Research on Adaptive Robust Control of Lifting Robotic Arm Based on Backstepping

Teng Gao* and Lei Han
Beijing University of Posts and Telecommunications, Beijing, China
1098893041@qq.com, hanl@bupt.edu.cn

Abstract—In the process of industrial production, the lifting robotic arm plays a great role in the handling and loading and unloading of heavy loads, and the previous lifting robotic arm is mostly operated manually in the process of operation, but there are safety hazards in the harsh environment, so the research on the control of the lifting robotic arm is of concern. In this paper, for the end trajectory tracking problem of the lifting robotic arm, an adaptive robust controller is designed based on the backstepping, which uses adaptive parameter estimation to compensate for the uncertain disturbances existing in the system model, and the robust feedback term to reduce the influence of disturbances and enhance the robustness of the system. Finally, it is proved by simulation experiments that the controller can effectively achieve the end trajectory tracking and better overcome the uncertainty and random interference of the system.

Keywords- lifting robotic arm; adaptive robust control; backstepping;

1. INTRODUCTION

In modern industrial development, the application of robotic arms can be seen everywhere, the lifting robotic arm system exists nonlinear, time-varying, strong coupling, and model uncertainty, etc. When the state of each joint of the lifting robotic arm (joint angle, angular velocity, angular acceleration) changes, the loads received by each joint also change in real time, which also leads to the coupling of loads due to the connection between the joints. The control accuracy of each joint of the robotic arm affects the trajectory tracking accuracy at the end. All these features bring great disturbance and uncertainty to the position control of the robotic arm, and therefore also put forward higher requirements for the anti-interference and robustness of the trajectory tracking of the robotic arm.

The trajectory tracking control of the robotic arm has always been the focus of attention of many scholars, and thus many effective control strategies have emerged. The dynamics control of the robotic arm mainly includes PID control, robust control, adaptive control, feedback control, as well as predictive control, nonlinear control and so on [1]. Article [2] combined PID control with reinforcement learning to form a PID control strategy with special adaptive iterative learning. Article [3] applied the particle swarm algorithm to the proportional, integral and differential gain rectification of PD and PID parameters, and achieved a better control effect. For the nonlinear complex system, the inverse step design method is a control strategy proposed for the continuous change of the control object as well as the existence of interference, decomposing the complex system, designing the intermediate control quantity and gradually inverting it to the whole system, so as to get the final control rate, which solves the problem of the uncertainty of the parameters of the control object. Article [4] proposed a backstepping adaptive control system for the problem of an incomplete robotic arm system that exists mass and inertia are difficult to determine, and proved the stability by using Lyapunov's theorem.

Adaptive control [5] maintains good control performance in the presence of uncertainty in the system model and changes in the model parameters, while traditional robust control has good robustness but produces jitter which affects the control performance. Article [6] proposed an Arduino-based particle swarm optimization H2 / H∞ robust control method for the presence of uncertainty in the robotic arm system and proved its reliability through simulation and experiment. Article [7] designed a PD adaptive robust iterative learning controller, which can effectively reduce the interference caused by the existence of parameter changes and non-model dynamic characteristics of the controlled robotic arm system. Article [8-11] designed a nonlinear sliding film controller based on a disturbance observer in order to overcome the jitter vibration phenomenon occurring in the tracking process, which effectively improves the anti-interference ability of the robotic arm. In addition, many researchers have also combined neural networks into the control of robotic arms, Article [12] applied adaptive neural networks to the tracking control of robotic arms. Article [13] designed a new valve-controlled memory RBF neural network for the uncertainty of the dynamic characteristics of robotic arms and carried out the simulation of a single-jointed robotic arm, which verified the feasibility of the algorithm. Article [14] proposed a sliding mode control strategy based on a deep convolutional neural network, which combines the advantages of DCNN and FOTSM manifolds and can suppress the jitter phenomenon in the control and improve the control performance.

In this paper, for the end trajectory tracking problem of lifting a robotic arm, an adaptive robust control strategy based on backstepping is adopted to improve the robustness and anti-interference of the control of the robotic arm and to realize the high-precision end trajectory tracking of the robotic arm.

2. MATHEMATICAL MODELLING OF ROBOTIC ARMS

The lifting robot arm used in this paper is a four-joint heavy-duty hydraulic robot arm, the first three joints are rotary joints, and the fourth joint is a translational joint, and its specific configuration is shown in Figure 1 below. Before motion control, the arm is modeled with respect to kinematics and

2835-2823/24/$31.00 ©2024 IEEE
DOI 10.1109/ISSSR61934.2024.00061

dynamics. In order to facilitate the study, the modeling of the hydraulic system is not considered in the modeling process.

Figure 1. Structure of lifting arm

2.1 Kinematic modelling

In this paper, the kinematic modeling of the robotic arm is carried out using the MDH modeling method [15], which is used to solve the relationship between the variables between each joint and the variation of the end position of the robotic arm.

Due to the special configuration of the manipulator, a fixed transformation 1' is added between joint 1 and joint 2. The list of MDH parameters of the robotic arm used in this paper can be seen in Table 1.

Table 1. MDH parameter table

i	θ_i	d_i	a_{i-1}	α_{i-1}
1	q_1	0	0	0
1'	90°	1.548	0	0
2	q_2	0	-0.552	pi/2
3	q_3	-0.19	2.622	0
4	pi	q_4	0.02	pi/2

According to the MDH parameter table of the crane arm, substituting each parameter into the chi-square coordinate transformation matrix, we can get the chi-square coordinate transformation matrices between each neighboring coordinate system ${}^0T_1$, ${}^1T_2$, ${}^2T_3$, ${}^3T_4$, and multiplying each chi-square coordinate transformation matrix by right-handedly in order to get the chi-square transformation matrices from the base to the end coordinate system, which is as follows in Eq. (1).

$$ {}^0T_4 = {}^0T_1\,{}^1T_2\,{}^2T_3\,{}^3T_4 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_x \\ R_{21} & R_{22} & R_{23} & P_y \\ R_{31} & R_{32} & R_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

where $R_{11} = s_1c_{23} - s_{123}$, $R_{12} = -c_1$, $R_{13} = s_{13}c_2 - s_{12}c_3$,
$R_{21} = s_{23}c_1 - c_{123}$, $R_{22} = -s_1$, $R_{23} = c_{12}s_3 + c_{13}s_2$,
$R_{31} = -s_2c_3 + c_2s_3$, $R_{32} = 0$, $R_{33} = s_{23} - c_{23}$,

$$ P_x = s_1(a_3s_{23} - a_3c_{23} - d_4s_2c_3 + d_4s_3c_2 - c_2a_2 - a_{1'}) + d_3c_1 , $$

$$ P_y = c_1(a_3c_{23} - a_3s_{23} + d_4c_2s_3 + d_4s_2c_3 + a_2c_2 + a_1) + s_1d_3 , $$

$$ P_z = s_2(a_3c_3 + d_4s_3 + a_2) + c_2(s_3a_3 - c_3d_4) + d_{1'} , $$

where $c_{ii+1} = \cos\theta_i \cos\theta_{i+1}$, $s_{ii+1} = \sin\theta_i \sin\theta_{i+1}$,
$c_is_i = \cos\theta_i \sin\theta_i$.

2.2. Kinetic modelling

In this paper, the dynamics of the robotic arm is modeled using the Lagrange modeling method [15], and its dynamics model is derived and simplified in the following form:

$$ \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{d}_t \tag{2} $$

Where $\boldsymbol{M}(\boldsymbol{q}) \in R^{4\times4}$ is the inertia matrix of the robotic arm, its ith diagonal term represents the sum of the rotational inertia of the connecting rods from the ith to the fourth rod with respect to the i joint, and the matrix $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in R^{4\times4}$ consists of two parts, the Coriolis force matrix and the centrifugal force matrix, which represents the effect of the uniform motion of the joints on the moments to be exerted by each of the joints. $\boldsymbol{G}(\boldsymbol{q}) \in R^4$. The gravity matrix represents the effect of the robotic arm's own gravity on the moment that needs to be applied to each joint. $\boldsymbol{q} \in R^4$ $\dot{\boldsymbol{q}} \in R^4$ and $\ddot{\boldsymbol{q}} \in R^4$ are the angle, angular velocity and angular acceleration vectors of the joints of the arm, $\boldsymbol{\tau} \in R^4$ represents the output moments of the joints of the arm, and $\boldsymbol{d}_t \in R^4$ is the perturbation vector, which consists of two parts, namely, the friction of the joints and the unknown external disturbances.

In this paper, we use a four-jointed robotic arm with the following properties:

(1). The inertia matrix $\boldsymbol{M}(\boldsymbol{q}) \in R^{4\times4}$ is a symmetric positive definite matrix, and there exists its invertible matrix $\boldsymbol{M}^{-1}(\boldsymbol{q}) \in R^{4\times4}$ with positive numbers $\boldsymbol{M}_{\max}$ and $\boldsymbol{M}_{\min}$ such that its norm is bounded.

$$ \boldsymbol{M}_{\min} \le \left\|\boldsymbol{M}(\boldsymbol{q})\right\| \le \boldsymbol{M}_{\max} \tag{3} $$

(2). $\boldsymbol{M}(q)-2\boldsymbol{C}(q,\dot{q})$ is a skew-symmetric matrix that satisfies the following relationship for any $\boldsymbol{\lambda} \in R^n$:

$$\boldsymbol{\lambda}^T\left(\boldsymbol{M}(\boldsymbol{q})-2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right)\boldsymbol{\lambda}=0 \tag{4}$$

3. DESIGN OF THE CONTROLLER

In the process of controller design, due to the characteristics of nonlinearity and model uncertainty of the four-joint robotic arm system, the backstepping method is used to design the controller, which uses online parameter adaptive updating of the model uncertainty term, and the design of the robust feedback term to compensate for the unknown disturbances.

3.1. Design of the adaptive rate

In adaptive control, reasonable assumptions are given for the parameter uncertainty terms, given as follows:

Assumption: the uncertainty parameter β is bounded and satisfies

$$\beta_{\min} \le \beta \le \beta_{\max} \tag{5}$$

Let the estimate of the uncertainty parameter β be $\hat{\beta}$, then the estimation error of β can be expressed as:

$$\tilde{\beta}=\hat{\beta}-\beta \tag{6}$$

In order to prevent $\hat{\beta}$ from being too large and having an impact on the control rate, as well as to ensure that the parameter estimation is always bounded, it is necessary to design the adaptive rate so that the variation of $\hat{\beta}$ is in the range of $\left[\beta_{\min},\beta_{\max}\right]$, and by using an algorithm with a non-continuous parameter projection mapping, the adaptation can be corrected as follows:

$$\dot{\hat{\beta}}=\boldsymbol{Proj}_{\hat{\beta}}(\psi\delta) \tag{7}$$

$$\boldsymbol{Proj}_{\hat{\beta}}(\bullet)=\begin{cases}0, & \hat{\beta}_i=\beta_{i\max} \text{ and } \bullet>0\\ 0, & \hat{\beta}_i=\beta_{i\min} \text{ and } \bullet<0\\ \bullet, & other\end{cases} \tag{8}$$

In the above equation, ψ is the adaptive learning rate and δ is the parametric adaptive function. With the parametric projection algorithm, the following conditions are guaranteed to hold:

$$\begin{cases}\hat{\beta}\in\Omega_{\beta}=\left\{\hat{\beta}:\beta_{\min}\le\hat{\beta}\le\beta_{\max}\right\}\\ \hat{\beta}^T\left[\psi^{-1}\boldsymbol{Proj}_{\hat{\beta}}(\psi\delta)-\delta\right]\le 0,\forall\delta\end{cases} \tag{9}$$

3.2. Dsignesign of Adaptive Robust Controller based on Backstepping

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate and use specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

In order to make the joint angle q of the robotic arm successfully track the desired joint angle q_{d}, the tracking error of the joint angle is defined as e_1:

$$\boldsymbol{e}_1=\boldsymbol{q}-\boldsymbol{q}_d \tag{10}$$

Differentiating the above equation with respect to time yields the tracking error of the joint angular velocity as:

$$\dot{\boldsymbol{e}}_1=\dot{\boldsymbol{q}}-\dot{\boldsymbol{q}}_d \tag{11}$$

Define the sliding mode function as:

$$\boldsymbol{z}(\mathrm{t})=\boldsymbol{K}_1\boldsymbol{e}_1+\dot{\boldsymbol{e}}_1 \tag{12}$$

where $\boldsymbol{K}_1$ is a 4×4 positive definite diagonal matrix.

The transfer function from $\boldsymbol{e}_1$ to $\boldsymbol{z}(\mathrm{t})$ is:

$$\boldsymbol{G}_{\mathrm{s}}=\frac{\boldsymbol{I}}{\boldsymbol{K}_1+\boldsymbol{s}} \tag{13}$$

The Laws criterion shows that the transfer function is stable when $\boldsymbol{K}_1$ is greater than 0. Therefore, when $\boldsymbol{z}(\mathrm{t})$ converges to 0, the joint angle error $\boldsymbol{e}_1$ will also converge to 0.

Derivation of Eq. (13) can be obtained:

$$\dot{\boldsymbol{z}}(\mathrm{t})=\boldsymbol{K}_1\dot{\boldsymbol{e}}_1+\ddot{\boldsymbol{e}}_1 \tag{14}$$

Combine the kinetic equations to find the kinetic formula for $z(t)$

$$\begin{aligned}\dot{z}(t) &= K_1\dot{e}_1 + \ddot{e}_1 \\ &= K_1\dot{e}_1 + (\ddot{q} - \ddot{q}_d) \\ &= K_1\dot{e}_1 + M(q)^{-1}\left[\tau - C(q,\dot{q})\dot{q} - G(q) - d_t\right] - \ddot{q}_d\end{aligned} \tag{15}$$

Define the tracking error of the joint output torque as e_2 :

$$e_2 = \tau - \tau_d \tag{16}$$

To ensure the stability of the system, the virtual control quantity is taken as the desired output torque τ_d , which has two parts in the form shown in the following Eq. (17):

$$\tau_d = \tau_{dc} + \tau_{df} \tag{17}$$

where τ_{dc} is the model compensation term, which can be obtained by making the dynamics equation of $z(t)$ equal to 0:

$$\tau_{dc} = M(q)(\ddot{q}_d - K_1\dot{e}_1) + C(q,\dot{q})\dot{q} + G(q) - \hat{d}_t \tag{18}$$

where d_t is the parameter uncertainty term of the model, which can be updated by the parameter adaptive rate:

$$\dot{\hat{d}}_t = Proj_{\hat{d}_t}(\psi\delta) \tag{19}$$

included among these $\delta = M(q)^{-1} z(t)$

The τ_{df} in Eq. (17) is the feedback term, and let the feedback term be:

$$\tau_{df} = -M(q) K_2 z(t) \tag{20}$$

The final control rate of the system is obtained as τ_d , where K_1 , K_2 are design parameters, which are positive definite matrices, and ψ_1 is also a design parameter.

3.3. Stability Analysis

To determine the stability of the system, the Lyapunov function is defined as:

$$V = \frac{1}{2} z^T(t) z(t) \tag{21}$$

Find the derivative of the above equation with respect to time:

$$\dot{V} = z^T(t)\dot{z}(t) \tag{22}$$

Substituting Eq. (15) into Eq. (22) can be deformed as:

$$\begin{aligned}\dot{V} = z^T(t)[&K_1 e_1 + M(q)^{-1}(\tau + C(q,\dot{q})\dot{q} \\ &- G(q) - d_t) - \ddot{q}_d]\end{aligned} \tag{23}$$

From Eq. (16) we know that $\tau = \tau_d + e_2$ and substituting it into Eq. (21), we can get:

$$\begin{aligned}\dot{V} &= z^T(t)[K_1 e_1 + M(q)^{-1}(\tau_d + e_2 + C(q,\dot{q})\dot{q} - G(q) - d_t) - \ddot{q}_d] \\ &= z^T(t) M(q)^{-1} e_2 - z^T(t) M(q)^{-1}\tilde{d}_t - z^T(t) K_2 z(t)\end{aligned} \tag{24}$$

Define the augmented Lyapunov function with parameter adaptivity as:

$$V_a = \frac{1}{2} z^T(t) z(t) + \frac{1}{2}\tilde{d}_t^T \psi^{-1}\tilde{d}_t \tag{25}$$

Deriving the above equation with respect to time and substituting Eq. (22) can be obtained:

$$\begin{aligned}\dot{V}_a &= z^T(t)\dot{z}(t) + \tilde{d}_t^T\psi^{-1}\dot{\tilde{d}}_t \\ &= z^T(t) M(q)^{-1} e_2 - z^T(t) K_2 z(t) \\ &\quad - z^T(t) M(q)^{-1}\tilde{d}_t + \tilde{d}_t^T\psi^{-1}\dot{\tilde{d}}_t\end{aligned} \tag{26}$$

When $e_2 = 0$, if the uncertainty in the model is not considered, make $\tilde{d}_t = 0$, $\dot{V}$ constant less than or equal to 0. When the nonlinear uncertainty is considered, substituting the adaptive rate Eq. (19) into the Eq. (25), $\dot{V}_a$ is constantly less than or equal to 0, and the stability of the system is proved.

4. SIMULATION ANALYSIS

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

The lifting robot arm experiences a long time in the lifting process, so this paper intercepts ten seconds of it, in which the first two joints of the robot arm basically remain stationary,

and the main movement is completed by the third and fourth joints.

The physical parameters and inertia tensor of each linkage can be seen in Table 1. During the simulation, the desired trajectory of the robotic arm is as follows. The initial position of each joint of the robotic arm is:

$$\boldsymbol{q}_0 = \left[0.2129, 1.1229, -0.1416, 4.0915\right]$$

and each parameter in the controller is:

$$\boldsymbol{K}_1 = diag\left[200, 200, 200, 200\right], \quad \boldsymbol{K}_2 = diag\left[200, 200, 200, 200\right].$$

The perturbation signal contains two main parts $\boldsymbol{d}_t = \boldsymbol{\tau}_{c1} + \boldsymbol{\tau}_{c2}$, $\boldsymbol{\tau}_{c1}$ for the part related to the joint velocity will be defined as:

$$\boldsymbol{\tau}_{c1} = diag\left[200, 200, 200, 100\right]\dot{\boldsymbol{q}} \tag{27}$$

$$\boldsymbol{\tau}_{c2} = 30\sin\left(\pi t\right) diag\left(\text{rand}(\ ), \text{rand}(\ ), \text{rand}(\ ), \text{rand}(\ )\right) \tag{28}$$

Table 2. Physical parameters of the arm

joint	mass (kg)	shot length (m)	inertial tensor $(kg\cdot m^2)$
1	500	1.2	$diag([230 \quad 230 \quad 45])$
2	800	2.2	$diag([280 \quad 280 \quad 20])$
3	500	1	$diag([121 \quad 121 \quad 10])$
4	500	6	$diag([125q_4^2 \quad 125q_4^2 \quad 0])$

In this paper, the torque-based PD controller is also simulated and compared with the present controller; the control rate of torque-based PD controller is given in the following equation:

$$\boldsymbol{\tau}_{\text{PD}} = \boldsymbol{M}\left(\boldsymbol{q}\right)\left(\ddot{\boldsymbol{q}} + \boldsymbol{K}_d\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}\right) + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}\left(\boldsymbol{q}\right) \tag{29}$$

where $\boldsymbol{K}_p = 10^3\, diag\left(\left[9,9,9,9\right]\right)$

$\boldsymbol{K}_d = diag\left(\left[80,80,80,80\right]\right)$.

Figures 2 and 4 show the comparison curves of the position error of each joint and the comparison curves of the velocity error of the two controllers during the tracking process, respectively. Observing Figures 2 and 4, it can be seen that the torque-based PD controller has a larger overshoot and a slower convergence of the error in the first 1 second of the control, compared with the adaptive robust control. Figures 3 and 5 show the tracking curves of the two controllers with respect to the desired positions and velocities of the joints, and it can be seen that the adaptive robust controller has a higher tracking accuracy and better robustness in the presence of interference.

In order to evaluate the tracking performance of the two controllers, the following four error metrics are selected to measure the response characteristics and tracking effect of the controllers; the following four error metrics are all Cartesian path error metrics. The results are shown in Table 3; it can be seen that the adaptive robust controller based on backstepping is better than the PD controller based on torque. The specific indicators are selected as follows:

(1) Root Mean Square Error (RMS) in the three directions x, y, z characterizes the average tracking accuracy in each direction of the tracking process:

$e_{RMS} = \sqrt{\sum_{i=1}^{N}\left(e_i\right)^2 / N}$ The tracking error at the i sampling point is the position tracking error, where N is the number of sampling points and e_i is the position tracking error at the sampling point.

(2) Cartesian path tracking error, characterizing the comprehensive tracking performance:

$$e_{RSE} = \frac{1}{N}\sum_{i=1}^{n}\sqrt{\left|e_{xi}\right|^2 + \left|e_{yi}\right|^2 + \left|e_{zi}\right|^2}.$$

(3) Maximum error-velocity ratio to characterize the integrated tracking performance of the controller: $\rho = \frac{\left|e_{si}\right|_{\max}}{\left|\dot{s}\right|_{\max}}$, $i = 1,2,\ldots,N$, where e_{si} is the end-of-arm error at the ith sample, and $\dot{s}$ is the end-of-arm velocity.

(4) Maximum tracking error to characterize the transient performance of the controller: $e_{\max} = \max\left\{\left|e_i\right|\right\}$

Table 3. Comparison table of controller indicators

controllers	orientations	RMS	RSE	ρ	$e_{\max}$
$\boldsymbol{\tau}$	x	$8.4101e^{-6}$	$2.7134e^{-6}$	$1.2635e^{-3}$	$2.8703e^{-4}$
	y	$2.6113e^{-6}$			
	z	$1.4869e^{-5}$			
$\boldsymbol{\tau}_{PD}$	x	$1.2333e^{-4}$	$8.6740e^{-5}$	$4.49e^{-2}$	$1.8e^{-3}$
	y	$5.6889e^{-4}$			
	z	$3.9024e^{-4}$			

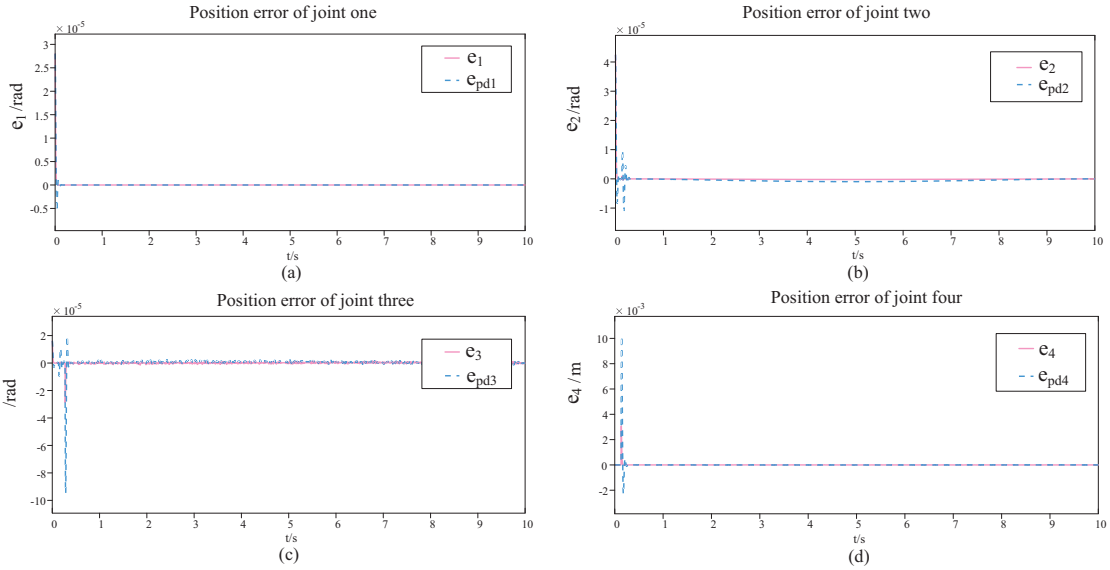

Figure2. Comparison of tracking error of each joint position

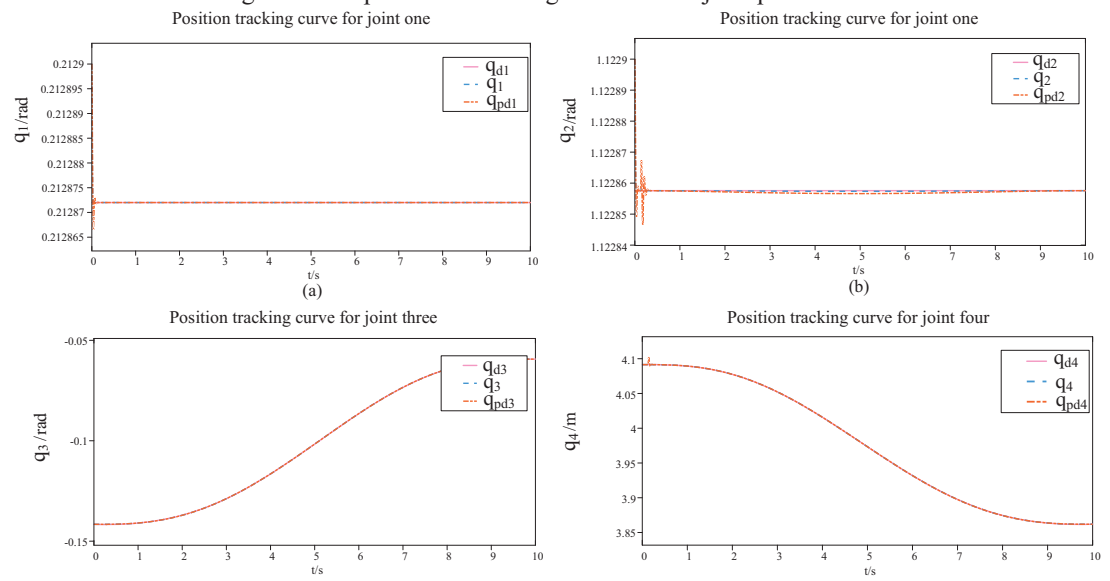

Figure3. Comparison of tracking curves for each joint position

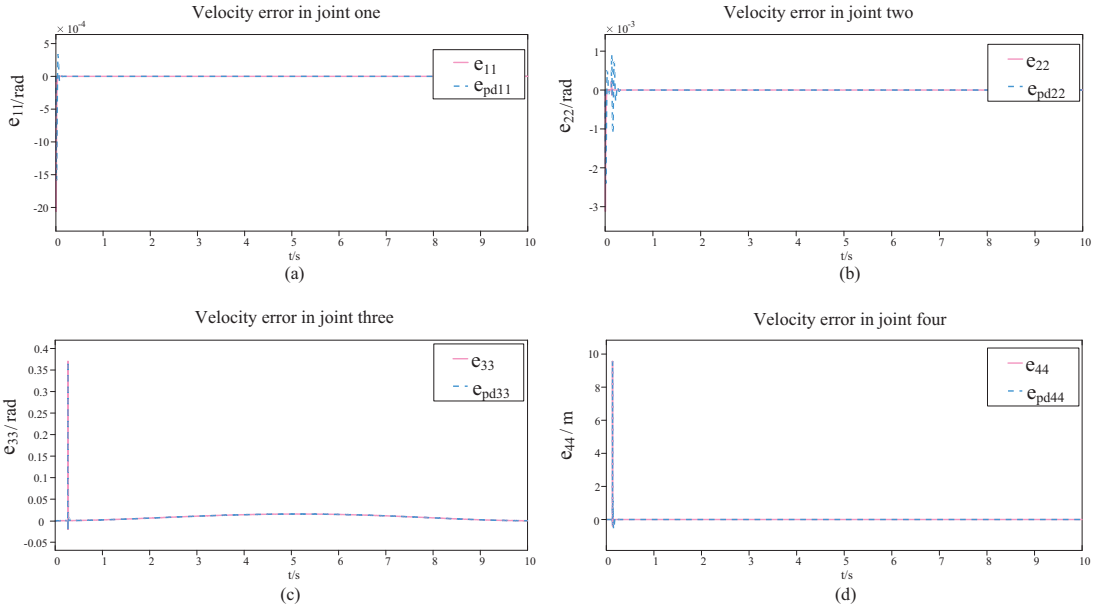

Figure4. Comparison of speed tracking error of each joint

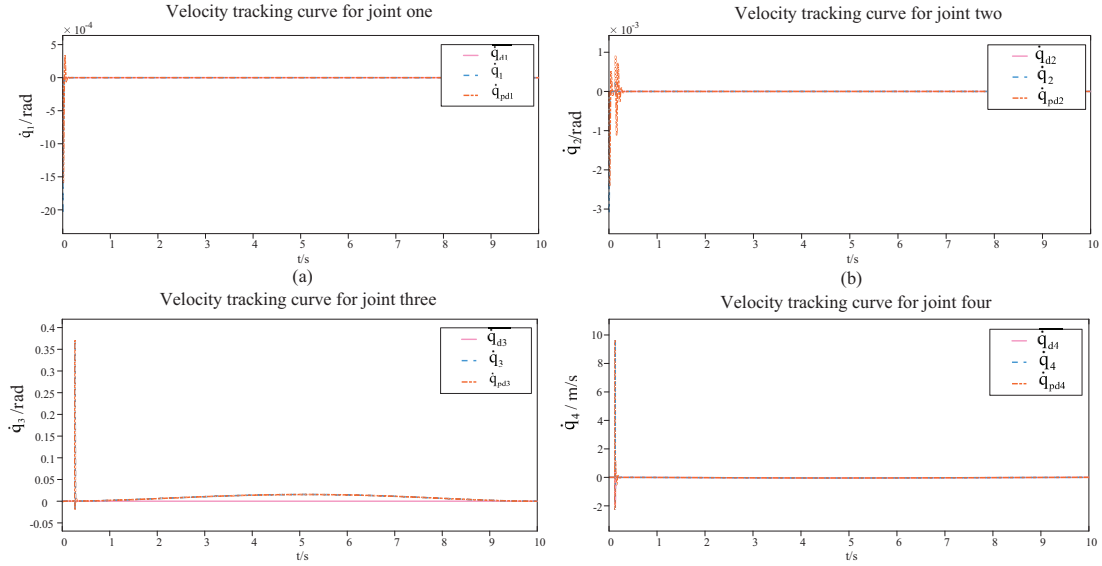

Figure 5. Comparison of velocity tracking curves for each joint

5. SUMMARIES

In this paper, for the end trajectory tracking problem of lifting robotic arm, the kinematics model and dynamics model of the robotic arm are established, and the backstepping method is used to design the controller to solve the model mismatch, the parameter adaptive estimation algorithm is used to compensate for the uncertainty term in the model, and the feedback term is designed to inhibit the influence of the interference, and the control rate is finally obtained by the fusion of the adaptive and robust control through the nonlinear projection. The stability of the system is ensured by selecting appropriate design parameters. By comparing with the PD controller based on moment control, the simulation results show that the adaptive robust controller based on the backstepping method has better performance and can meet the control demand in the lifting process.

REFERENCES

[1] Ghorbanpour A. Cooperative Robot Manipulators Dynamical Modelling and Control: an Overview[J].Dynamics,2023,3(4):820-854.

[2] Hamed N R, Abolfazl Z, Holger V .Actor-critic learning based PID control for robotic manipulators[J].Applied Soft Computing,2024, 151111153-.

[3] Bounouara N, Ghanai M, Chafaa K .Metaheuristic Optimization of PD and PID Controllers for Robotic Manipulators[J].JESA,2021,54(6):835-845.

[4] Mateusz W, Tomasz R, Karol S. An Adaptive Backstepping Control for a Free-Floating Space Manipulator Using a Linearly Parametrized Dynamic Model[J]. Journal of Aerospace Engineering,2024,37(2).

[5] Samah R, Madjid K, Amar R. Adaptive robust nonsingular terminal sliding mode design controller for quadrotor aerial manipulator[J]. TELKOMNIKA (Telecommunication Computing Electronics and Control),2019,17(3):1501-1501.

[6] Petrus S, Bagus M W. Joint control of a robotic arm using particle swarm optimisation based H2/H∞robust control on arduino[J]. TELKOMNIKA (Telecommunication Computing Electronics and Control),2020,18(2):1021-1021.

[7] Wang X, Hairong D, Qiong W. Research of manipulator trajectory tracking based on adaptive robust iterative learning control[J]. Cluster Computing, 2019,22(2s):3079-3086.

[8] An B, Wang B, Wang Y, et al. Adaptive Terminal Sliding Mode Control for Reentry Vehicle Based on Nonlinear Disturbance Observer[J]. IEEE Access, 2019, PP (99): 1-1.DOI:10.1109/ACCESS.2019.2948963.

[9] Jiqing C, Chaoyang Z, Qingsong T, et al. Low Chattering Trajectory Tracking Control of Non-singular Fast Terminal Sliding Mode Based on Disturbance Observer[J]. International Journal of Control, Automation and Systems,2023,21(2):440-451.

[10] Danni S, Jinhui Z, Zhongqi S, et al. Adaptive sliding mode disturbance observer-based composite trajectory tracking control for robot manipulator with prescribed performance[J]. Nonlinear Dynamics,2022,109(4):2693-2704.

[11] Gaorong L, Bingqiang S, Yumei M, et al. Adaptive neural network command filtered backstepping impedance control for uncertain robotic manipulators with disturbance observer[J].Transactions of the Institute of Measurement and Control,2022,44(4):799-808.

[12] He W, Chen Y, Yin Z. Adaptive Neural Network Control of an Uncertain Robot With Full-State Constraints[J]. IEEE Transactions on Cybernetics, 2017, 46(3) :620-629. doi:10.1109/TCYB.2015.2411285.

[13] Yuan X, Liu J, Xie S. Design of the MVT RBF neural network robotic manipulator control system based on model block approximation[J]. Transactions of the Transactions of the Institute of Measurement and Control,2022,44(12):2350-2357.

[14] Mohamed A, Jyotindra N, K. S D. A systematic review on cooperative dual-arm manipulators: modelling, planning, control, and vision strategies[J]. International Journal of Intelligent Robotics and Applications,2023,7(4):683-707.

[15] Huo W. Robot Dynamics and Control [M]. Higher Education Press,2005.