

Research Article

Trajectory Tracking Control of Robot Manipulators Based on U-Model

Xianghua Ma , Yang Zhao, and Yiqun Di

School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai 201418, China

Correspondence should be addressed to Xianghua Ma; xhuam@sit.edu.cn

Received 6 April 2020; Accepted 20 April 2020; Published 12 May 2020

Guest Editor: Weicun Zhang

Copyright © 2020 Xianghua Ma 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.

A new trajectory tracking control method based on the U-model is proposed to improve the trajectory tracking speed of robot manipulators. The U-model method is introduced to relieve the requirement of the dynamic mathematical model and make the design of trajectory tracking controller of robot manipulators simpler. To further improve the trajectory tracking speed, an improved iterative learning control algorithm is used to suppress the influence of the initial state error with less computation time. Experimental results show that the proposed control method is effective and practical for the trajectory tracking control of robot manipulators, especially with a high real-time requirement.

1. Introduction

Nowadays, most robot manipulators need to possess the capacity of accurate and fast trajectory tracking. Trajectory tracking control is a key issue in the field of robot manipulator motion planning [1–3]. It aims to enable the joints or links of the robot manipulator to track the desired trajectory with ideal dynamic quality or to stabilize them in the specified position [4]. At present, there are two main ways to analyze and design the trajectory tracking control for robot manipulator [5]. The first one is with respect to a classical linear control method [6, 7]. This way is easy, and some classical control theory and methods can be used directly in some nonlinear properties which are approximated into a linear relationship or neglected. However, the trajectory tracking control of robot manipulators is a multi-input and multioutput (MIMO) control problem, and the motion equation of a robot manipulator with multiple degrees of freedom is highly coupled and complex [8]. If each joint is supposed to be independent and the inertia of each joint be constant, it may lead to inconsistent system damping and other unexpected situations in the entire workspace [9]. Therefore, there is the second way, which focuses on the nonlinear control method without the requirements of the above assumptions [10–12]. However, this nonlinear

method also has some limits; for example, it needs to get the exact data of the structure and precise parameters when modelling the manipulator so as to correctly describe all these uncertainties, for example, the interaction among robot joints and the change in centroid.

The initial state error has a great influence on the trajectory tracking speed. Iterative learning control method is used to rapidly suppress the influence of the initial state error. Hongfeng et al. [13] proposed an iterative learning trajectory tracking control algorithm for discrete nonlinear systems in which the initial states are not strictly identical to given expected values in track tracking. Jin [14] proposed a new iterative learning control method which can track the nonrepetitive motion trajectory without assuming that the initial state error of the system is unchanged. Hui et al. [15] proposed a data-driven iterative learning control method based on an extended state observer. The concept of ESO is introduced into the iteration direction. The random initial state and the disturbance are taken as the extended state of a whole. Theoretical analysis shows that the method is robust to initial displacements and perturbations of iterative changes.

A reasonable learning gain matrix can improve the trajectory tracking speed. In order to obtain the optimal learning gain, many researches have been done including

variable exponential gain method [16], fuzzy PID method [17], and the method combining neural network controller and compensation controller [18]. However, the complex control law and the learning law will bring a huge amount of computation time and affect the trajectory tracking speed.

According to the above analysis, both the classical linear control method and the nonlinear control method have their advantages and also some limitations, respectively. If there is a way to combine the benefits of the classical linear and the nonlinear control methods, not only the analysis and design of trajectory tracking control can be easier but also the speed and accuracy of trajectory tracking control may be improved as well. U-Model belongs to a generic systematic approach, which converts the nonlinear polynomial model into a controller output-based time-varying polynomial model [19–23]. Inspired by this idea, the U-model theory is introduced into the trajectory tracking control, which can help understand the inner structure and parameters of the robot manipulator better and reduce the structure complexity of control system. Iterative learning control method is used with a new iterative learning control law, which can greatly decrease the computation time and further improve the trajectory tracking speed.

The remainder of the paper is organized as follows. Section 2 describes trajectory tracking U-control system in detail. Section 3 analyzes the asymptotic convergence of the control law and the stability of the system. In Section 4, simulation experiment is carried out to prove the effectiveness of the proposed method. Conclusions are given in Section 5.

2. Trajectory Tracking U-Control System

In this paper, a trajectory tracking U-control method is proposed, and the control system mainly includes robot manipulator and a U-controller, as shown in Figure 1. The U-controller is composed with an iterative learning controller (ILC for abbreviation) and the U-model of a robot manipulator, where G_c is the mathematical model of the linear controller, G_p is the mathematical model of the robot manipulator, and G_p^{-1} is the dynamic inverse of robot manipulator model based on U-model theory. The desired output trajectory is denoted as $y_d(t)$, and the system output is denoted as $y_k(t)$ at time $t \in R^+$. k is the number of iterations.

2.1. Realization of U-Controller. Consider a robot manipulator with n DOF; its general dynamic model can be represented as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + u_a = u, \quad (1)$$

where $q_{n \times 1} = [q_1, q_2, \dots, q_n]^T \in R^n$ is the angular displacement vector; $\dot{q}_{n \times 1}$ and $\ddot{q}_{n \times 1}$ are angular velocity and angular acceleration vectors of each joint, respectively; $u \in R^{n \times 1}$ is the torque vector; $M(q) \in R^{n \times n}$ represents the inertia term; $C(q, \dot{q}) \in R^{n \times n}$ is the Coriolis and centrifugal terms; $G(q) \in R^n$ is the gravitational term; and $u_a \in R^{n \times n}$ is the unknown disturbance term.

Based on the U-model theory, the mathematical model G_p of the robot manipulator (1) can be converted into the following U-model expression:

$$\begin{aligned} \ddot{y}(t) = & -M^{-1}y(t)[C(y(t), \dot{y}(t))\dot{y}(t) + G(y(t), \dot{y}(t))] \\ & + M^{-1}y(t)u(t) - M^{-1}y(t)u_a, \end{aligned} \quad (2)$$

where $y(t) = q(t)$.

In order to consistent with the U-model theoretical expression, equation (2) can be transformed as follows:

$$v_k(t) = \alpha_0(t) + \alpha_1(t)u_k(t) + E(t), \quad (3)$$

where

$$\begin{cases} v_k(t) = \ddot{y}(t), \\ \alpha_0(t) = -M^{-1}y(t)[C(y(t), \dot{y}(t))\dot{y}(t) + G(y(t), \dot{y}(t))], \\ \alpha_1(t) = M^{-1}y(t), \\ E(t) = -M^{-1}y(t)u_a. \end{cases} \quad (4)$$

From the transformation of equation (2) to equation (3), it can be seen that the realization of the U-model is different from the general approximate linear transformation and the properties of the original robot manipulator dynamic model will not be changed.

As seen from Figure 1, the U-controller consists of ILC and the dynamic inverse controller G_p^{-1} based on the U-model. After the U-model expression G_p of the manipulator dynamic model (3) is obtained, the U-controller can be designed based on the U-model theory. The output of the ILC is shown as follows:

$$v_k(t) = v_{k-1}(t) + G_c e(t), \quad (5)$$

where trajectory error $e(t) = y_d(t) - y_k(t)$.

According to the definition of U-model theory, in this trajectory tracking U-control system, the general mathematical express of G_p can be obtained by

$$v_k^{(M)}(t) = \sum_{j=0}^M \alpha_j(t) u_k^{(N)}(t) + E(t), \quad M > N, \quad (6)$$

where $v_k^{(M)}(t)$ and $u_k^{(N)}(t)$ are the M th and N th orders of derivatives of $v_k(t)$ and $u_k(t)$, respectively. $\alpha_j(t)$ is a time-varying coefficient of $u_k^{(N-1)}(t), \dots, u_k(t)$ and $v_k^{(M-1)}(t), \dots, v_k(t)$.

By solving the root of equation (6), the output $u_k(t)$ of the U-controller can be obtained and expressed as

$$u_k(t) = F \left[v_k^{(M)}(t) - \sum_{j=0}^M \alpha_j(t) u_k^{(N)}(t) - E(t) \right], \quad (7)$$

where $F[*]$ is a root-solving algorithm. In order to improve the convergence speed of iterative algorithms, based on the Newton–Raphson iterative algorithm, the model of the U-controller is obtained as shown in equation (8), which can optimize the output of ILC so as to reduce the number of iterations and speed up the tracking convergence.

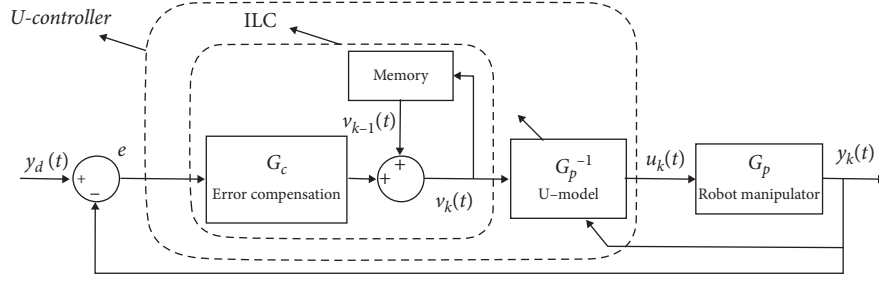


FIGURE 1: U-Control system framework.

The mathematical model of the U-controller can be expressed as

$$u_k(t) = \frac{v_k(t) - \alpha_0(t) - E(t)}{\alpha_1(t)}. \quad (8)$$

From the above equation, it can be seen that the output of the U-controller only needs to search a single value of $u_k(t)$ and obtain the output of ILC through fewer iterations, which greatly reduces the complexity of operation.

2.2. Iterative Learning Control Law. Iterative learning control algorithm is a common method for trajectory tracking control of manipulator. In order to improve the trajectory tracking speed of the manipulator, it is necessary to reduce the number of iterations required for the convergence while ensuring the convergence of the iterative learning control law. Therefore, both the initial state error and the convergence speed of the error are considered when the iterative learning control law is designed.

Let $x_k(t) = [q \ \dot{q}]^T$, the robot manipulator (1) can be rewritten into the following expression (9), and the displacement and velocity of each joint of the manipulator are used as the system output:

$$\begin{cases} \dot{x}_k(t) = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = f(x_k(t), t) + B(t)u_k(t) + W_k(t), \\ y_k(t) = C(t)x_k(t) + V_k(t), \end{cases} \quad (9)$$

where $W_k(t)$ and $V_k(t)$ are state disturbance and output disturbance, respectively. k is the number of iterations.

$$\begin{aligned} f(x_k(t), t) &= \begin{bmatrix} \dot{q} \\ -M^{-1}(C(q, \dot{q})\dot{q} + G(q)) \end{bmatrix}, \\ B(t) &= \begin{bmatrix} 0 \\ M^{-1}(q) \end{bmatrix}, \\ W_k(t) &= \begin{bmatrix} 0 \\ -M^{-1}(q) \end{bmatrix} u_a, \\ C(t) &= I. \end{aligned} \quad (10)$$

It is assumed that the above robot manipulator system satisfies the following conditions.

Assumption 1. $f(x_k(t), t)$ satisfies the Lipschitz condition; that is, there exists a constant L_f ($L_f > 0$) that satisfies the following:

$$\|f(x_{k+1}(t), t) - f(x_k(t), t)\| \leq L_f \|x_{k+1}(t) - x_k(t)\|. \quad (11)$$

Assumption 2. The adjacent state disturbance variation and the output disturbance variation are bounded, expressed as follows:

$$\begin{cases} \|W_{k+1}(t) - W_k(t)\| \leq b_W, \\ \|V_{k+1}(t) - V_k(t)\| \leq b_V. \end{cases} \quad (12)$$

Assumption 3. $B(t)$ and $C(t)$ are bounded.

Assumption 4. Expected trajectory $y_d(t)$ is continuous for all $t \in [0, T]$.

It is assumed that the system has a random initial error, which is denoted as $e_k(t) = y_d(t) - y_k(t)$, and the initial state of the k th iteration is $x_k(0)$. The control law is as follows:

$$v_k(t) = v_{k-1}(t) + \Gamma e_k(t) + \Gamma \dot{e}_k(t) + \phi_k(t)X_k(0), \quad t \in [0, T], \quad (13)$$

where Γ is the constant gain matrix and $G_c = \Gamma(e_k(t) + \dot{e}_k(t)) + \phi_k(t)X_k(0)$.

The learning law for the initial state is

$$\phi(t) = \begin{cases} \frac{2a^k}{h} \left(1 - \frac{a^k}{h}t\right), & t \in \left[0, \frac{h}{a^k}\right], \\ 0, & t \in \left[\frac{h}{a^k}, T\right], \end{cases} \quad (14)$$

$$a > 1, 0 < h < T,$$

$$X_k(0) = B(0)\Gamma e_k(0) + x_k(0) - x_{k+1}(0), \quad (15)$$

where $a > 1, 0 < h < T$.

From equation (14), the initial state error can only affect the trajectory tracking speed within the time interval $[0, (h/a^k)]$. After $t = h/a^k$, the initial state error will be 0. With the increase in the number of iterations k , the time instant h/a^k will tend to 0 quickly, which means the time the initial state error affects the trajectory tracking speed will become very short as well. It means that the trajectory tracking speed can be improved by the use of the proposed iterative learning control law.

3. Convergence Analysis

The stability of the trajectory tracking system requires that the actual output trajectory of the system converges to the desired trajectory under the action of the proposed U-controller; that is, the error between expected and actual trajectory will converge to 0 or within a boundary finally.

Lemma 1. For the robot manipulator described in equation (9), if it satisfies the following conditions,

$$\rho = \sup_{t \in [0, T]} \|1 - M^{-1}(q)G\| < 1, \quad (16)$$

then the tracking error of the system will eventually converge to a boundary.

Proof. From equation (9), we can get

$$\begin{aligned} & x_{k+1}(t) - x_k(t) \\ &= \int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds \\ &+ \int_0^t B(s)[u_{k+1}(s) - u_k(s)] ds + \int_0^t [W_{k+1}(s) \\ &- W_k(s)] ds + x_{k+1}(0) - x_k(0). \end{aligned} \quad (17)$$

According to equation (13), there is

$$\begin{aligned} & \int_0^t B(s)[v_{k+1}(s) - v_k(s)] ds \\ &= \int_0^t B(s)[\Gamma e_k(s) + \Gamma \dot{e}_k(s) + \phi_k(s)X_k(0)] ds \\ &= \int_0^t B(s)\Gamma e_k(s) ds + B(t)\Gamma e_k(t) - \int_0^t e_k(s)[\dot{B}(s)\Gamma + B(s)\dot{\Gamma}] ds \\ &+ X_k(0) \int_0^t [B(s)\phi_k(s) - 1] ds + x_k(0) - x_{k+1}(0). \end{aligned} \quad (18)$$

Then, equation (17) can turn into

$$\begin{aligned} & x_{k+1}(t) - x_k(t) \\ &= \int_0^t [f(x_{k+1}(s), s) - f(x_k(s), s)] ds + \int_0^t [W_{k+1}(s) - W_k(s)] ds \\ &+ \int_0^t B(s)\Gamma e_k(s) ds + B(t)\Gamma e_k(t) - \int_0^t e_k(s)[\dot{B}(s)\Gamma + B(s)\dot{\Gamma}] ds \\ &+ X_k(0) \int_0^t [B(s)\phi_k(s) - 1] ds. \end{aligned} \quad (19)$$

According to the tracking error definition $e_k(t) = y_d(t) - y_k(t)$,

$$\begin{aligned} & e_{k+1}(t) - e_k(t) \\ &= y_d(t) - y_{k+1}(t) - y_d(t) + y_k(t) \\ &= -C(t)[x_{k+1}(t) - x_k(t)] - [V_{k+1}(t) - V_k(t)]. \end{aligned} \quad (20)$$

Substituting equation (19) into equation (20),

$$\begin{aligned} & e_{k+1}(t) \\ &= e_k(I - CB\Gamma) - C(t) \left[\int_0^t [f(x_{k+1}(t), t) - f(x_k(t), t)] ds \right. \\ &+ \int_0^t B(s)\Gamma e_k(s) ds - \int_0^t e_k(s)[\dot{B}(s)\Gamma + B(s)\dot{\Gamma}] ds \left. \right] \\ &- C(t) \int_0^t [W_{k+1}(s) - W_k(s)] ds \\ &- C(t)X_k(0) \int_0^t [B(s)\phi_k(s) - 1] ds - [V_{k+1}(t) - V_k(t)]. \end{aligned} \quad (21)$$

According to equation (14), when $t \in [(h/a^k), T]$, we can get

$$\int_0^t (B(t)\phi_k(t) - 1) ds = 0. \quad (22)$$

According to Bellman–Gronwall lemma, equation (19), and assumptions (1) and (2), we have

$$\begin{aligned} & \|x_{k+1}(t) - x_k(t)\| \\ &\leq \int_0^t L_f \|x_{k+1}(s) - x_k(s)\| ds + b_W s|_0^t + \int_0^t \|B(s)\Gamma\| \|e_k(s)\| ds \\ &+ \|B(t)\Gamma\| \|e_k(t)\| + \int_0^t \|\dot{B}(s)\Gamma + B(s)\dot{\Gamma}\| \|e_k(s)\| ds. \end{aligned} \quad (23)$$

Multiply both sides of equation (23) by $e^{-\lambda t}$, where $\lambda > 0$, and take the maximum value of the right-hand side, and assume that $t = t_{\max} \in [0, T]$:

$$\begin{aligned} & \|x_{k+1}(t) - x_k(t)\|_\lambda \\ &\leq \frac{L_f}{\lambda} \|x_{k+1}(t) - x_k(t)\|_\lambda + b_W t + \frac{1 + \lambda}{\lambda} \sup_t \|B\Gamma\| \|e_k(t)\|_\lambda \\ &+ \frac{\sup_t \|\dot{B}\Gamma + B\dot{\Gamma}\|}{\lambda} \|e_k(t)\|_\lambda (t = t_{\max}). \end{aligned} \quad (24)$$

Let $L_f < \lambda$, and equation (24) can be simplified as

$$\begin{aligned} & \|x_{k+1}(t) - x_k(t)\|_\lambda \leq \frac{\lambda b_W t}{\lambda - L_f} \\ &+ \frac{(1 + \lambda) \sup_t \|B\Gamma\| + \sup_t \|\dot{B}\Gamma + B\dot{\Gamma}\|}{\lambda - L_f} \|e_k(t)\|_\lambda. \end{aligned} \quad (25)$$

In the same way, equation (21) can turn into

$$\begin{aligned}
\|e_{k+1}(t)\|_{\lambda} &\leq \|e_k(t)\|_{\lambda} \rho + \frac{\sup_t \|C(t)\|_{L_f}}{\lambda} \|x_{k+1}(t) - x_k(t)\|_{\lambda} \\
&+ \frac{\sup_t \|C(t)\| (\sup_t \|B\Gamma\| + \sup_t \|\dot{B}\Gamma + B\dot{\Gamma}\|)}{\lambda} \|e_k(t)\|_{\lambda} \\
&+ \sup_t \|C(t)\| b_w t + b_v.
\end{aligned} \tag{26}$$

Substituting equation (25) into (26),

$$\|e_{k+1}(t)\|_{\lambda} \leq P \|e_k(t)\|_{\lambda} + Q, \tag{27}$$

where

$$\begin{aligned}
P &= \rho + \sup_t \|C(t)\| \frac{(L_f + 1) \sup_t \|B\Gamma\| + \sup_t \|\dot{B}\Gamma + B\dot{\Gamma}\|}{\lambda - L_f}, \\
Q &= \frac{\lambda \sup_t \|C(t)\| b_w t}{\lambda - L_f} + b_v.
\end{aligned} \tag{28}$$

When the value of λ is big enough, according to equation (16), we can get $P \approx \rho < 1$. When λ is chosen, Q becomes a constant. Further deform equation (27) into

$$\|e_{k+1}(t)\|_{\lambda} \leq \rho^k \left(\|e_k(t)\|_{\lambda} - \frac{Q}{1-M} \right) + \frac{Q}{1-M}. \tag{29}$$

Therefore, we can have

$$\|e_k(t)\|_{\lambda} \leq \rho^{k-1} \left(\|e_{k-1}(t)\|_{\lambda} - \frac{Q}{1-M} \right) + \frac{Q}{1-M}. \tag{30}$$

With the number of iterations $k \rightarrow \infty$, we have $\rho^{k-1} \rightarrow 0$ ($\rho < 1$) and the following inequality:

$$\lim_{k \rightarrow \infty} \|e_k(t)\|_{\lambda} \leq \frac{\lambda \sup_t \|C(t)\| b_w t}{(1-\rho)(\lambda - L_f)} + \frac{b_v}{1-\rho}. \tag{31}$$

From equation (31), it can be seen that, after time $t = h/a^k$, the initial state error no longer affects the tracking error, and the tracking error is only determined by the degree of inaccuracy of the system model parameters and other external disturbances. For the value of the right part of inequality is constant, the tracking error of the system will eventually converge to a boundary.

4. Simulations

A 2-DOF planar robot manipulator is used to prove the effectiveness of the proposed control method. As shown in Figure 2, the parameters are as follows: link 1: length l_1 and mass m_1 , the distance between the center of mass of link 1 and base joint is l_{c1} , and the inertia of link 1 is I_1 . Link 2: length l_2 and mass m_2 ; the distance between the center of mass of link 2 and joint 1 is l_{c2} , and the inertia of link 2 is I_2 .

The robot manipulator is modeled by the Lagrange method. The parameters of inertia matrix $M(q)_{2 \times 2}$, Coriolis and centrifugal matrix $C(q, \dot{q})_{2 \times 2}$, and gravity term matrix $G(q)_{2 \times 1}$ are set as follows:

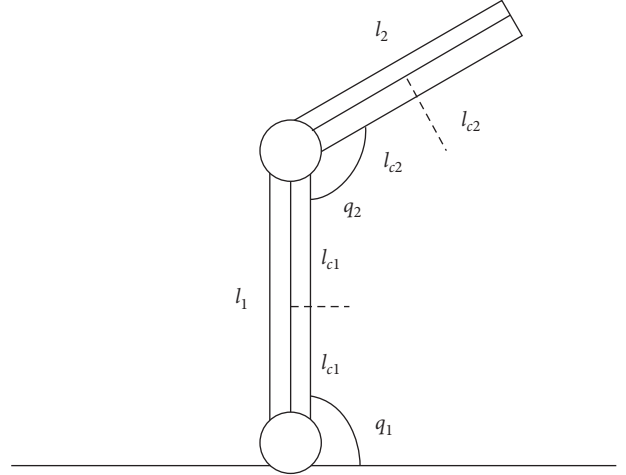


FIGURE 2: 2-DOF robot manipulator.

$$\begin{aligned}
M &= [m_{ij}]_{2 \times 2}, \\
m_{11} &= m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_2 \cos q_2) + I_1 + I_2, \\
m_{12} &= m_{21} = m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) + l_2, \\
m_{22} &= m_2 l_{c2}^2 + I_2, \\
C &= [c_{ij}]_{2 \times 2}, \\
c_{11} &= h \dot{q}_2, \\
c_{12} &= h \dot{q}_1 + h \dot{q}_2, \\
c_{21} &= -h \dot{q}_1, \\
c_{22} &= 0, \\
h &= -m_2 l_1 l_{c2} \sin q_2, \\
G &= [G_1, G_2]^T, \\
G_1 &= (m_1 l_{c1} + m_2 l_1) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2), \\
G_2 &= m_2 l_{c2} g \cos(q_1 + q_2).
\end{aligned} \tag{32}$$

The robot system parameters are

$$\begin{aligned}
m_1 &= m_2 = 1, \\
l_1 &= l_2 = 0.5, \\
l_{c1} &= l_{c2} = 0.25, \\
I_1 &= I_2 = 0.1, \\
g &= 9.81.
\end{aligned} \tag{33}$$

Set the expected trajectory of the system as

$$\begin{cases} y_{1d} = \sin(3t), \\ y_{2d} = \cos(3t). \end{cases} \tag{34}$$

The constant gain matrix of iterative learning control law is set as

$$\Gamma = \begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}. \tag{35}$$

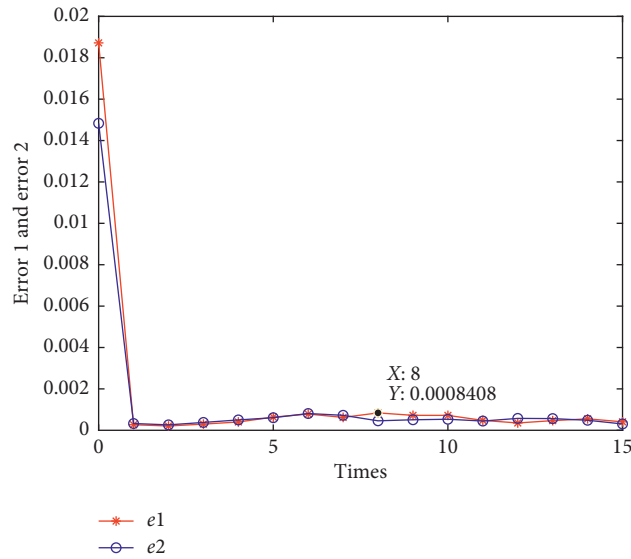


FIGURE 3: Error convergence curve with U-model.

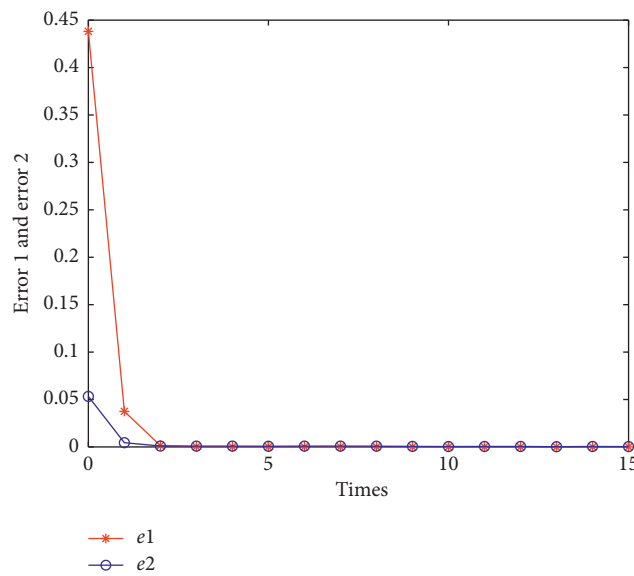


FIGURE 4: Error convergence curve without U-model.

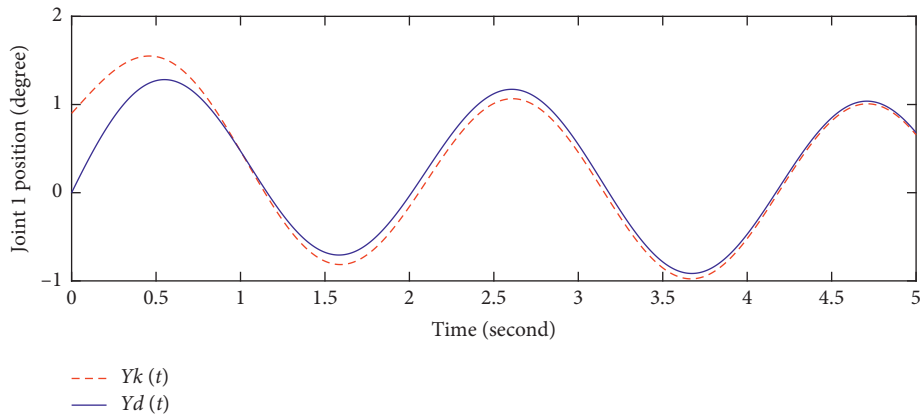
The remaining simulation parameters are initial input $u_0(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $h = 0.1$, and $a = 2$. The initial state of the system is randomly generated by the random function.

In order to illustrate the valid of the proposed U-controller, two experiments are carried out to show the difference of displacement errors rates of the two joints of the robot manipulator with and without U-model, respectively, and the results are shown in Figures 3 and 4. x axle represents the number of iterations, and y axle represents the displacement errors of joint 1 and joint 2, respectively.

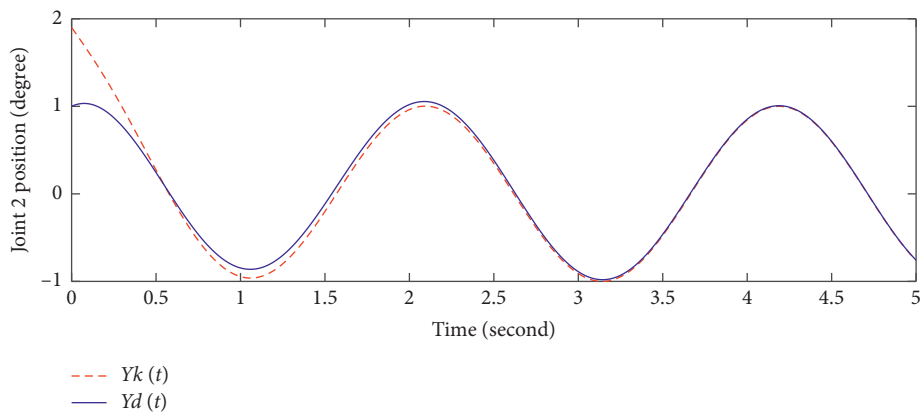
From Figure 3, we can see that when the U-model method is used, only one iteration is needed for each joint to make the displacement error converge to a boundary. However, at least

two iterations for each joint are needed without the use of U-model; from the Figure 4, we can see that the angular displacement errors of joint 1 and joint 2 converge to 0 after three iterations. It means that the U-model control framework can reduce the number of iterations and effectively improve the tracking speed. However, there are displacement errors caused by the U-model, but the loss of tracking accuracy can be acceptable for a robot manipulator, especially when it is used in the fields with high real-time requirement.

In order to show the valid of the proposed ILC law, two experiments are carried out using the U-controller with different ILC laws. The first ILC law is designed in this paper, and the other is the ILC only considering the initial state error. The number of iterations k is 5. The trajectory tracking

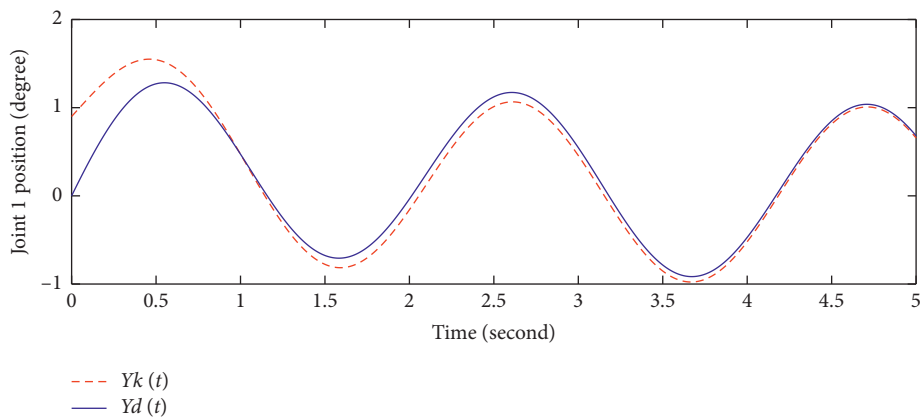


(a)



(b)

FIGURE 5: Trajectory tracking result with the proposed ILC law with 5 iterations.



(a)

FIGURE 6: Continued.

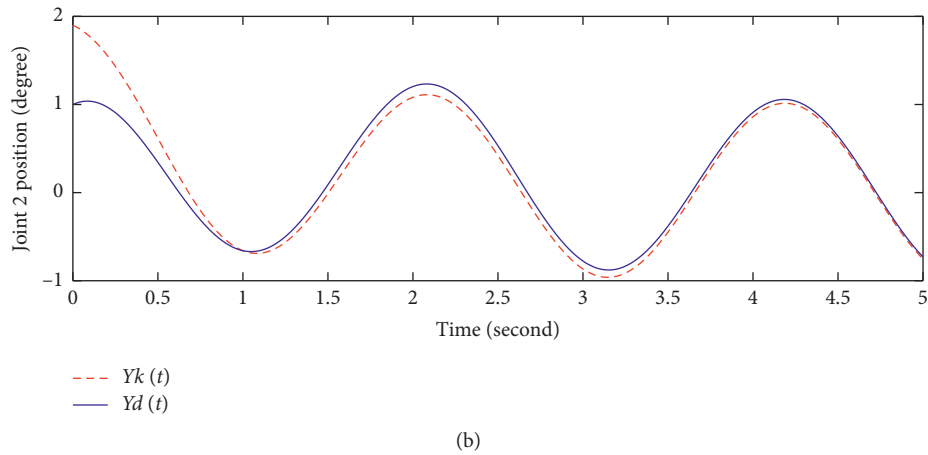


FIGURE 6: Trajectory tracking result with the other ILC law with 5 iterations.

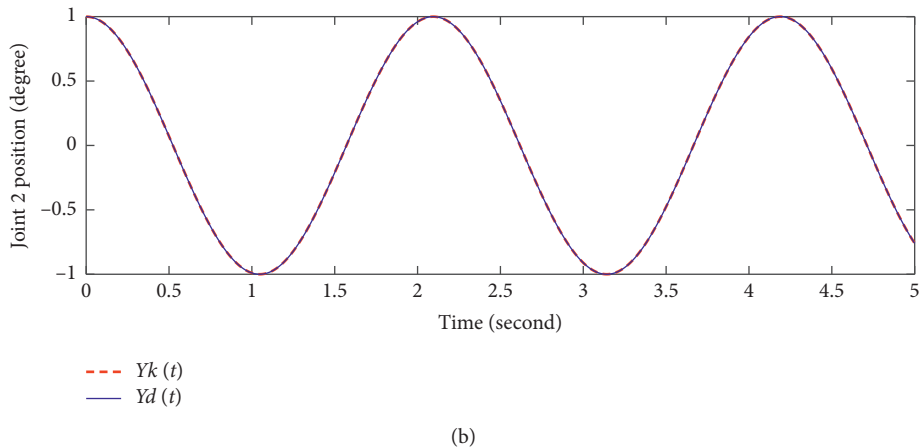
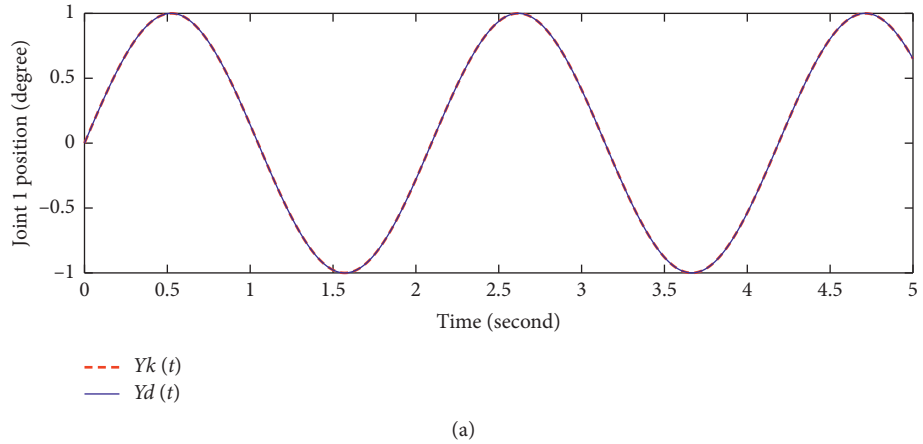


FIGURE 7: Trajectory tracking result with the proposed ILC law with 10 iterations.

curves are shown in Figures 5 and 6. The blue solid line stands for the expected trajectory, and the red dotted line stands for the actual trajectory of the robot manipulator.

From Figure 5, we can see that the actual output trajectory has totally tracked the desired trajectory after 2.5 seconds, while in Figure 6, it takes about 4.5 seconds for the

robot manipulator to track the desired trajectory. By the proposed ILC law, the speed of trajectory tracking is improved more obviously than other ILC law, which does not consider the convergence speed of the initial state error.

With the increasing k , the benefit of the proposed ILC law will be more obvious. We set $k = 10$ and repeat the above

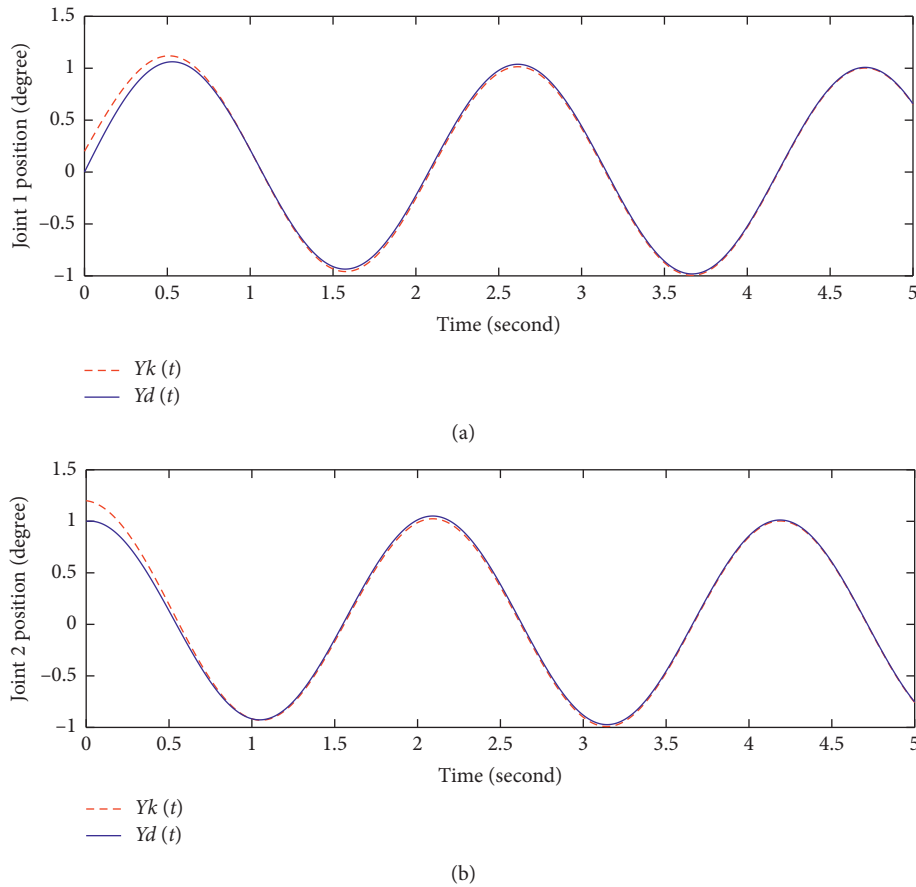


FIGURE 8: Trajectory tracking result with the other ILC law with 10 iterations.

experiment with the tracking results, as shown in Figures 7 and 8. In Figure 7, by using the proposed method, the actual trajectory output is totally tracking the desired trajectory even from the beginning, while in Figure 8, there is still tracking error until 0.5 seconds later.

5. Conclusions

The proposed trajectory tracking U-control method has a simple structure and does not need to change the original nonlinear characteristics of the manipulator system. The iterative learning algorithm based on the initial state errors and the speed of error convergence is designed to effectively suppress the influence made by the initial errors and fasten the trajectory tracking. The U-model control framework can reduce the number of iterations of ILC law and further improve the speed of the trajectory tracking. The proposed trajectory tracking the U-control method is particularly suitable for real-time applications.

Data Availability

The detailed mechanism model and parameters of U-controller are given in the manuscript. The results are computed on the Matlab software with the model and given parameters, while the relevant results are also given in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8112–8123, 2018.
- [2] M. Mehdi Fateh, S. Azargoshasb, and S. Khorashadizadeh, "Model-free discrete control for robot manipulators using a fuzzy estimator," *COMPEL—The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 33, no. 3, pp. 1051–1067, 2014.
- [3] I. Koryakovskiy, M. Kudruss, R. Babuška et al., "Benchmarking model-free and model-based optimal control," *Robotics and Autonomous Systems*, vol. 92, pp. 81–90, 2017.
- [4] F. Wang, Z. Chao, L. Huang et al., "Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode," *Cluster Computing*, vol. 7, pp. 1–11, 2017.
- [5] L. Pan, T. Gao, F. Xu, and L. Zhang, "Enhanced robust motion tracking control for 6 degree-of-freedom industrial assembly robot with disturbance adaption," *International Journal of Control, Automation and Systems*, vol. 16, no. 2, pp. 921–928, 2018.
- [6] J. Li, Y. Zhang, and M. Mao, "Continuous and discrete zeroing neural network for different-level dynamic linear system with

- robot manipulator control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, 2018.
- [7] Y. Wang, L. Gu, Y. Xu, and X. Cao, "Practical tracking control of robot manipulators with continuous fractional-order nonsingular terminal sliding mode," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6194–6204, 2016.
- [8] A. Dumlu and K. Erenturk, "Trajectory tracking control for a 3-DOF parallel manipulator using fractional-order $PI^{\lambda}D^{\mu}$ control," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3417–3426, 2014.
- [9] A. Izadbakhsh and S. Khorashadizadeh, "Robust task-space control of robot manipulators using differential equations for uncertainty estimation," *Robotica*, vol. 35, no. 9, pp. 1923–1938, 2017.
- [10] R. S. Mohammad, H. Jafaar, and S. Mahmoodreza, "Robust nonlinear control of robot manipulator with uncertainties in Kinematics, Dynamics and actuator models," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 8, pp. 5487–5489, 2012.
- [11] A. Andreev and O. Peregudova, "Trajectory tracking control for robot manipulators using only position measurements," *International Journal of Control*, vol. 92, no. 7, pp. 1490–1496, 2017.
- [12] P. Farzin, M. Mina, S. Forouzan, N. Iman, and E. Sara, "Design baseline computed torque controller," *International Journal of Engineering (IJE)*, vol. 6, no. 3, pp. 129–141, 2012.
- [13] T. Hongfeng, D. Xiaoqi, and Y. Huizhong, "Robust optimization and application of iterative learning trajectory tracking algorithm for discrete nonlinear systems," *Information & Control*, vol. 45, no. 4, pp. 385–390, 2016.
- [14] X. Jin, "Iterative learning control for non-repetitive trajectory tracking of robot manipulators with joint position constraints and actuator faults," *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 6, pp. 859–875, 2017.
- [15] Y. Hui, R. Chi, B. Huang, and Z. Hou, "Extended state observer-based data-driven iterative learning control for permanent magnet linear motor with initial shifts and disturbances," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2019.
- [16] Y. Tian, Y. Wang, H. Liu, R. Miao, and Z. Hao, "Variable gain iterative learning control with initial error correction," in *Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 412–416, Tianjin, China, August 2019.
- [17] H. Reham, F. Bendary, and K. Elserafi, "Trajectory tracking control for robot manipulator using fractional order-fuzzy-PID controller," *International Journal of Computer Applications*, vol. 134, no. 15, pp. 22–29, 2016.
- [18] T. Ye, Z. Luo, and G. Wang, "Adaptive sliding mode control of robot based on fuzzy neural network," *Journal of Ambient Intelligence and Humanized Computing*, 2020.
- [19] Q. Zhu, S. Li, and D. Zhao, "A universal U-model based control system design," in *Proceedings of the 33rd Chinese Control Conference*, pp. 1839–1844, Nanjing, China, July 2014.
- [20] Q. M. Zhu and L. Z. Guo, "A pole placement controller for non-linear dynamic plants," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 216, no. 6, pp. 467–476, 2002.
- [21] S. S. A. Ali, M. A. Fouad, and S. Muhammad, "Learning feedforward control of MIMO nonlinear systems using U-model," in *Proceedings of the 9th IASTED International Conference on Control and Applications*, pp. 278–283, Innsbruck, Austria, February 2007.
- [22] S. S. A. Ali, F. M. Al-Sunni, M. Shafiq, and J. M. Bakhshwain, "U-model based learning feedforward control of MIMO nonlinear systems," *Electrical Engineering*, vol. 91, no. 8, pp. 405–415, 2010.
- [23] S. S. A. Ali, F. M. AL-Sunni, and M. Shafiq, "U-model based 2DoF multi-variable IMC for improved input-disturbance rejection: a case study on a 2-link robot manipulator," *International Journal of Advanced Robotic Systems*, vol. 8, no. 3, pp. 166–175, 2011.