Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 6, No. 2, 2023

Design of Robot End Trajectory Tracking Controller

Xuying Gao*, Geng Liang, Wei Pei+, Xiangyu Feng+

North China Electric Power University, Beijing 100000, China
+ .Are :the .third *authors
*Corresponding Author'

Abstract: The most important point for the manipulator to run quickly and efficiently is that we must explore the trajectory tracking control technology, so we need to master the end trajectory operation law. This topic is mainly to analyze the structural characteristics and working principle of the manipulator, create the control law of the end node system model of the workspace, calculate the actual joint torque, and conduct Kinematics analysis of the end trajectory of the manipulator, Finally, MATLAB simulation was used to verify the designed controller. Studying trajectory tracking control technology will have a positive impact on future life and can be applied practically in production and daily life to meet the needs of modern production.

Keywords: Robotic arm, End trajectory tracking, Control.

1. Introduction

With the rapid development of robotic arms in industry, new tracking and trajectory control methods have emerged one after another. With the increasing application of robotic arms in the industrial field, research has become a crossroads between home and outdoor activities [1]. For example, PID control algorithms and other methods can be applied to simple systems, but they are somewhat difficult to apply to complex systems. Often in practical applications, due to the lack of uniformity and linearity in the control system itself, it is difficult to obtain an accurate mathematical model. This requires the use of correct and accurate control algorithms to control and apply the control system to complete trajectory tracking work. Different control methods are used for different trajectory tracking effects[2].

Starting from analyzing the background and significance of the project, this article explores the current development status of robotic arms, studies the system's pose and coordinate system description, as well as trajectory tracking control methods[3]. The use of Lyapunov controllers is aimed at improving the tracking performance of robots under bounded and time-varying disturbances. The trajectory proposed relaxes the need for nonlinear dynamics knowledge, and based on interference suppression schemes, an asymptotically stable tracking controller in the sense of Lyapunov is constructed. The focus of this article is to use controller design to achieve trajectory tracking[4].

2. Establishment of System Model

We can find Robust control strategies to ensure tracking of the desired position and design techniques for such controllers. Our goal is to design a controller by first establishing a mathematical model and then writing dynamic equations. The trajectory tracking control algorithm proposed in this paper can improve tracking performance in system dynamics, and the motion of the end effector can be tracked to the desired trajectory input channel in a shorter time. Usually, dynamic modeling involves calculating the joint angle. When designing the articulated torque applied by the actuator, the angle and joint angular velocity can be followed. In engineering, it is usually necessary to control the trajectory of the end points. Therefore, it is necessary to establish a dynamic mode of Cartesian coordinates at the end points of each node in the workspace, and design further control laws through mapping relationships to obtain the actual joint torque[5,6].

2.1. Pose and Coordinate System Description

The physical structure of a regular operating system is a mechanical structure, with normal two arm operation of four degrees of freedom. The main servo controller of the control system directly operates the servo motors of the hands and joints to control the movement of the axes and extend the arms. The BOM control algorithm must be used to approximate the system accuracy curve, improve the positioning accuracy and handling speed of joints, complete the precise placement of hands, and give a predetermined angular displacement. The most common schematic diagram of mechanical arm structural parameters is shown in Figure 1.

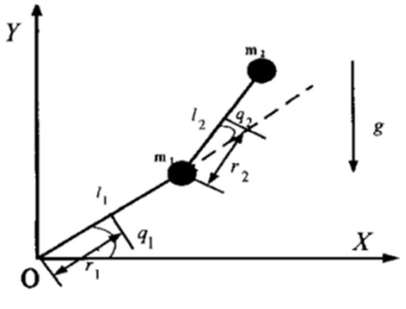

Figure 1. Schematic diagram of structural parameters of a dual joint robotic arm

In system analysis and modeling, in order to clearly express the relationship between the connecting rods in a complex system, the D-H method and coordinate system are usually used to describe the system.

The meaning of coordinate system parameters is as follows:

a_{i-1} is generally the length of the connecting rod of the joint, which is the minimum distance between the two axes of z_i and z_{i-1}; d_i is he distance between the connecting rods, which is the distance between the common perpendicular lines a_i and a_{i-1};

Θ_i is generally the angle between the connecting rods, which is the angle between x_{i-1} and x_i, where the direction of rightward rotation around the z_{i-1} axis is positive;

α_{i-1} torsion angle is the angle between the two axes z_i and z_{i-1}, where the right rotation around x_i is positive. Table 1 shows the parameters of each connecting rod.

Table 1. Robot Link Parameters (Base Coordinate System)

a_{i-1}	α_{i-1}	d_i	θ_i	The range of joint variables	Connecting rod parameters
a_1	-90°	0	90°	- 180° ~ 180°	
a_2	0	0	-90°	- 100° ~ 100°	a_1 =100
a_3	-90°	0	0	-60° ~ 60°	
0	90°	d_4	0	-200° ~ 200°	a_2=705
0	-90°	0	0	- 120° ~ 120°	
0	0	0	0	-400° ~ 400°	a_1=135

2.2. Conversion of Cartesian Coordinates

A dual link planar manipulator can move on the x-y plane. Firstly, model the robotic arm system in Adams. The Adams model has two inputs (actuator torque) and four outputs (first joint angle, second joint angle, and end effector position$x_1 - x_2$component). Then, export the created robotic arm system to Matlab to complete the controller design work in this software on Simulink. The problem of robot inverse Kinematics is to translate the Cartesian coordinates of joint end nodes in the workspace (x_1, x_2) convert to the position of the two joint w angle (q_1, q_2) For the problem, we adopt forward thinking, and Figure 2 shows the Cartesian coordinate system established by a two degree of freedom robotic arm.

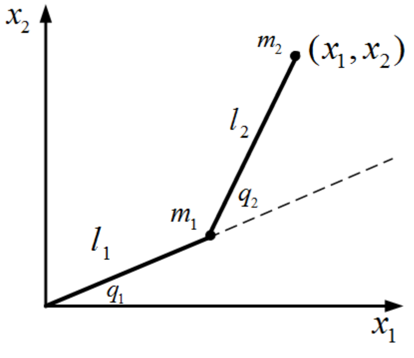

Figure 2. Two degree of freedom robotic arm

$$x_1 = l_1\cos q_1 + l_2\cos(q_1 + q_2) \tag{1}$$

$$x_2 = l_1\sin q_1 + l_2\sin(q_1 + q_2) \tag{2}$$

Adding the square of (1) to the square of equation (2) yields:

$$x_1^2 + x_2^2 = l_1^2 + l_2^2 + 2l_1l_2\cos q_2 \tag{3}$$

Thus, we can obtain:

$$q_2 = \cos^{-1}\left(\frac{x_1^2+x_2^2-l_1^2-l_2^2}{2l_1l_2}\right) \tag{4}$$

take $p_1 = \arctan\frac{x_1}{x_2}$, $p_2 = \arccos\frac{x_1^2+x_2^2+l_1^2-l_2^2}{2l_1\sqrt{x_1^2+x_2^2}}$, then

$$q_1 = \begin{cases} p_1 - p_2, & q_2 > 0 \\ p_1 + p_2, & q_2 \ll 0 \end{cases} \tag{5}$$

x=[x1 x2], q=[q1 q2], then $dx = \frac{\partial y}{\partial x}dq$, $J = \frac{\partial x}{\partial q}$, then $dx = J\,dq$

In the equation, J=$\begin{bmatrix} \frac{\partial x1}{\partial q1} & \frac{\partial x1}{\partial q2} \\ \frac{\partial x2}{\partial q1} & \frac{\partial x2}{\partial q2} \end{bmatrix}$, representing the Jacobian matrix and determinant of the relationship between the end point velocity of the manipulator and the joint angle of the manipulator. It can be obtained from the formula that $\frac{\partial x_1}{\partial q_1} = -l_1\sin q_1 - l_2\sin(q_1+q_2)$, $\frac{\partial x_1}{\partial q_2} = -l_2\sin(q_1+q_2)$, $\frac{\partial x_2}{\partial q_1} = -l_1\cos q_1 + l_2\cos(q_1+q_2)$, $\frac{\partial x_2}{\partial q_2} = l_2\cos(q_1+q_2)$.

2.3. Modeling of Workspaces

For a rigid Robotic arm with n joints, its dynamic characteristics are as follows:

$$D_x(q)\ddot{x} + C_x(q,\dot{q})\dot{x} + G_x(q) = \tau \tag{6}$$

In order to convert the dynamic equation of the set angle into a dynamic equation based on the final position, it is necessary to control the final position. In the static equilibrium state, there is a Linear map relationship between the joint torque and the force transferred to the manipulator. According to the principle of virtual work:

$$F_x = J^{-T}(q)\tau \tag{7}$$

In the workspace, the dynamic equation based on the final position can be written as:

$$D_x(q)\ddot{x} + C_x(q,\dot{q})\dot{x} + G_x(q) = F_x \tag{8}$$

$$d_{xkj}(q) = \theta_{kj}^T \varepsilon_{kj}(q) \tag{9}$$

$$c_{xkj}(q,\dot{q}) = \theta_{kj}^T \varepsilon_{kj}(z) \tag{10}$$

$$g_{xk}(q) = \beta_k^T \eta_k(q) \tag{11}$$

Among, d_{xkj}, c_{xkj}, g_{xk} are matrixElements in D_x, C_x, G_x.

This section introduces the D-H method and coordinate system to describe the system in system analysis and modeling, in order to clearly express the relationship between various connecting rods in a complex system. The problem of robot inverse Kinematics is to translate the Cartesian coordinates of joint end nodes in the workspace (x_1, x_2) Convert to the position of the two joint w angle (q_1, q_2) We adopt forward thinking to establish a Cartesian coordinate system system established by a two degree of freedom manipulator. In order to convert the dynamic equation of the set angle into the dynamic equation based on the final position, it is necessary to control the final position and establish a system model.

3. Controller Design

Assuming the ideal trajectory in the workspace is $x_d(t)$, then $\dot{x}_d(t) - x(t)$ and $\ddot{x}_d(t)$ ideal velocity and acceleration, respectively.

definition

$$e(t) = x_d(t) - x(t) \tag{12}$$

$$\dot{x}_r(t) = \dot{x}_d(t) + \Lambda e(t) \tag{13}$$

$$r(t) = \dot{x}_r(t) - x(t) = e(\dot{t}) \dot{+} \Lambda e(t) \tag{14}$$

Among Λ is a Positive-definite matrix.

The neural network control process is shown in Figure 2. By assigning the initial value of the control object to the Equations of motion, the neural network model will get a new cad state after passing T.Compare this state with the ideal state, calculate the error according to the error model shown in formulas, and use the error correction network weight to obtain new C_xG_x from the systemThen, a neural network model is introduced to calculate the next state, in order to achieve the purpose of control.

The control law is designed by combining feedforward and feedback control, and the design of the control law is as follows:

$$Fx = Kpe + Kde + Dx(q)xd + \dot{C}x(q, \dot{q}\overset{..}{)}xd + \ddot{G}x(q) \tag{15}$$

In the formula, $K_p>0$; $K_d>0$.

At this point, the formula can be written as

$$Dx(q)(\ddot{x}d - \dot{x}) + Cx(q, \dot{q})(\dot{x}d - \dot{x}) + Kd\dot{e} + Kpe = 0 \tag{16}$$

$$Dx(q)\ddot{e} + (Cx(q, \dot{q}) + Kd)\dot{e} + Kpe = 0 \tag{17}$$

The control objective is e and $\dot{e}$ approaching 0.

Take the Lyapunov function

$$V = \frac{1}{2}\dot{e}^T D_x(q)\dot{e} + \frac{1}{2}e^T K_p e \tag{18}$$

According to the positive definiteness of Dx (q) and Kp, if V is globally positive definite, then

$$\dot{V} = \dot{e}^T Dx\ddot{e} + \frac{1}{2}\dot{e}^T \dot{D}x\dot{e} + \dot{e}^T Kpe \tag{19}$$

From oblique symmetry, it can be seen that $\dot{e}^T \dot{D}x\dot{e} = 2\dot{e}^T Cx\dot{e}$,

$$\dot{V} = \dot{e}^T Dx\ddot{e} + \dot{e}^T Cx\dot{e} + \dot{e}^T Kp\dot{e} = \dot{e}^T(Dx\ddot{e} + Cx\dot{e} + Kp\dot{e}) = \dot{e}^T Kp\dot{e} <= 0 \tag{20}$$

That is, starting from any Initial condition, when t tends to infinity, x tends to infinity x_d, $\dot{x}$ approaching $\dot{x}_d$.

This section elaborates on the design of the entire controller from two aspects: controller overview and controller design. When studying the control problem of robotic arm tracking, it is assumed that the navigation path of the robot's target point is always within the workspace at the end of the robotic arm. Starting from their respective initial positions, the robotic arms transmit the collected target points and end effector information in real-time to the controller for processing, and then achieve real-time adjustment of the robotic arm joints through control. After multiple feedbacks and information processing, the end effector of the robotic arm is finally able to track the trajectory of the target point.

4. System Simulation

4.1. Simulation analysis

For the simulation example, we select a two joint Robotic arm model for simulation, and the dynamic equation of the manipulator is

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \tag{21}$$

Where p is the load, P is the parameter vector of the robot itself, and l_1 and l_2 are the lengths of joint 1 and joint 2, respectively.

The actual parameters when processing the arm program are $l_1 = l_1 = 1$

in order to observe whether the controller can reduce load disturbance, the load value changed from 0 to 0.5 at t=4.0.

A trajectory is a circle with a radius of 1.0 and a center located at (1.0 1.0). Under Initial condition, the actuator at the end of the robot arm is located at the center of the circle (1.0 1.0).

The ideal tracking trajectory in space is

$$\begin{cases} X_{d1} = \cos t \\ X_{d2} = \sin t \end{cases} \tag{22}$$

The Cartesian coordinates of the joint end in the workspace must be the angle of the second joint angle (x_1, x_2), and the joints must be connected according to the formula (q_1, q_2) in the first part.

The controlled object and controller take the above equation, and the controller gain is selected as

$$k_p = k_d = \begin{matrix} 30 & 0 \\ 0 & 30 \end{matrix} \tag{23}$$

The joint torque can be calculated from the above equation. The designed controller ensures the stability of the closed-loop system and good performance of the end effector tracking error.

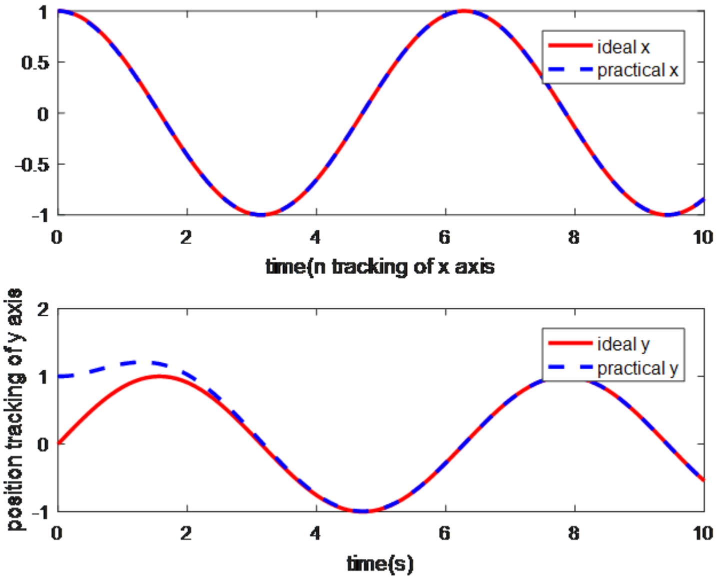

Figure 3. Position tracking of end joint nodes

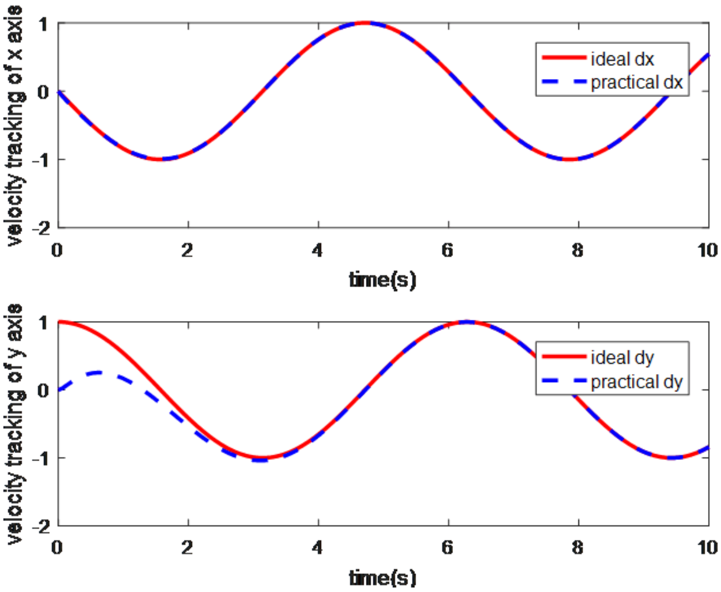

Figure 4. Velocity Tracking of End Joint Nodes

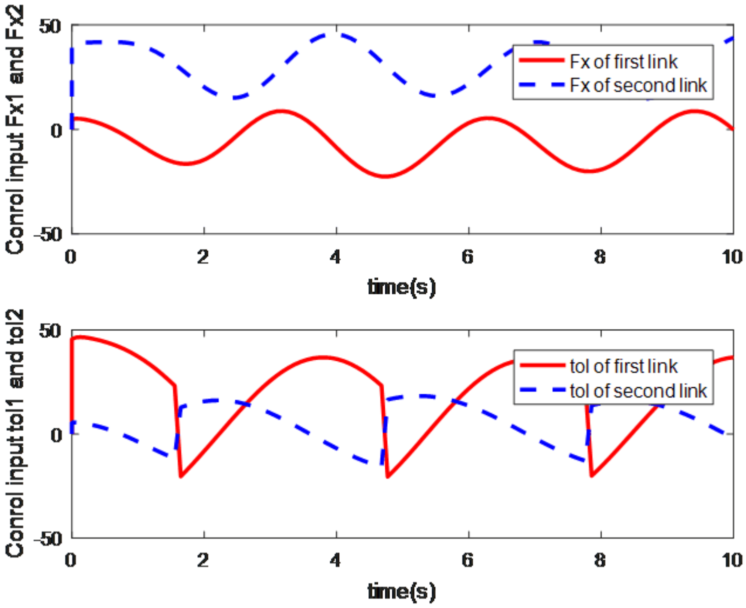

Figure 5. Control Input Fx and τ

As shown in Figure 3, the position tracking of the end joint node is shown. The red line in the figure represents the ideal position, and the blue line represents the position obtained under the neural network model. It can be seen that the neural network model can track the ideal trajectory well in less than 2 seconds; As shown in Figure 4, the velocity tracking of the end joint node is shown. The red line in the figure represents the ideal velocity, and the blue line represents the velocity obtained under the neural network model. Similarly, the neural network model can track the ideal velocity well in less than 2 seconds. From the simulation graph, we can see that utilizing the infinite approximation performance of multi-layer neural networks, with ideal positioning accuracy, the position and velocity of the end joints are tracked. One of the main goals of robot system design is to design structures that can perform multiple tasks.

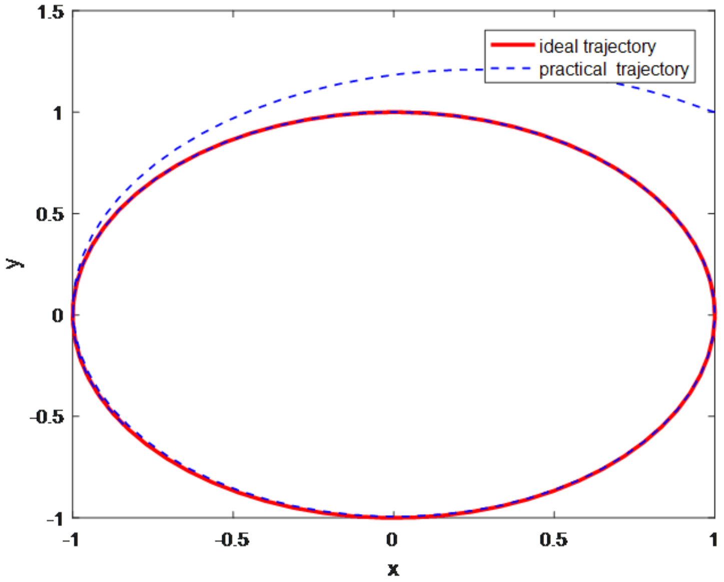

Figure 6. Track tracking effect

Figure 5 shows the control inputs Fx and τ the red represents the control input of the first joint, the blue represents the control input of the second joint, and there is some delay in the control input of the second joint. The overall trend is basically consistent with the system parameters of the first joint; Figure 6 shows the trajectory tracking effect, with red indicating the ideal trajectory and blue indicating the actual trajectory. We found that the negative half axis trajectory of the y-axis completely coincides, but the positive half axis trajectory of the y-axis deviates.

This article explores two degree of freedom robots, but common industrial robots, due to their high degrees of freedom, expand the workspace of end effectors and are widely used in multitasking systems. The simulation results show that using RBF neural network control to compensate for the imprecise problem of the mathematical model can reduce the impact of this problem on the system, while also obtaining asymptotically convergent positioning error, and high-precision tracking of the angular velocity and angular velocity of the two joints. Based on the invariance of parameters rather than the change of the system, the Lyapunov function is selected to obtain the location error of Asymptomatic convergence, and the stability of the control system is proved.

Robot trajectory tracking is a research topic that needs further exploration. The research in this article is only an initial stage, and there are still many issues that need further research. The above content is only a small part of the field of robot trajectory tracking, and there is still a certain gap between it and the application of robotic arm trajectory tracking technology. Establishing a manipulation model and selecting control algorithms are the key to trajectory tracking. The method proposed in this paper has certain advantages compared to traditional methods, and a deeper study is conducted on how to extend it to high degree of freedom operations through new methods, effectively improving the accuracy of the robotic arm and obtaining better trajectories.

5. Conclusion

The mechanical design, Kinematics and dynamics simulation, manufacturing and control of a two degree of freedom manipulator has a high degree of freedom, which can cover a large workspace and perform different tasks in different environments. The results indicate that the proposed mechanical structure can be extended to multi degree of freedom robotic arms. Research has shown that the proposed Lyapunov based controller can suppress interference in a short period of time while tracking a predetermined trajectory. This article considers the effects of friction and saturation, assuming that the connecting rod is rigid and considering the influence of flexibility, and studies the accuracy and accuracy of Lyapunov based controllers.

We describe the working process of a robotic arm as the problem of tracking the robot with the end effector of the robotic arm. Many methods have been proposed to address the purpose of trajectory tracking control. Based on the computational torque theory and RBF neural network, the tracking control method of the joint robot is designed. It is an adaptive Robust control method and Iterative learning control method, which have been recognized and studied by experts at home and abroad in tracking the path of the control system.

Because this control algorithm can play a role of boundary in the control system, the uncertainty is corrected and compensated, So it is usually possible to achieve common high-precision trajectory tracking.

References

[1] Mingyue Cui, Zhaojing Wu. Trajectory tracking of flexible joint manipulators acted by DC motors under random disturbances [J]. Journal of the Franklin Institute, 2019.

[2] Decao Chen, Shuai Li. A recurrent neural network applied to optimal motion control of mobile robots with physical constraints [J]. Applied Soft Computing Journal, 2019.

[3] Hamed Jabbari Asl, Jungwon Yoon. Robust trajectory tracking control of cable driven parallel robots [J]. Online Dynamics, 2017.

[4] Van Cuong P, Nan W Y. Adaptive trajectory tracking neural network control with robust compensator for robot manipulators [J] Neural Computing and Applications, 2016.

[5] Zhang W, Qi N, Yin H. PD Control of Robot Manipulators with Uncertainties Based on Neural Network [C] International Conference on Intelligent Computing Technology & Automation IEEE, 2010.

[6] Ge S S, Hang C C, Woon L C. Adaptive Neural Network Control of Robotic Manipulators [J] World Scientific, London, 1998.