Adaptive Control of 3R Manipulator in the Workspace Based on Radial Basis Function Neural Network

Dejin Wang[1], Min Li[2], Fuyong Wang[1], Zhongxin Liu[1](✉), and Zengqiang Chen[1]

[1] College of Artificial Intelligence, Nankai University, Tianjin 300350, China
lzhx@nankai.edu.cn

[2] School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430000, China

Abstract. In this paper, an adaptive control method based on radial basis function(RBF) neural network is proposed for the force control of 3R manipulator in the workspace. The control goal is to make the manipulator track the impedance trajectory to realize the flexible contact between the manipulator and the environment. Firstly, the Lagrangian-Euler method is used to model the dynamic of 3R manipulator. According to the characteristics of the dynamical model of 3R manipulator, a RBF neural network is adopted to design the adaptive controller, and the stability of the controller is analyzed by using Lyapunov criterion. Through the Simulink module in MATLAB, the effectiveness of the proposed algorithm is verified.

Keywords: 3R manipulator · Adaptive control · Dynamical model · Radial basis function · Neural network

1 Introduction

With the rapid development of control theory and computer science, the importance of manipulators is gradually improved. In modern manufacturing system [1], industrial manipulator has become an indispensable role, which greatly improves the production efficiency and quality of manufacturing industry. The wide use of manipulators in mechanical manufacturing, aviation, electronic product manufacturing and other fields has promoted the development of manipulator control theory. But the dynamic equation of multi-joint manipulators is extremely complex and nonlinear [2], which restricts the further development of industrial manipulators. The traditional PID control has a good performance on the linear system [3]. While it maybe impractical to the nonlinear and strong real-time manipulator system [4]. With the improvement of artificial intelligence algorithm and control theory, advanced intelligent control methods have been proposed, which provide an effective solution for the control of complex dynamic manipulator system [5].

Y. Jia et al. (Eds.): CISC 2022, LNEE 950, pp. 443–451, 2022.
https://doi.org/10.1007/978-981-19-6203-5_43

At present, neural network adaptive control is widely adopted to control the manipulator systems. Shao et al. used neural network adaptive control to control the manipulator with unknown disturbance. The results show that it can ensure the transient response of the system and realizes asymptotic convergence [6]. Bang et al. proposed a robust neural network adaptive control method, which can effectively compensate the system uncertainty, and can ensure the rapid stability of the manipulator system [7]. To solve the tracking control problem of n-joint manipulator with large jump parameters, Li et.developed a weighted multi-model neural network adaptive control method, and the simulation showd that it has good performance [8].

Force control of the manipulator is the basis for the manipulator to be applied to tasks such as assembly, polishing, deburring, etc. [9]. Force control technology can significantly improve the performance of the manipulator when it is in contact with the environment, and can improve the intelligence of the manipulator [10]. Most existing paper focus on position control of the manipulator rather than the force control [11], which motivate our works. The main contribution of this paper can be summarized as follows:

(i). The Lagrangian-Euler method is used to model the dynamics of 3R manipulator and the RBF network is adopted to fit the dynamics model of 3R manipulator.
(ii). An adaptive control algorithm based on RBF neural network is proposed for the force control of 3R manipulator in the workspace. The control goal is to make the manipulator track the impedance trajectory to realize the flexible contact between the manipulator and the environment.

The rest of this paper is organized as follows: In Sect. 2, dynamics modelling of 3R manipulator is given. In Sect. 3, the neural network modelling is given and an adaptive controller of 3R manipulator is designed, Sect. 4 shows the simulation results and Sect. 5 gets the conclusions.

2 Dynamics Modeling of 3R Manipulator

2.1 Structure and Standard D-H Parameters of 3R Manipulator

In this paper, the parameter table of 3R manipulator is established by standard D-H method, the D-H coordinate system can be represented by a_i,α_i,θ_i,d_i.

The mechanical structure of 3R manipulator is shown Fig. 1. The length of connecting rod 1, connecting rod 2 and connecting rod 3 are $l_1 = 0.5\,\text{m}$, $l_2 = 1\,\text{m}$, $m_1 = 0.8\,\text{m}$, respectively. The mass is $m_1 = 0.5\,\text{kg}$, $m_2 = 1\,\text{kg}$, $m_3 = 0.8\,\text{kg}$, respectively. The rotation angles of the three joints are q_1, q_2, q_3, and the standard D-H parameters of the manipulator can be obtained in Table 1 [12].

Table 1. 3R manipulator standard DH parameters

Coordinate system number	a_i	α_i	d_i	q_i
1	0	90^o	0	q_1
2	l_2	0	l_1	q_2
3	l_3	0	0	q_3

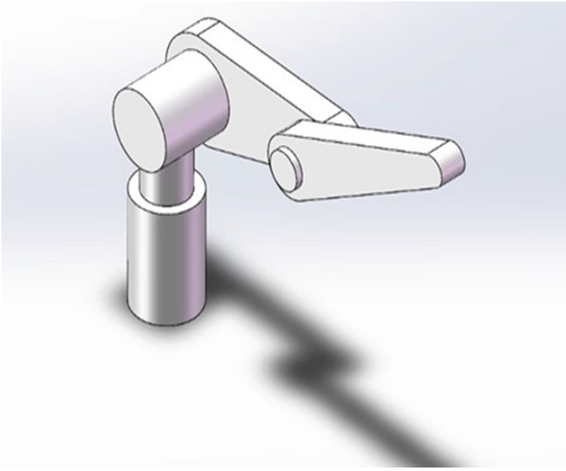

Fig. 1. Mechanical structure of 3R manipulator

2.2 Lagrange Euler Dynamics Modeling of 3R Manipulator

The Lagrange-Euler dynamic equation is closed-loop, which is convenient for the design and implementation of the controller. Therefore, the Lagrange-Euler method is adopted for dynamic modelling. For a 3-joint manipulator, the Lagrangian dynamic equation of its joint space can be expressed as

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{1}$$

In (1), q represents the joint angle, τ represents the joint torque matrix, which are all three-dimensional vectors, $D\,(q)$ represents the inertia matrix, $C\,(q,\dot{q})$represents the centrifugal force and Coriolis force matrix, $G\,(q)$represents the gravity matrix, $D(q)$, $C(q)$ is a 3×3 square matrix and $G(q)$ is a 3×1 matrix.

However, the force control of 3R manipulator is working in the workspace. So it is necessary to convert the dynamic equation of joint space into the workspace by using the Jacobian matrix. Let x be the position coordinates of the end of the manipulator in the base coordinate system, then we get [13]

$$\dot{x} = J(q)\dot{q} \tag{2}$$

$$\ddot{x} = J\left(q\right)\ddot{q} + \dot{J}(q)\dot{q} \tag{3}$$

$$\tau = J^{T}(q)F_{x} \tag{4}$$

$$D_{x}\left(q\right)\ddot{x} + C_{x}\left(q,\dot{q}\right)\dot{x} + G_{x}\left(q\right) = F_{x} \tag{5}$$

$$D_{x}\left(q\right) = J^{-T}(q)D\left(q\right)J^{-1}(q) \tag{6}$$

$$C_{x}\left(q,\dot{q}\right) = J^{-T}(q)(C\left(q,\dot{q}\right) - D\left(q\right)J^{-T}(q)\dot{J}(q))J^{-1}(q) \tag{7}$$

$$G_{x}\left(q\right) = J^{-T}(q)G\left(q\right) \tag{8}$$

The Jacobian matrix $J(q)$ is defined as $J(q) = \partial h(q)/\partial q$. F_x denotes the force in the workspace, $D_x\left(q\right)$, $C_x\left(q,\dot{q}\right)$, $G_x\left(q\right)$ denotes the inertia matrix, centrifugal force and Coriolis force matrix, gravity matrix in the workspace, respectively.

3 3R Manipulator Neural Network Adaptive Force Control

3.1 3R Manipulator Neural Network Modeling

The dynamic model of the manipulator in the workspace can be expressed as:

$$D_{x}\left(q\right)\ddot{x} + C_{x}\left(q,\dot{q}\right)\dot{x} + G_{x}\left(q\right) = F_{x} \tag{9}$$

The main aim of neural network modeling is to use a RBF neural network to fit the inertia matrix $D_x\left(q\right)$, centrifugal force and Coriolis force matrix $C_x\left(q,\dot{q}\right)$, gravity matrix $G_x\left(q\right)$.

$$d_{xkj}\left(q\right) = \sum_{i=1}^{l} Q_{kjl}\xi_{kjl}\left(q\right) + \epsilon_{dkj}(q) = Q_{kj}^{T}\xi_{kj}(q) + \epsilon_{dkj}(q) \tag{10}$$

$$c_{xkj}\left(q,\dot{q}\right) = \sum_{i=1}^{l} \alpha_{kjl}\xi_{kjl}\left(z\right) + \epsilon_{ckj}\left(z\right) = \alpha_{kj}^{T}\xi_{kj}\left(q\right) + \epsilon_{ckj}\left(z\right) \tag{11}$$

$$g_{xk}\left(q\right) = \sum_{i=1}^{l} \beta_{kl}\eta_{kl}\left(q\right) + \epsilon_{gk}(q) = \beta_{k}^{T}\eta_{k}(q) + \epsilon_{gk}(q) \tag{12}$$

In (10), d_{xkj} represents inertia Matrix $D_x\left(q\right)$ k-row j-column elements, similarly c_{xkj}, g_{xk} in (11), (12). Q_{kjl} and β_{kl} are the weight that fit j-row elements of $D_x\left(q\right)$, k-row elements of $G_x\left(q\right)$, respectively. $\xi_{kjl}(q)$ and $\eta_{kl}(q)$ are the radial basis functions with input q. $\epsilon_{dkj}(q)$ and $\epsilon_{gk}(q)$ are the modeling errors of $d_{xkj}(q)$ and $g_{xk}(q)$, respectively, and assume that they are bounded.

In (11) where $z = \left[q^T \ \dot{q}^T\right]$, α_{kjl} is the weight, ξ_{kjl} is the radial basis function whose input is the vector z, which is also the Gaussian basis function, $\epsilon_{ckj}(z)$ is $c_{xkj}(q, \dot{q})$ modeling errors and assume it is bounded.

Using the GL matrix [15] and its multiplication operation $D_x(q)$ can be written as:

$$D_x(q) = \left[\{Q\}^T . \{E(q)\}\right] + E_D(q) \tag{13}$$

where $\{Q\}^T$ and $\{E(q)\}$ are GL matrices whose elements are Q_{kj} and ξ_{kj}, respectively. $E_D(q)$ is a matrix whose elements are modeling errors ϵ_{dkj}. Similarly, for $C_x(q, \dot{q})$, $G_x(q)$ there are:

$$C_x(q, \dot{q}) = \left[\{A\}^T . \{Z(z)\}\right] + E_c(z) \tag{14}$$

$$G_x(q) = \left[\{B\}^T . \{H(q)\}\right] + E_G(q) \tag{15}$$

where $\{A\}$, {Z(z)}, {B}, {H(q)} is the GL matrix whose elements are α_{kj}, $\xi_{kj}(z)$, β_k, $\eta_k(q)$. $E_c(z)$ and $E_G(q)$ are matrices whose elements are modeling errors ϵ_{ckj} and ϵ_{gk}.

3.2 Position-Based Impedance Control

The impedance control of the manipulator is to establish the relationship between the displacement of the manipulator end and the contact force by equivalently controlling the force/position of the end of the manipulator as a "spring-mass-damping" system. [14].

The contact force at the end of the manipulator is F_e, and the kinetics are described as [16]

$$M_m(\ddot{x}_c - \ddot{x}) + B_m(\dot{x}_c - \dot{x}) + K_m(x_c - x) = F_e \tag{16}$$

where x_c is the reference trajectory of the command position, x is the actual position, so the position error can be denoted as $x_c - x$, $x(0) = x_c(0)$, M_m, B_m, K_m are the mass, damping and stiffness coefficient matrices, respectively. By solving the differential Eq. (16), one can get the impedance trajectory x_d.

In impedance control, the environmental dynamics model needs to be considered. For simplicity, the environmental dynamics model is assumed to be the stiffness model.

$$F = K_e(x - x_e) \tag{17}$$

In 3R manipulator control, F_e is the 3×1-dimensional vector, which represents the contact force between the manipulator end and the environment. x_e represents the location of the environment. Assuming the environment as a stiffness model, where K_e is a 3×3-dimensional matrix represents the stiffness matrix of the environment, similar to M_d, K_d, B_d in impedance control, is also a positive definite diagonal matrix, which means that the environment is decoupled in each coordinate direction in the workspace.

3.3 Neural Network Adaptive Force Control Controller Design and Stability Analysis

Let x_d be the impedance trajectory in the workspace, then $\dot{x}_d$ and $\ddot{x}_d$ are the ideal impedance velocity and impedance acceleration, respectively.

Define

$$e(t) = x_d(t) - x(t) \tag{18}$$

$$\dot{x}_r(t) = \dot{x}_d(t) + \Lambda \mathrm{e}(t) \tag{19}$$

$$r(t) = \dot{x}_r(t) - x(t) = \dot{e}(t) + \Lambda \mathrm{e}(t) \tag{20}$$

Using $(\hat{.})$ to represent the estimated value of $(.)$, define $(\tilde{.}) = (.) - (\hat{.})$, then $\left\{\hat{Q}\right\}$, $\left\{\hat{A}\right\}$ and $\left\{\hat{B}\right\}$ represent the estimated values of $\{Q\}$, $\{A\}$ and $\{B\}$, respectively.

The controller can be designed as:

$$F_x = \left[\left\{\hat{Q}\right\}^T . \{E(q)\}\right] \ddot{x_r} + \left[\left\{\hat{A}\right\}^T . \{Z(z)\}\right] \dot{x}_r + \left[\left\{\hat{B}\right\}^T . \{H(q)\}\right] + F_e + Kr + k_s sgn(r) \tag{21}$$

In (21), where $K \in R > 0$, $k_s > E$, $E = E_D(q)\ddot{x_r} + E_C(z)\dot{x}_r + E_G(q)$. The first three terms of the controller are model-based control, the Kr term is equivalent to proportional-differential control, F_e is the contact force, and the last term of is the robust term to inhibit the neural network construction modeling error. The dynamic model of the manipulator can be modeled as

$$D_x(q)\ddot{x} + C_x(q, \dot{q})\dot{x} + G_x(q) + F_e = F_x \tag{22}$$

The stability of system (22) is given by the following theorem.

Theorem 1: For the closed-loop system (22), if $K \in R > 0$, $k_s > \|\mathrm{E}\|$, the design adaptive law is

$$\begin{cases} \dot{\hat{Q}}_k = \mathit{\Gamma}_k . \{\xi_k(q)\} \ddot{x}_r r_k \\ \dot{\hat{\alpha}}_k = \mathrm{M}_k . \{\xi_k(z)\} \dot{x}_r r_k \\ \dot{\hat{\beta}}_k = \mathrm{N}_k \eta_k r_k \end{cases} \tag{23}$$

where ${\Gamma_k}^T = \Gamma_k > 0$, ${M_k}^T = M_k > 0$, ${N_k}^T = N_k > 0$, And $\hat{Q}_k$, $\hat{\alpha}_k$ are matices whose elements are $\hat{Q}_{kj}$ and $\hat{\alpha}_{kj}$, respectively. Then $\hat{Q}_k$, $\hat{\beta}_k$, $\hat{\alpha}_k, \in \mathrm{L}_\infty$, $e \in L_2^n \cap L_\infty^n$. e is continuous, and when $\mathrm{t} \to \infty$, $\mathrm{e} \to 0$, $\dot{e} \to 0$.

Proof. In [15], the author has proved the position controller is stable under the adaptive law (23), with the similar approach, it is easily to prove that designed force controller is stable.

4 Simulation

Consider the situation: the manipulator is tracking spiral trajectory (24), but there is an obstacle at $x = 1$, therefore, when $x \geq 1$, the manipulator will have contact with the obstacle.

$$\begin{cases} x_d = 1 + 0.3\cos(\pi t) \\ y_d = 1 + 0.3\sin(\pi t) \\ z_d = 0.05t \end{cases} \tag{24}$$

The mechanical model of this obstacle is the stiffness model, that is,

$$F_e = K_e(x - x_e) \tag{25}$$

Set obstacle stiffness $K_e = \begin{bmatrix} 800 & 0 & 0 \\ 0 & 800 & 0 \\ 0 & 0 & 800 \end{bmatrix}$, the stiffness matrix of the manipulator is $K_d = \begin{bmatrix} 900 & 0 & 0 \\ 0 & 900 & 0 \\ 0 & 0 & 900 \end{bmatrix}$, the damping matrix is $B_d = \begin{bmatrix} 40 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 40 \end{bmatrix}$ inertia matrix is $M_b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, the gain coefficient of the adaptive law is $\Gamma_k = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}$ $\mathrm{M}_k = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}$ $\mathrm{N}_k = \begin{bmatrix} 0.6 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 0.6 \end{bmatrix}$. Figure 2 shows the impedance trajectory, as long as this impedance trajectory is tracked, the manipulator can achieve flexible contact with the environment. Figure 3 offers the tracking performance of the impedance trajectory. Figure 4 shows the tracking performance of the ideal trajectory. From Fig. 3 and Fig. 4, we can conclude that when $x > 1$, the manipulator track the impedance trajectory rather than the ideal trajectory. Therefore, the flexible contact between the manipulator and the environment can be achieved. From Fig. 5, we can conclude that the contact force does not experience a sudden increase, which means the compliant control is successful.

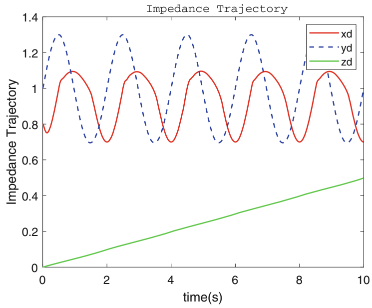

Fig. 2. Impedance trajectory

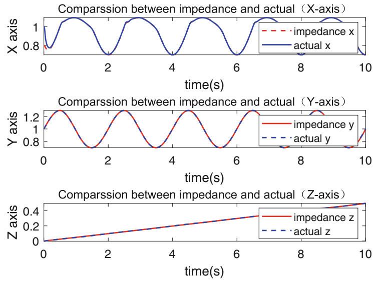

Fig. 3. Compression between impedance and actual

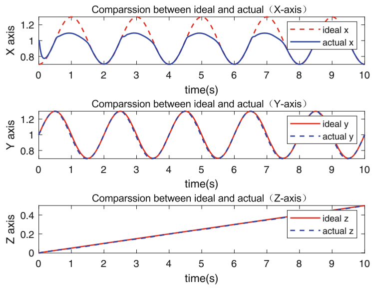

Fig. 4. Compression between impedance and actual

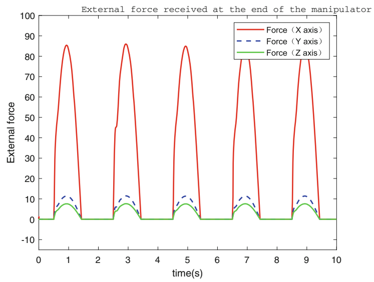

Fig. 5. External force

5 Conclusion

The main focus of this paper is the neural network adaptive control of 3R manipulator. By applying the Lagrangian-Euler method to the 3R manipulator to model the joint space dynamics and use the Jacobian matrix to convert the joint space dynamics into the workspace. A RBF neural network is adopted to fit the dynamical model. In force control, our focus is impedance control based on position control. According to the force measured by the force sensor at the end of the manipulator, the impedance trajectory is calculated, and the desired impedance model can be generated by tracking this impedance trajectory. If the manipulator encounters an obstacle in the middle, the contact force between the objects changes the trajectory to have a compliant contact with the environment.

Acknowledgements. This work was supported by Tianjin Natural Science Foundation of China (20JCYBJC01060), the National Natural Science Foundation of China (62103203, 61973175), and the Fundamental Research Funds for the Central Universities, Nankai University(63211120).

References

1. Zhang, F.: The application of industrial robot in intelligent manufacturing. Int. J. Comput. Eng. **4**(4), 70–84 (2019)
2. Lin, S.T., Tsai, H.C.: Impedance control with on-line neural network compensator for dual-arm robots. J. Intell. Robot. Syst. **18**(1), 87–104 (1997). https://doi.org/10.1023/A:1007992700881
3. Richard, C.D., Robert, H.B.: Modern Control Systems. Twelfth Publishing House of Electronics Industry, Beijing (2015)
4. Bruno, S.: Robotics: Modelling, Planning and Control. Xi'an Jiaotong University Press, Xi'an (2020)
5. Su, W.: Research on Trajectory Tracking Adaptive Control Algorithm for Industrial Manipulators. Southeast University, Nanjing (2018)
6. Shao, S., Zhang, K.: RISE-adaptive neural control for robotic manipulators with unknown disturbances. IEEE Access **8**, 97729–97736 (2020). https://doi.org/10.1109/ACCESS.2020.2997383
7. Xing, B., Zhang, W.: Robust adaptive control for robotic manipulators based on RBFNN. In: Applied Mechanics and Materials, vol. 397, pp. 1477–1481. Trans Tech Publications Ltd (2013)
8. Li, J., Zhang, W., Zhu, Q.: Weighted multiple-model neural network adaptive control for robotic manipulators with jumping parameters. Complexity (2020)
9. Wang, D.: Research on Key Technologies of Force / Position Control of Robotic Manipulators Based on Impedance Control. Shandong University, Jinan (2018)
10. Lopes, A.M., Almeida, F.: Acceleration based forceimpedance control. In: Proceedings of the 25th IASTED International Conference on Modelling, Identification, and Control, pp. 73–103. Lanzarote (2006)
11. Zhe, L., Li, S., Zhang, X., et al.: Trajectory optimization of six-degree-of-freedom manipulator based on RBF neural network. Modular Mach. Tool Autom. Manuf. Tech. **1**(06), 17–20 (2021). https://doi.org/10.13462/j.cnki.mmtamt.2021.06.004
12. Xiong, Y.: Fundamentals of Robot Techniques. Huazhong University of Science and Technology Press, Wuhan (1996)
13. Ge, S.S., Lee, T.H., Harris, C.J.: Adaptive neural network control of robotic manipulators. World Sci. Ser. Robot. Intell. Syst. **19**, 396 (1998). https://doi.org/10.1142/3774
14. Li, Z.: Research and Application of Robot Force /Position Control Methods for Robot-Environment Interaction. Huazhong University of Science and Technology, Wuhan (2011)
15. Liu, J.: Design of Robot Control System and MATLAB Simulation. Tsinghua University Press, Beijing (2008)
16. Hogan, N.: Impedance control: an approach to manipulation. In: 1984 American Control Conference, pp. 304–313. IEEE (1984)