text{where} \quad \delta_k(t) = \delta_k(t) - \delta_k(t-1) = \Gamma \dot{e}_k^T(t)\text{sgn}(\dot{e}_k(t)). \]

Finding the first derivative of \( V_k(t) \) and then integrating the two sides, we have
\[ V_k(t) = V_k(0) + \int_0^t (\dot{V}_3(\eta) + \dot{V}_4(\eta))d\eta. \]

According to Assumption 2, \( V_k(0) = 0 \). In order to address \( \dot{V}_3(t) \), note that

\[ M_k(q)\dot{E}_k(t) = M_k(q)[k_1\dot{e}_k(t) + k_2\dot{\delta}_k(t)] \]
\[ = k_1M_k(q)\dot{e}_k(t) + k_2M_k(q)\dot{\delta}_k(t) \]
\[ = k_1\dot{\delta}_k(t) + k_2\left[ \tau_k + D_\delta(t) + d_\delta - C_\delta(q, \dot{q})\dot{\delta}_k(t) \right] \]
\[ = k_1\dot{\delta}_k(t) + k_2\left[ \tau_k + D_\delta(t) + d_\delta - f - C_\delta(q, \dot{q}) \frac{E_k(t) - k_1\dot{e}_k(t)}{k_2} \right] \]
\[ = k_1\dot{\delta}_k(t) + k_2\left[ \tau_k + D_\delta(t) + d_\delta - f - k_1C_\delta(q, \dot{q})\dot{e}_k(t) - C_\delta(q, \dot{q})E_k(t) \right]. \]
Substituting (67) into (63) renders to

\[
\dot{V}_3(t) = 2E_k^T(t)[k_1 \overline{M}_x(q) \dot{c}_k(t) + k_2 [\tau_k + d_x + D_x(t) - f] + k_1 \overline{C}_x(q,q) \epsilon_k(t) - \overline{C}_x(q,q)E_k(t)] + E_k^T(t) \overline{M}_x(q) E_k(t) \\
= 2E_k^T(t)[k_1 \overline{M}_x(q) \dot{c}_k(t) + k_2 [\tau_k + d_x + D_x(t) - f] + k_1 \overline{C}_x(q,q) \epsilon_k(t)] - 2E_k^T(t) \overline{C}_x(q,q),
\]

\[
E_k(t) + E_k^T(t) \overline{M}_x(q) E_k(t)
\]

\[
= 2E_k^T(t)[k_1 \overline{M}_x(q) \dot{c}_k(t) + [k_2 \tau_k + d_x + D_x(t) - f] + k_1 \overline{C}_x(q,q) \epsilon_k(t)] + E_k^T(t) [\overline{M}_x(q) - 2 \overline{C}_x(q,q)] E_k(t).
\]

According to the Property 1, (68) can be further simplified as follows:

\[
\dot{V}_3(t) = 2E_k^T(t)[k_1 \overline{M}_x(q) \dot{c}_k(t) + k_2 [\tau_k + d_x + D_x(t) - f] + k_1 \overline{C}_x(q,q) \epsilon_k(t)] \\
= 2k_1 E_k^T(t) \overline{M}_x(q) \dot{c}_k(t) + 2k_1 E_k^T(t) \overline{C}_x(q,q) \epsilon_k(t) + 2k_2 E_k^T(t) [\tau_k + d_x + D_x(t) - f].
\]

Secondly, \( \dot{V}_4(t) \) is to be addressed. By considering the RBFNN updating law (28),

\[
\dot{W} = W^* - \dot{\hat{W}} = -\psi (u) E_k^T(t).
\]

Combining (63) and (70) leads to

\[
\dot{V}_4(t) = 2k_2 tr \left[ \overline{W}^T \psi (u) E_k^T(t) \right] \\
= -2k_2 tr \left[ \overline{W}^T \psi (u) E_k^T(t) \right]
\]

Then, substituting (75) and control law (24) into (74) gives

\[
\dot{V}_k(t) = 2k_2 E_k^T(t) \left[ -\delta_k(t) \text{sgn}(\dot{c}_k(t)) + W^* \varphi(u) + \epsilon(u) - \overline{W}^T \varphi(u) \right] - 2k_2 E_k^T(t) \overline{W}^T \varphi(u)
\]

\[
= -2k_2 E_k^T(t) \delta_k(t) \text{sgn}(\dot{c}_k(t)) + 2k_2 E_k^T(t) \left( W^* - \overline{W}^T \right) \varphi(u) - 2k_2 E_k^T(t) \overline{W}^T \varphi(u) + 2k_2 E_k^T(t) \epsilon(u)
\]

\[
= -2k_2 E_k^T(t) \delta_k(t) \text{sgn}(\dot{c}_k(t)) + 2k_2 E_k^T(t) \epsilon(u).
\]
Integrating the two sides of (76), we have

\[
V_k(t) = -2k_2 \int_0^t E_k^T(\eta) \tilde{\delta}_k(\eta) \text{sgn}(\dot{\epsilon}_k(\eta)) \, d\eta \\
+ 2k_2 \int_0^t E_k^T(\eta) \epsilon(u) \, d\eta.
\]

(77)

Combination of (64) and (77) gives that

\[
\Delta W_k(t) = -V_{k-1}(t) - 2k_2 \int_0^t E_k^T(\eta) \tilde{\delta}_k(\eta) \text{sgn}(\dot{\epsilon}_k(\eta)) \, d\eta \\
- k_2^2 \int_0^t \Gamma^{-1} \left( \frac{\epsilon^2}{\tilde{\delta}_k(\eta)} + \frac{\eta}{\tilde{\delta}_k(\eta)} \right) \, d\eta \\
+ 2k_2 \int_0^t E_k^T(\eta) \epsilon(u) \, d\eta.
\]

(78)

From (65), we have

\[
\Delta W_k(t) < -V_{k-1}(t) - 2k_2 \int_0^t E_k^T(\eta) \text{sgn}(\epsilon_k(\eta)) \, d\eta \\
- 2k_2 \int_0^t E_k^T(\eta) \text{sgn}(\epsilon_k(\eta)) \, d\eta \\
+ 2k_2 \int_0^t E_k^T(\eta) \epsilon(u) \, d\eta.
\]

(79)

From (21), we have that

\[
\tilde{\epsilon}_{k}(t) = \frac{E_k^T(t) - k_1 \epsilon_k(t)}{k_2}
\]

(80)

Further, (79) can be simplified as follows:

\[
\Delta W_k(t) < -V_{k-1}(t) + 2k_2 \int_0^t E_k^T(\eta) \epsilon(u) \, d\eta - 2k_2 \int_0^t E_k^T(\eta) \text{sgn}(\epsilon_k(\eta)) \, d\eta
\]

(81)

Further, using the Schwarz inequality,

\[
E_k^T(t) \epsilon(u) \leq \| E_k(t) \| \cdot \| \epsilon(u) \|
\]

(82)

Thus, we can get

\[
\Delta W_k(t) \leq -V_{k-1}(t) - 2k_2 \int_0^t \| E_k(\eta) \| \| \delta - \epsilon(u) \| \, d\eta
\]

(83)

where condition (15) is applied, by selecting \( \delta - \epsilon_N > 0 \). Thus, \( \Delta W_k(t) \leq 0 \), and the Lyapunov-like composite energy function \( W_0(t) \) is a nonincreasing sequence.

Part 5. Finiteness of \( W_0(t) \).

Then, considering the time derivation of (62) and combining it with (76), we have

\[
\dot{W}_0(t) = 2k_2 E_0^T(t) \epsilon(u) + k_2^2 \Gamma^{-1} \tilde{\delta}_0(t),
\]

(84)

\[
-2k_2 E_0(t) \tilde{\delta}_0(t) \text{sgn}(\dot{\epsilon}_0(t)).
\]

Further, according to (15) and (87), (86) can be simplified as follows:

\[
\dot{W}_0(t) \leq -k_2^2 \left| \Gamma^{-1} \left( \frac{\epsilon^2}{\tilde{\delta}_0(t)} + \frac{\eta}{\tilde{\delta}_0(t)} \right) \right| + 2 \| \dot{\epsilon}_0(t) \| \| \epsilon(u) \| - 2k_1 k_2 \| \epsilon_0(t) \| \left( \| \tilde{\delta}_0(t) \| - \| \epsilon(u) \| \right)
\]

(88)
Convergence of totally. As we can see from (21), it is clear that the tracking error will converge to zero asymptotically once the combined error is bounded. In addition, the learning gain is linearly stable for given $E_k(t)$ and $W_k(t)$, $V_i \in [0, T]$ is consistently continuous and bounded.

Part 6. Convergence of $E_k(t)$ and $W_k(t)$.

In order to prove the convergence of error, we have

$$W_k(t) = W_0(t) + \sum_{i=1}^{k} \Delta W_i(t).$$

According to (62), we have

$$W_k(t) \leq W_0(t) + \sum_{i=1}^{k} V_{i-1}(t),$$

$$\leq W_0(t) - \sum_{i=1}^{k} E_i(t) \bar{M}_k(q) E_i(t),$$

$$\sum_{i=1}^{k} k_i \tau_{i} \left[ \bar{W}_{i-1}(t) \psi^{-1} \bar{W}_{i-1}(t) \right].$$

Thus, we have

$$\sum_{i=1}^{k} E_i(t) \bar{M}_k(q) E_i(t) + \sum_{i=1}^{k} k_i \tau_{i} \left[ \bar{W}_{i-1}(t) \psi^{-1} \bar{W}_{i-1}(t) \right],$$

$$\leq (W_0(t) - W_k(t)) \leq W_0(t).$$

Since $W_0(t)$ is bounded and continuous, we have

$$\lim_{k \to \infty} W_k(t) = \lim_{k \to \infty} E_k(t) = 0, \quad t \in [0, T].$$

Therefore, CEF $E_k(t)$ would converge to zero asymptotically. As we can see from (21), it is clear that the tracking error $e_k$ is linearly stable for given $E_k(t)$. Thus, $e_k$ and $\dot{e}_k$ will converge to zero asymptotically once the combined error $E_k(t)$ does. It is worthy of noticing that the boundedness of $x_\alpha$ and $\bar{x}_\alpha$, the generalized end effector trajectories $x_k$, and the velocities $\dot{x}_k$ are bounded. As such, together with the boundedness of $\bar{x}_\alpha$ and the RBF function $\psi(u)$, we can see from (16) and (24) that the control torque $\tau_x$ is bounded. According to the robot dynamics (22), the boundedness of $\dot{x}_k$ is clear. Both $\dot{e}_k(t)$ and $u = [e_k^T, \dot{e}_k^T]^T$ are bounded. In a word, all the signals involved in the closed-loop control system are bounded.

Remark 1. The proposed controllers in this paper show that the tracking error, at the first iteration, can be made arbitrarily small, over the finite time interval $[0, T]$, by adjusting the minimal eigenvalues of control parameters $k_1, k_2$ and learning gain $\Gamma$.

Noticing (21), it follows that

$$\dot{\delta}_{0_{\max}}(t) = \sup_{t \in [0, T]} \delta_i,$$

$$\dot{\delta}_{0_{\max}}(t) = \sup_{t \in [0, T]} \delta_i(t).$$

where

$$\dot{\delta}_{0_{\max}}(t) = \dot{\sup}_{t \in [0, T]} \delta_i(t).$$

According to Assumption 2, we know that the initial set value of the system is bounded, and we can get that $\delta_{0_{\max}}(t)$ is bounded. In addition, $W_0(t)$ is bounded and $W_0(t), V_i \in [0, T]$ is consistently continuous and bounded.

Rule 1. A larger $(k_1/k_2)$ will yield a better transient performance.

Rule 2. Larger $k_1, k_2$ will induce a faster convergence of tracking error $E_k(t)$, due to the high-gain feedback and learning.

Nevertheless, it cannot be denied that the adverse effect of noise/unmodeled disturbances will be amplified when we adopt high-gain feedback or learning. In practical applications, it is hard to employ the high-gain feedback structure because of the limited sampling rate. Thus, taking the trade-off problems between nominal performance and robustness into consideration, it is vital to adopt appropriate control parameters $k_1, k_2$ and learning gain $\Gamma$ by accessing the actual control performance.

As such, considering the trade-off problems between tracking and disturbance rejection and between nominal performance and robustness, it is suggested to design $k_1, k_2$ by following rules (1) and (2), but restrict their magnitudes by assessing the actual control performance.

In next section, there are compared simulations to be summarized to illustrate the effectiveness of the proposed control schemes.

5. Simulation Results

The simulation results which are performed in MATLAB 9.0/Simulink of a personal computer equipped with Windows 10 operating system and Intel CPU core i5 have been carried out on a model of 2DOF robot manipulator to analyze the effectiveness of the controller we developed above. The inertia nominal matrix $M(q)$, the centripetal and Coriolis nominal terms $C(q, \dot{q})$, the gravity nominal vector $G(q)$, and the Jacobian matrix $J(q)$ are given as follows:
\[
M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},
\]
\[
C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix},
\]
\[
G(q) = \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix},
\]
\[
J(q) = \begin{bmatrix} J_{11} \\ J_{21} \end{bmatrix},
\]

where
\[
M_{11} = M(1) + M(2) + 2M(3)\cos(q_2),
\]
\[
M_{12} = M(2) + M(3)\cos(q_2),
\]
\[
M_{21} = M(2) + M(3)\cos(q_1),
\]
\[
M_{22} = M(2),
\]
\[
C_{11} = -M(3)\dot{q}_2 \sin(q_2),
\]
\[
C_{12} = -M(3)(\dot{q}_1 + \dot{q}_2)\sin(q_2),
\]
\[
C_{21} = M(3)\dot{q}_1 \sin(q_2),
\]
\[
C_{22} = 0,
\]
\[
G_{11} = M(4)g \cos(q_1) + M(5)g \cos(q_1 + q_2),
\]
\[
G_{21} = M(5)g \cos(q_1 + q_2),
\]
\[
J_{11} = -I_1 \sin(q_1) - I_2 \sin(q_1 + q_2),
\]
\[
J_{12} = -I_1 \sin(q_1 + q_2),
\]
\[
J_{21} = I_1 \cos(q_1) + I_2 \cos(q_1 + q_2),
\]
\[
J_{22} = I_2 \cos(q_1 + q_2),
\]

where \( M(i) \) is given by \( M = N + p_1 L \).
\[
N = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 \end{bmatrix}^T,
\]
\[
M = \begin{bmatrix} M(1) & M(2) & M(3) & M(4) & M(5) \end{bmatrix}^T,
\]
\[
L = \begin{bmatrix} l_1^2 & l_2^2 & l_1 l_2 \end{bmatrix}^T.
\]

As is shown in Figure 1, the relationship between end effector trajectory and joint angle can be described as follows:
\[
x = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2).
\]
\[
y = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2).
\]

Adding the square of (98) and (99) gives
\[
x^2 + y^2 = l_1^2 + l_2^2 + 2l_1l_2 \cos q_2.
\]

Therefore, we can get
\[
q_2 = \cos^{-1} \left( \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} \right).
\]

According to [22], we define
\[
P_1 = \arctan \frac{y}{x},
\]
\[
P_2 = \arccos \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 \sqrt{x^2 + y^2}}.
\]
Figure 2: Tracking performance of $x$ axis when the pure AILC controller with $k_1 = 14, k_2 = 1.2$, and $\Gamma = 5$ applied.

Figure 3: Tracking performance of $y$ axis when the pure AILC controller with $k_1 = 14, k_2 = 1.2$, and $\Gamma = 5$ applied.

Figure 4: Tracking error profile of $x$ axis when the pure AILC controller with different $k_2$ is applied.
Figure 5: Tracking error profile of $y$ axis when the pure AILC controller with different $k_1$ is applied.

Figure 6: The modeling error and compensation of joint 1.

Figure 7: The modeling error and compensation of joint 2.
Figure 8: Tracking control errors of $x$ axis for pure AILC and the proposed control.

Figure 9: Tracking control errors of $y$ axis for pure AILC and the proposed control.

Figure 10: Tracking error of $x$ axis for pure AILC, other controls, and the proposed control.
shown in Figure 5. Although we can clearly see that, at the beginning, the modeling error changes greatly in Figures 6 and 7, it decreases and becomes stable gradually with the compensation of NNs.

To further test the effectiveness of the proposed controller by add-on RBFNNs, we make a comparison test with our given controller where the neural network compensation part is removed. As can be seen from Figures 8 and 9, the steady-state control errors are further reduced owing to the add-on RBFNNs. For the current scenario of simulation, a combination of AIL and RBFNNs outperforms AILC in tracking control of robot manipulators. In order to illustrate the superiority of our proposed control scheme, we make a comparison with the pure controller AILC with $k_1 = 14$, $k_2 = 1.2$, and $\Gamma = 5$ and the classical approach introduced in [25] with $K_P = K_D = 20J_{2\times2}$ and $\Gamma = 5$, where $I_{i\times i}$ is an $i \times i$ identity matrix. Besides, the proposed method is also compared with the approach introduced in the recent study [49].

As can be seen from Figures 10 and 11, we find that, after 20 iterations, all of four approaches have a good control performance in steady state. However, obviously, we find the proposed control scheme has achieved a better performance in position tracking error as well as the learning speed of the system. Figure 12 illustrates the trajectory of the manipulator end effector in Cartesian space.

6. Conclusion and Future Recommendation

In this paper, we discussed the trajectory tracking problem of the robot manipulator system subject to additional
disturbances and uncertainties. We divide these uncertainties of the robot manipulator system into two parts, the periodic part and nonperiodic part. In order to address these problems, a novel neural network adaptive iterative learning control algorithm is proposed, where the AILC is used to iteratively learn the periodic uncertainties while the NN part is adopted to compensate and approximate all nonperiodic ones. The convergence of the system tracking error is investigated in a Lyapunov-like composite energy function way by introducing a combined error factor, which consists of the weighted sum of tracking error and its derivative. Compared with the pure controller AILC and other classical approaches, the results of simulations show that the performance of tracking error and learning speed is much better by adding NNs. The proposed approach can be widely applied to other robot systems subjected to various uncertainties and disturbances. Our next research phase is to test the proposed control scheme experimentally. In addition, we plan to combine our algorithm with other approximation ways, such as the observer disturbance and the fuzzy systems, to take the place of the neural networks in the future.

Data Availability

The data that support our article’s conclusions are some open-access articles that have been properly cited, and the readers can easily obtain these articles to verify the conclusions.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation (11872069 and 62003278), Key Project of Sichuan Science and Technology Department (2018YJ0463), Sichuan University Research Innovation Team “Machine Vision and Intelligent Control” (18TD0024), National Ministry of Education “Chunhui Plan” Scientific Research Project (Z2017076), and Xihua University Graduate Innovation Fund (YCYJ2020058).

References

[1] Q. Yu, Z. Hou, and J.-X. Xu, “D-type ILC based dynamic modeling and norm optimal ILC for high-speed trains,” IEEE Transactions on Control Systems Technology, vol. 26, no. 2, pp. 652–663, 2018.
[2] G. Liu and Z. Hou, “RBNN-based adaptive iterative learning fault-tolerant control for subway trains with actuator faults and speed constraint,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, pp. 1–15, 2019.
[3] X. Li, Q. Ren, and J. Xu, “Precise speed tracking control of a robotic fish via iterative learning control,” IEEE Transactions on Industrial Electronics, vol. 63, no. 4, pp. 2221–2228, 2016.
[4] D. Meng, Y. Jia, and J. Du, “Robust consensus tracking control for multiagent systems with initial state shifts, disturbances, and switching topologies,” IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 4, pp. 809–824, 2015.
[5] M. Sun, S. S. Ge, and I. M. Y. Mareels, “Adaptive repetitive learning control of robotic manipulators without the requirement for initial repositioning,” IEEE Transactions on Robotics, vol. 22, no. 3, pp. 563–568, 2006.
[6] X.-D. Li, M.-M. Lv, J. K. L. Ho, and J. Ho, “Adaptive ILC algorithms of nonlinear continuous systems with non-parametric uncertainties for non-repetitive trajectory tracking,” International Journal of Systems Science, vol. 47, no. 10, pp. 2279–2289, 2016.
[7] G. Li, C.-E. Ren, C. L. P. Chen, and Z. Shi, “Adaptive iterative learning consensus control for second-order multi-agent systems with unknown control gains,” Neurocomputing, vol. 393, pp. 15–26, 2020.
[8] R. Lee, L. Sun, Z. Wang, and M. Tomizuka, “Adaptive iterative learning control of robot manipulators for friction compensation,” IFAC-PapersOnLine, vol. 52, no. 15, pp. 175–180, 2019.
[9] X. Li, D. Shen, and J.-X. Xu, “Adaptive iterative learning control for MIMO nonlinear systems performing iteration-varying tasks,” Journal of the Franklin Institute, vol. 356, no. 16, pp. 9206–9231, 2019.
[10] X. Bu, S. Wang, Z. Hou, and W. Liu, “Model free adaptive iterative learning control for a class of nonlinear systems with randomly varying iteration lengths,” Journal of the Franklin Institute, vol. 356, no. 5, pp. 2491–2504, 2019.
[11] M. Sun, T. Wu, L. Chen, and G. Zhang, “Neural AILC for error tracking against arbitrary initial shifts,” IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 7, pp. 2705–2716, 2018.
[12] J. M. Wilfing and S. S. Kumar, “Joschka Boedecker, martin Riedmiller, Ulrich Eger,” adaptive long-term control of biological neural networks with deep reinforcement learning,” Neurocomputing, vol. 342, pp. 66–74, 2019.
[13] P. Xu and Q. Xu, “Neural network-based AILC for nonlinear discrete-time system with iteration varying initial error and reference trajectory,” in Proceedings of the 2018 5th International Conference on Information Science and Control Engineering (ICISCE), pp. 854–858, Zhengzhou, China, July 2018.
[14] K. Patan, M. Patan, and D. Kowalow, “Neural networks in design of iterative learning control for nonlinear systems * *This work was supported by National Science Centre in Poland under the grant 2014/15/B/ST7/03208,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 13402–13407, 2017.
[15] X.-S. Wang, Yu-Hu Cheng, and W. Sun, “Constructive RBF network based iterative learning controller for manipulators,” in Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, vol. 8, pp. 4714–4719, Guangzhou, China, August 2007.
[16] Y. Wang, C. Chien, and R. Chi, “A fuzzy neural network based adaptive terminal iterative learning control for nonlinear discrete-time systems,” in Proceedings of the 2014 International Conference on Fuzzy Theory and Its Applications (IFUZZY2014), pp. 163–167, Kaohsiung, China, November 2014.
[17] Y.-C. Wang, C.-J. Chien, and D.-T. Lee, “A fuzzy neural network direct adaptive iterative learning controller for robotic systems,” IFAC Proceedings Volumes, vol. 41, no. 2, pp. 6519–6524, 2008.
[18] J. Li and J. Li, “Adaptive fuzzy iterative learning control with initial-state learning for coordination control of leader-following multi-agent systems,” Fuzzy Sets and Systems, vol. 248, pp. 122–137, 2014.
[19] J. Xu, R. Yan, J. Xu, and R. Yan, "Adaptive learning control for finite interval tracking based on constructive function approximation and wavelet," *IEEE Transactions on Neural Networks*, vol. 22, no. 6, pp. 893–905, 2011.

[20] C. He, J. Li, C. He, and J. Li, "Adaptive learning control for finite interval tracking based on variable wavelet neural network," in *Proceedings of the 32nd Chinese Control Conference*, pp. 3031–3036, Xi’an, China, July 2013.

[21] K.-S. Tzeng, D. C. Tzeng, and J.-S. Chen, "An enhanced iterative learning control scheme using wavelet transform," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 3, pp. 922–924, 2005.

[22] S. S. Ge, C. C. Hang, and L. C. Woon, "Adaptive neural network control of robot manipulators in task space," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 6, pp. 746–752, 1997.

[23] W. He, Z. Yan, Y. Sun, Y. Ou, and C. Sun, "Neural-learning-based control for a constrained robotic manipulator with flexible joints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 6093–6003, 2018.

[24] 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.

[25] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.

[26] X. Dong, Y. Zhao, H. R. Karimi, and P. Shi, "Adaptive variable structure fuzzy neural identification and control for a class of MIMO nonlinear system," *Journal of the Franklin Institute*, vol. 350, no. 5, pp. 1221–1247, 2013.

[27] S. Li, J. He, Y. Li, and M. U. Rafique, "Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 2, pp. 415–426, 2017.

[28] R. Wai and R. Muthusamy, "Fuzzy-neural-network inherited sliding-mode control for robot manipulator including actuator dynamics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 2, pp. 274–287, 2013.

[29] S. Li, M. Zhou, and X. Luo, "Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4791–4801, 2018.

[30] T. Li, S. Duan, J. Liu, L. Wang, and T. Huang, "A spinntron transimpedance-based neural network with radial basis function for robotic manipulator control implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 4, pp. 582–588, 2016.

[31] S. Mobayen, "Design of LMI-based sliding mode controller with an exponential policy for a class of underactuated systems," *Complexity*, vol. 21, no. 5, pp. 117–124, 2016.

[32] S. Mobayen and J. Ma, "Robust finite-time composite nonlinear feedback control for synchronization of uncertain chaotic systems with nonlinearity and time-delay," *Chaos, Solitons & Fractals*, vol. 114, pp. 46–54, 2018.

[33] J. Moreno-Valenzuela and L. González-Hernández, "Operational space trajectory tracking control of robot manipulators endowed with a primary controller of synthetic joint velocity," *ISA Transactions*, vol. 50, no. 1, pp. 131–140, 2011.

[34] M. Galicki, "Robust task space finite-time chattering-free control of robotic manipulators," *Journal of Intelligent & Robotic Systems*, vol. 85, no. 3–4, pp. 471–489, 2017.

[35] S. Mobayen and F. Tchier, "Nonsingular fast terminal sliding-mode stabilizer for a class of uncertain nonlinear systems based on disturbance observer," *Scientia Iranica*, vol. 24, no. 3, pp. 1410–1418, 2017.

[36] J. -. Slotine and W. Li, "Adaptive manipulator control a case study proceedings," in *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, pp. 1392–1400, Raleigh, NC, USA, March–April 1987.

[37] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: a tutorial," in *Proceedings of the 27th IEEE Conference on Decision and Control*, vol. 2, pp. 1575–1584, Austin, TX, USA, December 1988.

[38] Y. Zhu and Z. Hou, "Data-driven MFAC for a class of discrete-time nonlinear systems with RBF neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 1013–1020, 2014.

[39] S. Seshagiri and H. K. Khalil, "Output feedback control of nonlinear systems using RBF neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 69–79, 2000.

[40] H. Han and J. Qiao, "Adaptive computation algorithm for RBF neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 2, pp. 342–347, 2012.

[41] H.-G. Han, L. Zhang, Y. Hou, and J.-F. Qiao, "Nonlinear model predictive control based on a self-organizing recurrent neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 402–416, 2016.

[42] S. Han, H. Wang, Y. Tian, and N. Christov, "Time-delay estimation based computed torque control with robust adaptive RBF neural network compensator for a rehabilitation exoskeleton," *ISA Transactions*, vol. 97, pp. 171–181, 2020.

[43] F. Liu and S. Fan, "Adaptive RBFNN based fuzzy sliding mode control for two link robot manipulator," in *Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence*, pp. 272–276, Shanghai, China, November 2009.

[44] A. O. Hoori and Y. Motai, "Multicolumn RBF network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 766–778, 2018.

[45] S. Xie, Y. Xie, T. Huang, W. Gui, and C. Yang, "Generalized predictive control for industrial processes based on neuron adaptive splitting and merging RBF neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1192–1202, 2019.

[46] Y. Yang, D. Huang, and X. Dong, "Enhanced neural network control of lower limb rehabilitation exoskeleton by add-on repetitive learning," *Neurocomputing*, vol. 323, pp. 256–264, 2019.

[47] Y. Zuo, Y. Wang, H. Gui, and C. Yang, "Generalized predictive control for industrial processes based on neuron adaptive splitting and merging RBF neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1192–1202, 2019.

[48] B. Farah and Z. Michel, "Trajectory tracking control for perturbed robot manipulators using iterative learning method," *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 5, pp. 2013–2022, 2016.