

Accuracy Improvement of Robot Arm Trajectory based on Adaptive Neural Network Algorithm

Peibo Li¹, Peixing Li², Chen Yanpeng³

1.School of Mechanical Engineering, Donghua University, Shanghai, 201620, China

2.School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai, 201620, China

3.Shanghai Institute of Spacecraft Equipment, Shanghai, 200240, China

Abstract

An adaptive neural network control method was proposed to solve the problems such as unstable motion and large trajectory tracking error when the robot arm was disturbed by the external environment. The dynamic equations of the manipulator were given and the dynamic characteristics of the manipulator were studied by using the positive feedback neural network. Then the adaptive neural network control system was designed, and the stability and convergence of the closed-loop system were proved by the Lyapunov function. Later, the model diagram of the robot arm was established, and the dynamics parameters of the manipulator were simulated by MATLAB /Simulink software. At the same time, they were compared with the simulation results of the PID control system for analysis. The simulation results showed that the trajectory tracking error and input torque fluctuation were smaller when the trajectory of the robot arm was disturbed by the external world. When adopting the control method of the adaptive neural network, the robot arm could improve the control precision of the trajectory, thus reducing the jitter of the robot arm motion.

Key words: neural network; Mechanical arm; Motion Trajectory, Accuracy Improvement

I. Introduction

The arm of a robot is called a robot arm, which is a mechanical device performing tasks automatically. The robot arm can move according to the program trajectory set by people, so as to complete the grasping and carrying of the object. Its task is to replace the heavy work of human beings and achieve the mechanization and automation of industry. Especially in the dangerous working environment, the robot arm can protect people's personal safety and reduce casualty accidents. Therefore, robot arms are widely used in various fields such as medical treatment, aviation and ocean [1-2].

The trajectory control of the robot arm is very important when it is performing high-precision tasks. To reduce the error between the actual trajectory and the theoretical trajectory of the robot arm, the control method of the robot arm was studied by researchers. For example, Literature studied the motion control process of the robot arm on the production line. The PID control method was adopted in the control process, and the control precision and response speed of the robot arm were improved through the multi-processor cooperative control. Literature studied the fuzzy PD control method of the robot arm, proposed a parallel fuzzy controller, deduced the fuzzy PD control method, and then simulated the motion trajectory tracking task of the robot arm's fuzzy PD control. This control method can improve the motion trajectory tracking accuracy of the robot arm. In Literature, according to the conservation relationship of the moment of momentum of the system, the Lagrange method was used to establish the system dynamics equation of the robot arm, and then an adaptive fuzzy compensation controller was introduced at the end of the robot arm [3-5]. The system simulation proved that the control method could improve the tracking accuracy of the motion trajectory of the robot arm. Although the tracking error of

the robot has been reduced in the previous research, the tracking error of the manipulator was larger in the case of external disturbance. In this paper, the adaptive feedforward neural network control method was introduced at the end of the robot arm, and the stability and convergence of the closed-loop system of the neural network were proved by using the Lyapunov function. The trajectory tracking error and input torque of the neural network control method were simulated in MATLAB /Simulink environment. Meanwhile, it was compared with the PID control method. The simulation curve showed that when the robot arm was disturbed by the external environment, the trajectory tracking accuracy of the adaptive feedforward neural network control method was higher, the manipulator dither phenomenon was weakened, the system movement was relatively stable, so it could meet the requirements for the high-precision trajectory tracking task of the robot arm[6].

II. Dynamic characteristics of robot arm

For the N-DOM robot arm, the connection position variable was denoted by $q = [q_1, q_2, \dots, q_n]^T$. Suppose the research object could be represented by m variables $x = [x_1, x_2, \dots, x_m]^T$ ($m < n$), and at the same time, $x = f(q)$, f was the kinematics positive solution equation. After quadratic differentiation of x , the follow formula can be obtained:

$$\ddot{x} = J'(q)\dot{q}' + J(q)\ddot{q}'' \tag{1}$$

Where $J(q) = \partial f(q) / \partial q$, $J(q) \in R^{m \times n}$ represents the Jacobian matrix of the robot arm.

The pseudo inverse matrix of the Jacobian matrix $J(q)$ of the robot arm (expressed by $J^+(q)$) was defined as follows :

$$J^+ = J^T(JJ^T)^{-1} \tag{2}$$

Where $JJ^+ = I_m$, $I_m \in R^{m \times m}$ is the unit matrix.

For a robot arm with N coupling pair and rotary direct drive, it can be represented with the following relational expression[7-8]:

System Representation :

$$M(q)\ddot{q}'' + V_m(q, \dot{q}') + G(q) + F(\dot{q}') = \tau \tag{3}$$

Where $M(q) \in R^{n \times n}$ is the inertial matrix, $V_m(q, \dot{q}')$ is the centripetal Coriolis force matrix, $G(q) \in R^n$ is the gravity matrix, $F(\dot{q}')$ is the Friction matrix, and $\tau \in R^n$ is the Torque input vector.

III. Feedforward neural network

The double-layer feed-forward neural network containing n input units, m output units and N hidden layer units is shown in Fig. 1

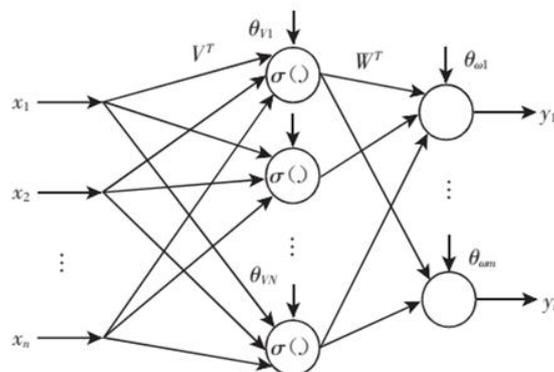


Fig.1 Feedforward Neural Network

The output vector z can be determined by the input vector according the following formula :

$$z_i = \sum_{j=1}^N [\omega_{ij} \sigma(\sum_{k=1}^n v_{jk} y_k + \theta_{vj}) + \theta_{\omega i}], i = 1, 2, \dots, m \quad (4)$$

Where $\sigma()$ is the activation function of hidden layer neurons, $\sigma(y) = 1/(1 + e^{-y})$; v_{jk} is the interconnection weight of the input layer to the hidden layer; ω_{ij} is the interconnection weight of the hidden layer to the output layer; θ_{vj} 、 $\theta_{\omega i}$ are the deviation weights.

The weight values of the neural network v_{jk} 、 ω_{ij} were substituted into the weight matrices, respectively, and the corresponding vectors of the neural network equation could be obtained as follows:

$$z = W^T \sigma(V^T y) \quad (5)$$

Where $\sigma(x) = [\sigma(x_1), \sigma(x_2), \dots, \sigma(x_n)]^T$ is the activation function vector, and $x \in R^n$.

The deviation weight value was taken as the first column parameter of the weight matrix. To accommodate the deviation weight value, the vectors y and $\sigma()$ should be expanded accordingly.

Property 1: Set the smoothing function from R^n to R^m as $h(y)$, and $U_y \subseteq R^n$. Under the condition that $y \in U_y$ and, there were N hidden layer neurons, weight matrix W and weight matrix V , and the relationship was $h(y) = W^T \sigma(V^T y) + \varepsilon$, then the estimated value of the smoothing function $h(y)$ could be expressed as $\hat{h}(y) = \hat{W}^T \sigma(\hat{V}^T y)$, where \hat{W} and \hat{V} are the ideal neuron weight estimation matrices obtained through the online weight adjustment algorithm.

IV. Adaptive neural network controller

4.1 Error analysis

By studying the input parameters of the control torque, the actual operation trajectory of the end-manipulator of the robot could match the expected operation trajectory as much as possible. Therefore, the input value of the controller should make use of the redundancy of the degree of freedom of the robot arm effectively to realize the effective tracking of the program subtasks. The operating space tracking error $e(t) \in R^m$ can be expressed as follows:

$$e(t) = x_d - x \quad (6)$$

Where $x_d \in R^m$ is the expected operation trajectory in the operation space.

The defined sub-task tracking error $e_N(t) \in R^n$ was expressed as follows:

$$e_N(t) = (I_n - J^+ D)(g - q') \quad (7)$$

Where $g(\cdot) \in R^n$ was established according to the sub-task control objective.

In order to provide incentives for the determination of program subtask control objectives, the derivative of time was conducted to Formula (6). After simplification, the closed-loop operating space trajectory tracking error system could be ultimately written as follows:

$$e' = -\alpha e + J r \quad (8)$$

Where $\alpha \in R^n$ is the positive definite gain diagonal matrix, and $r(t) \in R^n$; r is the tracking error signal after screening, and $r = J^+(x_d + \alpha e) + (I_n - J^+ D)g - q'$.

Formula (7) was used to set the value of the control input parameters to ensure that the error of the operating space and the tracking error after screening were adjusted within the specified range. The above requirements could be achieved by using the properties of the inverse matrix, and the following equation can be obtained:

$$e_N = (I_n - J^+)r \tag{9}$$

As can be seen from Formula (9), $e_N(t)$ could be adjusted through adjusting $r(t)$, thus realizing the subtask control of the program.

4.2 Neural network control

The design of the controller based on adaptive feedforward neural network was based on the following assumptions:

Hypothesis 1: $x_d, \dot{x}_d, \ddot{x}_d, g(t)$ and $g'(t)$ were both bounded functions of time.

Hypothesis 2: all dynamic and kinematic function parameters (e.g., $M(q), V_m(q, \dot{q}), G(q), J(q)$ and $J^+(q)$) were bounded under all conditions.

In order to express the symbols conveniently, the weight value matrix of neural network was defined as $Z \equiv \text{diag}\{W, V\}$, and the weight estimation error was expressed as $\tilde{W} = W - \hat{W}, \tilde{V} = V - \hat{V}$ and $\tilde{Z} = Z - \hat{Z}$. The weight values of ideal neural networks were bounded and satisfied $\|Z\|_F \leq Z_M$ and $Z_F^2 = \text{tr}(Z^T Z)$ under condition that Z_M had been unknown. At the same time, under condition that y had been unknown, the hidden layer output value error $\tilde{\sigma} = \sigma(V^T y) - \sigma(\hat{V}^T y)$.

As for the tracking error of the robot arm after screening, its dynamic characteristics could be written as:

$$M\dot{r}' = -V_m r - Kr + \hat{W}^T \hat{\sigma} \tilde{V}^T y + \tilde{W}^T \hat{\sigma} + \omega - J^T e \tag{10}$$

Where $\omega = \tilde{W}^T \hat{\sigma}' \tilde{V}^T y + W^T O(\tilde{V}^T y) + \varepsilon$, K is the positive definite gain matrix.

Then the boundary value of the disturbance term $\omega(t)$ could be determined by the normal numbers c_0, c_1 and c_2 . The weight estimation \hat{W} and \hat{V} were both bounded and the operation space and the sub-task tracking error could be adjusted by the weight value to be arbitrarily small.

Proof: For the following candidate item of Lyapunov function :

$$L = \frac{1}{2} e^T e + \frac{1}{2} r^T M r + \frac{1}{2} \text{tr}(\tilde{W}^T F_\omega^{-1} \tilde{W}) + \frac{1}{2} \text{tr}(\tilde{V}^T G_v^{-1} \tilde{V}) \tag{11}$$

Taking the derivative of time for Lyapunov function, and combining with the equation $\tilde{W}' = -\hat{W}', \tilde{V}' = -\hat{V}'$ and the adaptive learning rule, the following formula could be obtained:

$$L' = -e^T \alpha e - r^T K r \tag{12}$$

The stability of Lyapunov function was determined by $L > 0$ and $L' \leq 0$, so as to ensure that $r(t), \hat{V}$ and \tilde{W} (and the corresponding \tilde{W}, \hat{V}) were bounded.

The boundedness of the $r(t)$ function guaranteed the boundedness of the functions $e(t)$ and $e'(t)$, because of the boundedness of x, \dot{x} and \ddot{x} reflected by the boundedness of the expected trajectory. The boundedness of signals represented on the right side of Equation (10) verified the boundedness of r' and L' as well as the uniform continuity of L' . Since the change of L' with time tended to be zero, so did $e(t)$ and $r(t)$. Therefore, it could be concluded from Equation (9) that $e_N(t)$ also tended to be zero.

V. Simulation and analysis

The simulation verification was carried out for the established model diagram of the robot arm, as shown in Figure 2.

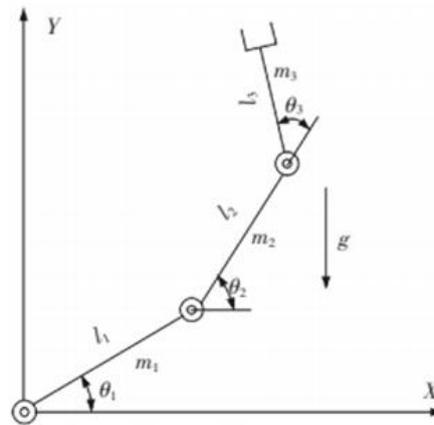


Fig.2 The Model of Robot Arm

In order to verify the trajectory tracking effect of the robot arm controlled by the adaptive neural network, the trajectory tracking error of the robot arm was simulated in Matlab/Simulink environment. The trajectory tracking error of robot arm 3 was simulated and verified. The simulation results of robot arms 1 and 2 were similar to those of robot arm 3, so the simulation verification wouldn't be carried out here. The simulation parameters were set as follows: The expected motion trajectory of robot arm 3 was $\theta_3 = 0.4\cos(2\pi t)$, initial condition $\theta(0) = [000]^T$, control parameter $K = \text{diag}(40,40,40)$, disturbance parameter $\tau = 20\cos(\pi t)$, and connecting rod parameters of the robot arm $l_1 = 0.62\text{m}$, $l_2 = 0.41\text{m}$, $l_3 = 0.34\text{m}$, $m_1 = 3.5$, $m_2 = 2.5\text{kg}$, $m_3 = 2.0\text{kg}$, $g = 9.82\text{m/s}^2$, $t = 2\text{s}$. In the case of no external disturbance, the motion trajectory tracking simulation results of the robot arm 3 are shown in Figure 3, and its input torque simulation results are shown in Figure 4. In the case of external disturbance, the motion trajectory tracking simulation results of the robot arm 3 are shown in Figure 5, and its input torque simulation results are shown in Figure 6.

It can be seen from Figs. 3 and 4 that, in the absence of external disturbance, both adaptive neural network control and PID control could effectively track the desired trajectory of the robot arm. The input torque of PID control showed a large fluctuation range and serious jitter phenomenon. Therefore, the adaptive neural network control method was better. As shown in Figs. 5 and 6, the adaptive neural network control could track the desired trajectory of the robot arm well under the circumstance of external disturbance, and the input torque fluctuation amplitude was relatively small. However, the expected trajectory tracking error of PID control for the robot arm was relatively large, the input torque fluctuation was larger, and the jitter was more serious. By comprehensive comparison, the adaptive neural network control could complete the trajectory tracking task of the robot arm well.

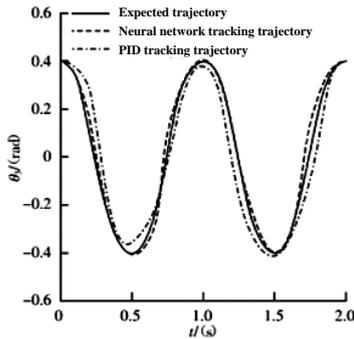


Fig.3 The Third Robot Arm Trajectory (Without External Disturbance)

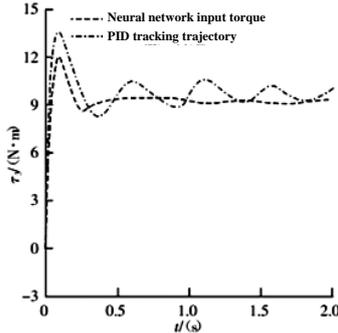


Fig.4 The Third Robot Arm Torque Control (Without External Disturbance)

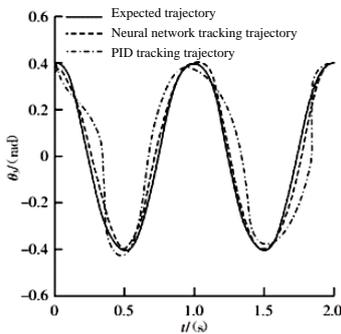


Fig.5 The Third Robot Arm Trajectory (External Disturbance)

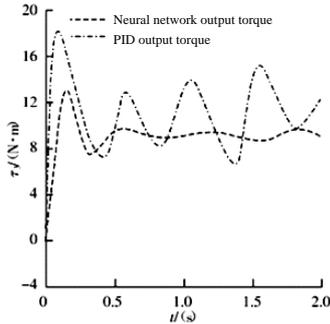


Fig.6 The Third Robot Arm Torque Control (External Disturbance)

VI. conclusion

The neural network control problem of the robot arm trajectory tracking was studied here. The neural network controller can effectively realize the trajectory tracking of the end-effector and the subtask tracking of the program. A feedforward neural network was used to realize the task of learning the unknown dynamic characteristics of the robot arm without prior learning conditions. Then the stability of the system was verified by the Lyapunov function obtained by the weight value matrix. The trajectory tracking error of the robot arm was simulated and verified in Matlab/Simulink environment. The simulation results showed that the adaptive neural network control method could not only accurately realize the trajectory tracking task of the robot arm but also effectively weaken the chattering phenomena of the robot arm under the external disturbance.

Reference

- [1] Cheng Xiaoning. Optimum control of robot arm trajectory based on visual feedback [D]. Zhejiang University.
- [2] Gu Yikun, Ni Fenglei, Liu Hong. RENN-based adaptive dynamic surface control of flexible joint robot arm [J]. Control and decision-making, 2011, 26 (12): 1783-1790.
- [3] Wang Sanxiu, Yu Li, Xu Jianming, etc. Robust adaptive trajectory tracking control for robotic arm [J]. Control Engineering, 2015, 22 (002): 241-245.
- [4] Celepin, Chen Zili, Qi Xiaohui. Robot adaptive inverse sliding mode control based on nonlinear disturbance observer [J]. Information and control, 2013, 42(4): 470-477.
- [5] Qing Guanghui, Qiu Jiajun, Liu Yanhong. Adaptive output feedback control design of robot arm based on neural network [J]. Applied Mathematics and Mechanics, 2013, 28 (1): 427-433.
- [6] Nai Yongqiang, Li Jun. Robot arm adaptive neural control based on limit learning machine [J]. Information and control, 2015, 44 (3): 257-262.
- [7] [7] Comrade Science, Shi Lichen, Zhang Ligang. Research on Track Control of Multi-Degree-of-Freedom Mechanical Arm of Railway Coal Sampler [J]. Mechanical Design, 2012 (09): 85-88.
- [8] Ying Wenjie, Xu Kai, Xu Shaoping. Multi-objective Programming of Badminton Robot Manipulator Track [J]. Computer Engineering and Application, 2017, 53 (003): 258-265.
- [9] Teng Juyuan, Xu Hongbin, Wang Yi, etc. Planning and designing simulation of trajectory of harvesting robot arm [J]. Computer simulation, 2017, 34 (004): 362-367.