Decentralized control of robot joints based on neural network observer

Yuedou Pan¹ *, Guoqing Cai²

¹ University of Science and Technology Beijing, USTB, BeiJing, 100000, China
² University of Science and Technology Beijing, USTB, BeiJing, 100000, China

Abstract. In actual work, the system parameters of the robot joints will change in real time or cannot be measured, and the coupling relationship between the various subsystems and the existence of modeling errors make the system model difficult to determine. Based on such problems, a neural network observation is proposed Decentralized control method for robot joints of the robot. In the actual control of the manipulator, firstly, the model of each joint subsystem is established by the decentralized control theory, and the nonlinear function approximation ability of the neural network is used to approach the uncertain part of the manipulator subsystem online through input and output data. The design observer can estimate the state of the system, and use the estimated state to design a sliding mode controller to dynamically estimate and compensate the unknown model dynamics of each independent joint, and realize the self-control of the system when the speed information and model information are unknown. Adaptive control greatly enhances the robustness and adaptability of the robotic arm system. Finally, the stability criterion of the neural network observer and the sliding mode controller is given by the Lyapunov function method, and the simulation results prove the effectiveness of the design method.

1 Introduction

Robot joints are now widely used in a series of environments such as high-risk operations, military, medical, service, agricultural production and processing [1,2]. The robot joint system is a multivariable, nonlinear, and strongly coupled system [3]. In actual work, many parameters and states of the robot joint system (inertial matrix, centrifugal force and Coriolis force terms, joint angular velocity, etc.) will change or unable to measure, there is a coupling relationship between the various subsystems and is susceptible to external interference, and the system is affected by modeling errors, which makes it difficult to establish a model of the system [4]. Traditional computational torque control (CTC) and proportional-integral-derivative (PID) control methods have been difficult to meet the needs of high precision and high stability of the system, and the control algorithm with poor robustness will make the system move in an unpredictable direction Development, and even accidents, have higher requirements for the adaptability and robustness of the controller. Therefore, the control of the robot joint system under the condition of system state (position and angular velocity) and unknown model is an urgent problem to be solved.

Considering that neural networks have a good approximation effect on nonlinear functions [5], many researchers are committed to combining neural networks with modern control methods to solve the problem of excessive dependence on models and system state information. In the method of designing the controller, literature [6] and [7] combined neural network with sliding mode variable structure control, and designed different sliding mode reaching laws to improve the performance of traditional sliding mode control and speed up the trajectory tracking effect. References [8] and [9] use neural networks to combine with adaptive control, use neural networks to estimate system uncertainties, and design adaptive neural network controllers.

In view of the unknown system dynamics of the robot, another method is to design the state observer of the system so that the error between the state of the observer and the system state approaches zero within a finite time, so as to use the state of the state observer to estimate the system state. Literature [10] designed an extended state observer to dynamically estimate and compensate the uncertain state and disturbance of the system, and adopt nonlinear state feedback control to improve the control performance of the system; in order to observe the disturbance signal, the literature [11] designed an adaptive backstepping sliding mode control strategy based on nonlinear disturbance observer. Literature [12] proposed an improved design of nonlinear disturbance observer for the limited model and control input chattering, but the control method remained unchanged.

Aiming at the good approximation effect of the neural network to the nonlinear function, the uncertain part of the system can be approximated online, and the
observer can estimate the state of the system at the same time, and the neural network adaptive observer is designed to meet the constraints in the cross-linking term. In the case of, after completing the state estimation, the sliding mode variable structure control method is used to design the controller, which can control the system when the system speed is unknown and the model is unknown.

The innovation of this paper is to combine decentralized control theory with neural network methods to design neural network observers for each sub-joint. As shown in Figure 1, the centrifugal force and Coriolis force term, gravity term, and friction force are separated from the system dynamics equation (1), and the associated term is to simplify the model and use the sliding mode method to control the robot according to the estimated state of the system. Joint tracking control.

![Figure 1. Control principle block diagram](https://doi.org/10.1051/e3sconf/202123301019)

### 2 System description

According to the energy relation of the Lagrange kinematics equation, the dynamic model of robotic manipulators with the following expression of the form:

$$\begin{align*}
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + P(\dot{q}) + H(q) &= u \\
y &= q
\end{align*}$$

(1)

Where \(q, \dot{q} \in \mathbb{R}^n\) are the angle and angular velocity of the joint, \(u \in \mathbb{R}^n\) is the control input, \(M(q) \in \mathbb{R}^{n \times n}\) is the inertia matrix of the robot manipulator, \(C(q, \dot{q})\dot{q} \in \mathbb{R}^n\) represents the Coriolis and centripetal term, \(P(\dot{q}) \in \mathbb{R}^n\) is the surface friction and \(H(q) \in \mathbb{R}^n\) denotes the gravitational torque vector, \(y \in \mathbb{R}^n\) is the angle vector of the output.

The arm joint of the joint robot system is considered as a subsystem. The local variable \((q_i, \dot{q}_i, \ddot{q}_i)\) is

The dynamic differential equation of the subsystem can be inferred as follows:

$$\begin{align*}
M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + P_i(\dot{q}_i) + H_i(q_i) + Z_i(q_i, \dot{q}_i, \ddot{q}_i) &= u_i \\
y_i &= q_i
\end{align*}$$

(2)

Where

$$Z_i(q_i, \dot{q}_i, \ddot{q}_i) = \begin{cases} 
\sum_{j=1, j \neq i}^n M_{ij}(q_j)\ddot{q}_j + [M_{ij}(q_j) - M_{ij}(q_i)]\ddot{q}_i \\
+ \sum_{j=1, j \neq i}^n C_{ij}(q_j, \dot{q}_j)\dot{q}_j + [C_{ij}(q_j, \dot{q}_j) - C_{ij}(q_i, \dot{q}_i)]\dot{q}_i \\
+ [\Pi_i(q_i) - H_i(q_i)]
\end{cases}$$

is the cross-linking term of the system.

**Theorem 1.** Ultimately uniformly bounded (UUB): To analyze the stability of the uncertain system, the solution of the differential equation describing the system is \(x(t_0) = x_0\). If the system is ultimately uniformly bounded for a set \(W \in \mathbb{R}^n\) containing an origin, then there exists a nonnegative constant \(T(x_0, W) < \infty\) such that it holds for all \(t \geq t_0 + T:\)

$$\|x(t)\| \leq \delta \Rightarrow x(t) \in W$$

(3)

Where \(W\) may be related to \(x_0\), but it must not be related to \(t_0\).

**Lemma1.** Barbatra lemma: Let \(x : [0, \infty) \rightarrow \mathbb{R}\) be a first-order continuous differentiable; if \(t \rightarrow \infty\) it has a limit, then it \(\dot{x}(t), t \in [0, \infty)\) is uniformly continuous, then \(\lim_{t \to \infty} \dot{x}(t) = 0\).

Some fundamental properties are addressed as follows:

**Property1.** Inertia matrix \(M(q)\) is a bounded symmetric positive definite matrix; There are 1, 2 such that

$$0 < M_{\min}(q) \leq M(q) \leq M_{\max}(q)$$

(4)

**Property2.** The absolute values of \(P(\dot{q})\) and \(G(q)\) satisfy

$$|P(\dot{q})| \leq p$$

and

$$|H(q)| \leq h$$

(5)

Where \(p\) and \(h\) are positive constants.

In what follows, we can make the following assumption.

**Assumption1.** The expected trajectory \(x_{id}, \dot{x}_{id}, \ddot{x}_{id}\) is continuous and bounded.

**Assumption2.** The associated term \(Z(x_i, \dot{x}_i, \ddot{x}_i)\) is bounded and satisfies [14].
\[ Z(x_i, \hat{x}_i, \ddot{x}_i) \leq \sum_{j=1}^{n} d_j S_j \]  
(6)

Where \( S_j \leq 1 + |s_j| + |s_j|^2, d_y \geq 0 \). Let the maximum value of \( d_y \), \( \max \{d_y\} = \delta \).

### 3 Neural network observer design

Taking the subsystem model of the manipulator as the object, considering the known dynamics model of the manipulator to design the RBF neural network adaptive observer, and the adaptive control can be realized when the speed information and the model are unknown.

Let \( x_{i1} = q_i, x_{i2} = \dot{q}_i \), the equation (2) is transformed into the state space equation as

\[
\dot{x}_{i1} = x_{i2} \\
\dot{x}_{i2} = M_i^{-1}(x_i)(u_i - C_i(x_{i1}, x_{i2})x_{i2} - P_i(x_{i2}) - H_i(x_i) - Z_i(x_{i1}, x_{i2})) \\
y_i = x_{i1}
\]  
(7)

Where \( F_i(x) = M_i^{-1}(x_i)(-C_i(x_{i1}, x_{i2})x_{i2} - P_i(x_{i2}) - H_i(x_i), G_i(x) = -M_i^{-1}(x_i)Z_i(x_{i1}, x_{i2}) \). The general formula is

\[
\dot{x}_i = Ax_i + b(F_i(x_i) + G_i(x_i)u_i + D_i(x_i)) \\
y_i = cx_i
\]  
(8)

Where \( x = [x_1 \ x_2]^T, A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, b = [0 \ 1]^T, c = [1 \ 0]^T \). In the system, \( M_i(x_{i1}), C_i(x_{i1}, \hat{x}_{i1}) \) are uncertain, and \( Z_i(x_{i1}, x_{i2}) \) is the cross-linking term between subsystems, so \( F_i(x), G_i(x) \) are unknown nonlinear functions. The radial basis function neural network has fast approximation speed and high approximation accuracy for unknown nonlinear continuous function.

The approximation algorithm of the RBF network is

\[
\begin{align*}
h_j & = \exp(-\|x - c_{ij}\|^2 \ / b_j^2) \\
y & = W^T h(x) + \varepsilon
\end{align*}
\]  
(9)

Where \( h_j \) is the output of Gaussian function, that is, the output of the \( j \)th neuron, \( i \) is the number of inputs of the network, \( j \) is the number of nodes in the hidden layer, \( c_{ij} \) is the coordinate vector of the center point of the Gaussian basis function, \( b_j \) is the width of the Gaussian basis function, \( x \) is the input of the neural network, \( W = [w_1, w_2, \cdots, w_n] \) is the weight of the network, \( \varepsilon \) is the approximation error of the network, \( \varepsilon \leq \varepsilon_n \), the network output \( y(t) = W^T h = w_1 h_1 + w_2 h_2 + \cdots + w_n h_n \).

Define optimal approximation of nonlinear functions as

\[
\begin{align*}
\hat{F}_i(x_i) & = \hat{W}_{i1}^{*T} \cdot h_i(x_i) + \varepsilon_{i1F}, |\varepsilon_{i1F}| \leq \varepsilon_{i1} \\
\hat{G}_i(x_i) & = \hat{W}_{i2}^{*T} \cdot h_2(x_i) + \varepsilon_{i2G}, |\varepsilon_{i2G}| \leq \varepsilon_{i2}
\end{align*}
\]  
(10)

Where \( \hat{W}_j^* (j = 1, 2) \) is the ideal weight of the neural network; \( \varepsilon_{i1F}, \varepsilon_{i2G} \) are the approximation error of the network; \( \varepsilon_{i1}, \varepsilon_{i2} \) are positive constants.

Given a small positive number \( (\eta_{i0}, \varepsilon_{i0}) \) and the continuous functions \( F_i(x) \) and \( G_i(x) \) to be approximated, there exists an ideal weight vector \( W^* \) such that

\[
\begin{align*}
\max \| F_i(x_i) - \hat{F}_i^*(x_i) \| & \leq \eta_{i0} \\
\max \| G_i(x_i) - \hat{G}_i^*(x_i) \| & \leq \varepsilon_{i0}
\end{align*}
\]  
(11)

Define the estimation error as

\[
\begin{align*}
\tilde{W}_j = \hat{W}_j - \hat{W}_j^* (j = 1, 2), \text{ where the optimal weight is}
\end{align*}
\]  
(12)

\[
W^* = \arg \min_{W \in \Omega} \sup F_i(x_i) - \hat{F}_i^*(x_i) \]  
(12)

Design the neural network observer for the subsystem as

\[
\dot{x}_i = Ax_i + b(F_i(x_i) + G_i(x_i)u_i + D_i(x_i)) \\
y_i = c^T x_i
\]  
(13)

Where \( \hat{x}_i \) is estimated value of \( x_i, K = [k_{i1} \ k_{i2}]^T \) is the feedback gain of the observer, and the state error and output error are

\[
\dot{\tilde{x}}_i = x_i - \hat{x}_i, \dot{\tilde{y}}_i = y_i - \hat{y}_i, \hat{F}_i(x_i) = \hat{W}_{i1}^{*T} \cdot h_i(\tilde{x}_i), \]  
(14)

\[
\hat{G}_i(x_i) = \hat{W}_{i2}^{*T} \cdot h_2(\tilde{x}_i), \hat{W}_j(j = 1, 2) \text{ are the weights of function approximation and } (A - KC_i^T) \text{ satisfy Hurwitz criterion.}
\]

From (8) and (13), we have

\[
\dot{\tilde{x}}_i = (A - K_i^T) \tilde{x}_i + b[\hat{F}_i(x_i), \tilde{x}_i] + \hat{G}_i(x_i, \tilde{x}_i) u_i,
\]  
(14)

Define approximation errors of nonlinear functions as
\[ \dot{\xi}_i = F_i^*(x_i) + \varepsilon_{iiF} - \tilde{F}_i(\hat{x}_i) \]
\[ + \left[(\tilde{G}_i^*(x_i) + \varepsilon_{i2G}) - \tilde{G}_i(\hat{x}_i)\right]u_i \]
\[ = \tilde{W}_{i1}^{\top}h_i(x_i) + \varepsilon_{iiF} \]
\[ - \tilde{W}_{i2}^{\top}h_i(x_i) + \left[(\tilde{W}_{i2}^{\top}h_i(x_i) + \varepsilon_{i2G}) \right]u_i \]
\[ (15) \]
Assumption 3. The approximation error 1 of nonlinear function neural network is bounded and satisfies the requirement:
\[ |\dot{\xi}_i| \leq \xi_{i0} \]
\[ (16) \]
Where \( \xi_{i0} \) is a positive constant, the weight update rate of the neural network is as follows:
\[ \left\{ \begin{array}{l}
\dot{\tilde{W}}_{i1} = \rho_{i1}\tilde{\xi}_i h_i(\hat{x}_i) \\
\dot{\tilde{W}}_{i2} = \rho_{i2}\tilde{\xi}_i h_i(\hat{x}_i)u_i
\end{array} \right. \]
\[ (17) \]
Proof. Consider the Lyapunov function candidate
\[ V_{io} = \frac{1}{2}\tilde{\xi}_i^2 + \frac{1}{2}\tilde{x}_i^2 + \frac{1}{2}\tilde{\xi}_i^{\top}\tilde{W}_{i1}\tilde{W}_{i1} \]
\[ (18) \]
Calculating the time derivative of \( V_{io} \)
\[ \dot{V}_{io} = \tilde{\xi}_i\dot{\tilde{\xi}}_i + \tilde{x}_i\dot{\tilde{x}}_i + \frac{1}{\rho_{i1}}\tilde{W}_{i1}^{\top}\dot{\tilde{W}}_{i1} \]
\[ + \frac{1}{\rho_{i2}}\tilde{W}_{i2}^{\top}\dot{\tilde{W}}_{i2} \]
\[ (19) \]
According to the formulas (14) and (19), we obtain
\[ \dot{V}_{io} = \tilde{x}_i\dot{\tilde{x}}_i - k_{i1}\tilde{x}_i^2 - k_{i2}\tilde{x}_i^2 + \frac{1}{\rho_{i1}}\tilde{W}_{i1}^{\top}\dot{\tilde{W}}_{i1} - \frac{1}{\rho_{i2}}\tilde{W}_{i2}^{\top}\dot{\tilde{W}}_{i2} \]
\[ (20) \]
Taking the weight update rate formula (17) of the neural network and substitute it into equation (20), the following equality is obtained:
\[ \dot{V}_{io} = \tilde{x}_i\dot{\tilde{x}}_i - k_{i1}\tilde{x}_i^2 - k_{i2}\tilde{x}_i^2 + \tilde{\xi}_i\dot{\tilde{\xi}}_i \]
\[ (21) \]
In view of (9) and (16), we can get
\[ \dot{V}_{io} \leq -(k_{i1} - \varepsilon_{i0})\tilde{x}_i^2 - \frac{1}{2}(\tilde{x}_i^2 - \tilde{x}_i^2) - |\tilde{x}_i|\varepsilon_{i0} \]
\[ (22) \]
Let \( V_{io} \leq 0 \), we have
\[ |\tilde{x}_i| \leq \frac{\xi_{i0}}{k_{i2} - 1/2} \]
\[ (23) \]
Thus \( V_{io} \leq 0 \). According to the barbat lemma, the neural network observer is asymptotically stable. This completes the proof.

### 4 Design of sliding mode controller with neural network observer

Given (13), the state-space expression of the system with the observer is
\[ \begin{align*}
\dot{\hat{x}}_{i1} & = \hat{x}_{i1} + k_{i1}(x_{i1} - \hat{x}_{i1}) \\
\dot{\hat{x}}_{i2} & = \tilde{F}_i(\hat{x}_i) + \tilde{G}_i(\hat{x}_i)u_i + D_i(x_i) + k_{i2}(x_{i1} - \hat{x}_{i1})
\end{align*} \]
\[ (24) \]
Define State error as
\[ \begin{align*}
e_{i1} & = \hat{x}_{i1} - x_{i1d} \\
\hat{e}_i & = \hat{x}_{i2} - x_{i2d}
\end{align*} \]
\[ (25) \]
Where \( x_{i1d}, x_{i2d} \) are the expected trajectory vectors in joint space. It is hoped that \( x_{i1} \) it can track \( x_{i1d} \) in finite time.

Define the sliding modulus and substitute it into equation (25), we have the following inequality
\[ s_i = c_i e_i + \hat{e}_i = c_i(\hat{x}_{i1} - x_{i1d}) + (\hat{x}_{i2} - x_{i2d}) \]
\[ (26) \]
Taking the derivative of \( s_i \) and substitute it into equation (24), we have
\[ \dot{s}_i = c_i \dot{e}_i + \dot{\hat{e}}_i = c_i(\dot{\hat{x}}_{i1} - \dot{\hat{x}}_{i1d}) + (\dot{\hat{x}}_{i2} - \dot{\hat{x}}_{i2d}) \]
\[ (27) \]
Thus
\[ V_{kc} = \frac{1}{2}s_i^2 \]
\[ (28) \]
Taking the derivative of \( V_{kc} \) to design sliding mode control rate and using the equation (27), by Assumption2, we obtain
Therefore, the design control rate is
\[
V'_c = \sum_{i=1}^{n} \left( F_i(x) + G_i(x)u_i + k_{i2}(x_i - \bar{x}_i) - \bar{x}_{i,1d} \right) \\
+ \max \{ d_j \} \sum_{j=1}^{n} s_j \sum_{j=1}^{n} s_j 
\]
(29)

\[
s_i [c_i(\bar{x}_{i2} + k_{i1}(x_{i1} - \bar{x}_{i1})) + k_{i2}(x_{i1} - \bar{x}_{i1}) - \bar{x}_{i,1d}] + \sum_{i=1}^{n} (F_i(x) + G_i(x)u_i + k_{i2}(x_i - \bar{x}_i)) + \sum_{i=1}^{n} \delta |s_i| s_i |s_i|
\]

Therefore, the design control rate is
\[
u_i = \frac{1}{G_i(\bar{x}_i)} [-F_i(\bar{x}_i) - c_i(\bar{x}_{i2} + k_{i1}(x_{i1} - \bar{x}_{i1})) - k_{i2}(x_i - \bar{x}_i) + \bar{x}_{i,1d}] + \sum_{i=1}^{n} \delta |s_i| s_i |s_i|
\]
(30)

Thus \( V'_c \leq 0 \), which implies that the Lyapunov function of the whole system is \( V_i = V_{i0} + V_{iC}, V_i \leq 0 \). the closed-loop system is stable, and the trajectory tracking error is ultimately uniformly bounded. This completes the proof.

### 5 Simulation example

In order to verify the effectiveness of the distributed sliding mode controller based on the neural network observer, a multi-joint joint is used for numerical simulation, where the mass \( m_i = 1 \text{kg} \), the rod length \( l_i = 1 \text{m} \), \( M_i = 0.5 \text{kg} \cdot \text{m}^2 \). Choose \( c_i = 20 \) that satisfies the Hurwitz condition, the desired trajectory \( x_{i,1d} = \sin(t) \), the initial state \( x_i(0) = [0.2 \ 1]^T \), the feedback gain \( k_{i1} = 400, k_{i2} = 200 \), and the initial weight of the neural network is set to 0;

![Figure 2. Position tracking of the i-th joint.](image)

It can be seen from Figure 2 and Figure 3 that the decentralized control method based on the neural network observer can quickly track the desired position and speed, which means that the speed response is fast. It can be seen from Figure 4 that when the system is running stably, the position overshoot remains within 0.1, which means good stability. Figures 5 and 6 show that the joint position observer and the speed observer can achieve a better estimation effect on the system position and speed, and the observer state can be used to design the controller. In summary, the decentralized control method based on the neural network observer can achieve good tracking control of the system position and speed.

![Figure 3. Speed tracking of the i-th joint.](image)

![Figure 4. Position tracking error of the i-th joint.](image)

![Figure 5. Position estimation of the i-th joint.](image)

![Figure 6. Velocity estimation of the i-th joint.](image)
6 Conclusion

In view of the fact that many parameters of the robot joint system will change in actual work, the coupling relationship between the various subsystems and the existence of modeling errors make the model of the system difficult to determine, a decentralized control method based on neural network observers is designed. The RBF neural network observer is used to estimate the state of the system without actually measuring the signal of the original system. By processing the cross-linked items of multiple joints, the control method of each subsystem is designed to solve the problem of the unknown system state of the system model. The problem provides a theoretical basis for the sliding mode control of the neural network observer as an effective method of system control.

Acknowledgments

This work was supported by the National Key R&D Program of China, under Grant 2019YFB2005804.

References

8. Y. Wu, C. Wang, CTT, 30,990(2013)
14. B. Zhao, Y. Li, JCD,29, 226 (2014)