Robot Arm Trajectory Tracking based on adaptive neural Control

Luo Long*
Mechanical & Electrical Department, Guangzhou Institute of Technology, Guangzhou, Guangdong, 510075, P. R. China
*Corresponding author’s e-mail: gzvtcluolong@163.com

Abstract. Control system of robot arm is a multi-variable, strong coupling, and highly nonlinear uncertain system, and trajectory tracking requires the robot manipulator to move according to a desired trajectory that has been given. According to friction and disturbance problem, an adaptive neural network control is proposed. Neural network is used to compensate the dynamic uncertainty of system, and the neural network approximation error and friction and disturbance part are compensated by an parameter adaptive compensation. The simulation results show that this algorithm can improve the effectiveness and accuracy of mechanical arm trajectory tracking.

1. Introduction
Robotics is considered one of the most influential technologies in the world today, and has been included in major development plans in many countries around the world. Scholars from all over the world are making continuous contributions to the development of robot technology and have made a lot of achievements. The basic theories of robot research are also constantly improved. Robot is a kind of mechanical device that can be programmed and perform certain operations and mobile tasks under automatic control. Robot has been gradually applied to all aspects of people's production and life. In the application of robot, the most important actuator is the mechanical arm, which is responsible for completing the vast majority of work tasks of the robot in the practical application. Moreover, the mechanical arm can also be independently applied to the production practice as a kind of robot, playing an important role in the human society. Therefore, the research of robot arm occupies an important position in the field of robot research. It is a major premise of the development of robot technology, which attracts the attention and research of many scholars.

2. The robot arm dynamics model
The manipulator of any robot can be thought of as a series of links connected by joints. A coordinate system is established for each link of the manipulator and homogeneous transformation is used to describe the relative positions and postures of these coordinate systems. Usually, the homogeneous transformation describing the relation between one link and the next link is called A matrix. An A matrix is a homogeneous transformation that describes the relative translation and rotation between the link frames. If A1 represents the position and attitude of the first link with respect to the basis system, and A2 represents the position and attitude of the second link with respect to the first link, then the position and attitude of the second link in the basis system can be given by the matrix product below:

\[ T_2 = A_1A_2 \] (1)
Similarly, for the six-link manipulator, the following T matrix can be obtained:

\[ T_6 = A_1 A_2 A_3 A_4 A_5 A_6 \tag{2} \]

A six-link manipulator can have six degrees of freedom, each link contains a degree of freedom, and can be arbitrary positioning and orientation in its range of motion. Among them, three degrees of freedom are used to specify the position, while the other three degrees of freedom are used to specify the attitude. \( T_6 \) represents the position and attitude of the manipulator. This equation is also known as the robot kinematics equation. The right side of the equation is the product of transformation matrix from fixed reference frame to the coordinate system of hand links. On the left side of the equation, \( T_6 \) represents the product of these matrices, that is, the pose of the robot hand coordinate system relative to the fixed reference system. Analyze the matrix: the first three columns represent the hand posture; the fourth column shows the position of the center of the hand. They can be written as follows:

\[
\begin{bmatrix}
\begin{array}{cccc}
0 & \hat{R}_n & 0 & \hat{P}_n \\
\hat{R}_n & 0 & \hat{P}_n \\
0 & \hat{P}_n & 1
\end{array}
\end{bmatrix} = \begin{bmatrix}
\hat{p}_x & \hat{a}_x & p_x \\
\hat{a}_x & \hat{p}_x & \hat{a}_y \\
\hat{a}_y & \hat{a}_y & \hat{a}_z \\
0 & 0 & 0
\end{bmatrix}
\tag{3}
\]

In the case of robot arm of a connecting rod. On both ends of the connecting rod n relevant section \( n \) and \( n + 1 \). The connecting rod by two geometric parameters: the connecting rod length and Angle of twist. Because at the ends of the connecting rod joints respectively have their own joint axis, normally the two axis is spatial straight lines in different planes, then the two straight lines in different planes of the common normal of long an is the connecting rod length, the Angle between two straight lines in different planes \( n \) is the connecting rod torsion Angle.

On the base of the robot, can start from the first joint transform to the second joint, and then to the third... To the robot's hand, and ultimately to the end of the actuator. If convert each defined, it can be said many transformation matrix.. Total transformation between the base of the robot and the hand is as follows:

\[
^{r}T_H = ^{r}T_1 ^{r}T_2 ^{r}T_3 \cdots ^{r}T_{n} = A_1 A_2 A_3 \cdots A_n \tag{4}
\]

Where \( n \) is the number of joints.

Through right by said four movement of four matrix transformation matrix will be given A four, in turn, the movement of the matrix A. Because all the transformation is relative to the current coordinate system (that is, they are relative to the current local coordinate system to measure and implementation), so all of the matrix are right. And the results are as follows:

\[
^{n}T_{n+1} = A_{n+1} = \text{Rot}(z, {\theta}_{n+1}) \times \text{Tran}(0,0,d_{n+1}) \times \text{Tran}(a_{n+1},0,0) \times \text{Rot}(x, a_{n+1})
\]

Through right by said four movement of four matrix transformation matrix will be given A four, in turn, the movement of the matrix A. Because all the transformation is relative to the current coordinate system (that is, they are relative to the current local coordinate system to measure and implementation), so all of the matrix are right. And the results are as follows:

\[
^{n}T_{n+1} = A_{n+1} = \begin{bmatrix}
C\theta_{n+1} & -S\theta_{n+1} & 0 & 0 \\
S\theta_{n+1} & C\theta_{n+1} & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
C\alpha_{n+1} & -S\alpha_{n+1} & 0 & 0 \\
S\alpha_{n+1} & C\alpha_{n+1} & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

3. Neural Networks Model Reference Adaptive Control

Artificial neural network is the type of model which simulate microscopic structure and function of the human brain. It can simulate the part of the human mind. Its main characteristic is with nonlinear characteristics, adaptability and learning ability. Compared with the conventional control method, neural network control can be as an input to the output mapping, have approximate any nonlinear
mapping and self-learning ability; can solve the unknown nonlinear system control model without accurate mathematics model; can do modelling for the system and control the system; has a strong adaptability.

Adaptive control is produced to control the system with uncertainty. In the actual control process, because of the complexity of modern industrial devices and processes, the mathematical model of the controlled object is often difficult to accurately describe, and the interference of external environment are random and unmeasurable. So the control object often changes at any time or the environment changes, and the traditional control method is difficult to obtain satisfactory control effect. Adaptive control in the operation of the control system, by measuring the collection status, performance and parameters of the object, and on the current operation indicators and compared with the desired index, according to the error of control object to modify the controller itself to adapt to the control object and the dynamic changes of the external disturbance, adjusting the controller structure or parameters in real time, make the whole control system has satisfactory performance.

An adaptive neural network control scheme for a rigid manipulator is presented to solve the problem of trajectory tracking control caused by uncertain factors such as friction and disturbance. For any manipulator with n joint, there are uncertain factors such as friction and disturbance, and the model can be expressed as

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(q) + \tau_d(q,\dot{q},\ddot{q},t) = \tau$$

(6)

Among them $D(q) \in R^{n\times n}$ is the symmetric inertia matrix, $C(q,\dot{q}) \in R^{n\times n}$ is centrifugal force and coriolis force, $G(q) \in R^n$ is the moment of gravity, $q \in R^n$ is the generalized coordinates, $\tau \in R^n$ is the driving torque, among them $F(\dot{q}) \in R^n$ as the friction, it's a function of velocity $\tau_d(q,\dot{q},\ddot{q},t)$ is a time-varying bounded perturbation variable.

In this paper, an adaptive neural network system is constructed to approximate the actual uncertain friction $F_j(\dot{q})$.

Select the following ideal control law:

$$u^*(e, x) = -\alpha_n e_n - h(x)$$

(7)

Among them $x := [x^T, v^T] \in R^{n+1}$, $\alpha_n \in R^+$ is the control gain parameter, $h(\cdot)$, and is the overall uncertainty term determined by the following formula:

$$h(x, v) := (f(x) - v) / g(x)$$

(8)

So $y$ global asymptotic tracing $y_d$.

Where the optimal parameter vector $W^*$ are defined as follows

$$W^* := \arg \min_{W \in \Omega} \left\{ \sup_{x \in \Omega} |h(x) - \hat{h}(x) | \right\}.$$

(9)

The adaptive law of parameters is designed as follows:

$$\dot{\hat{W}} = \begin{cases} \gamma_e e_n \Phi(x)_{ij} f \hat{W} < C_e \\
\alpha \|\hat{W}\| = M_v \text{ and } e_n \Phi(x) \leq 0; \\
\gamma e_n e_n \Phi(x) - \Gamma e_n \hat{W} \hat{W}^T \Phi(x) \\
\text{if } (\|\hat{W}\| = M_v \text{ and } e_n \Phi(x) > 0). \end{cases}$$

(10)
The closed loop system can achieve partial asymptotic stability. That is,
\[ \lim_{t \to \infty} \| e(t) \| = 0. \]

4. Simulation discussion
The mechanical arm is selected as a double joint, its dynamic model is:
\[ M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F(q) + \tau_d = \tau \]

\[ D_{11}(q_2) = (m_1 + m_2) r_1^2 + m_2 r_2 \cos(q_2) \]
\[ D_{21}(q_2) = D_{22}(q_2) = m_2 r_2^2 + m_2 r_2 \cos(q_2) \]
\[ C_{12}(q_2) = m_1 r_2 \sin(q_2) \]
\[ g_1(q_1, q_2) = (m_1 + m_2) r_1 \cos(q_1) + m_2 r_2 \cos(q_1 + q_2) \]
\[ g_2(q_1, q_2) = m_2 r_2 \cos(q_1 + q_2) \]
\[ F_1(\dot{q}_1) = 10 \dot{q}_1 + 3 \text{sgn}(\dot{q}_1) \]
\[ F_2(\dot{q}_2) = 10 \dot{q}_2 + \text{sgn}(\dot{q}_2) \]
\[ \tau_d = 0.05 \sin(20t), \tau_d = 0.1 \sin(20t) \]

(11)

Where \( r_1 = 1 \text{m} \), \( r_2 = 0.8 \text{m} \), \( m_1 = 1.0 \text{kg} \), \( m_2 = 1.5 \text{kg} \); \( \lambda_1 = \lambda_2 = 0.0001 \); \( K = 20 I(2) \); \( \rho = \text{diag}(0.05, 0.1) \); \( \theta_1 \) and \( \theta_2 \) are selected:
\[ \{0.9; 0.9; 0.9; 0.9; -\frac{1.3\pi}{6}; -\frac{1.3\pi}{12}; 0; \frac{1.3\pi}{12}; \frac{1.3\pi}{6}; 42\pi; 42\pi; 42\pi; 42\pi; 42\pi\} \]

Using Simulink and S function to design the control system, the system simulation results are shown below:

Figure 1 speed tracking
The simulation case shows that the adaptive neural control method adopted in this paper is effective and feasible, and has a certain reference value in mechanical arm joint control. In the simulation, figure shows the mechanical arm joint Angle on the desired speed and position tracking performance. So the adaptive neural control is good intelligent control used in the mechanical arm.

Acknowledgments
This work was financially supported by project of Education department of Guangdong province science and technology (2017GKTSCX048).The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References
[1] Liu, H., Zhang, T. Adaptive neural network finite-time control for uncertain robotic manipulators[J]. Journal of Intelligent and Robotic Systems: Theory and Applications, 2014, 3(75):363-377.
[2] Liu jinkun. Robot control system design and MATLAB simulation the basic design method [M]. Tsinghua university press. 2016
[3] Sun C, He W, Hong J. Neural Network Control of a Flexible Robotic Manipulator Using the Lumped Spring-Mass Model[J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2017, 47(8):1863-1874.
[4] He W, Dong Y, Sun C. Adaptive Neural Impedance Control of a Robotic Manipulator With Input Saturation[J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2016, 46(3):334-344.
[5] M. M. Fateh1, S. M. Ahmadi and S. Khorashadizadeh. Adaptive RBF network control for robot manipulators[J]. Journal of AI and Data Mining, 2014, 2(2): 159-166.