Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 12, No. 1, 2025

Identification of Robot Dynamic Parameters Based on BP Neural Network

Yingjie Zhong

School of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan, Shanxi, China

Abstract: This paper proposes a dynamic parameter identification method based on the BP neural network to improve the accuracy of dynamic modeling and control performance of industrial robots. The method employs Fourier polynomial trajectory planning and constructs a BP neural network model for parameter identification. A six-degree-of-freedom (DOF) robotic arm is simulated using the Matlab Robotics Toolbox for validation. The results demonstrate that the proposed method can achieve accurate dynamic parameter identification under complex conditions, exhibiting robustness against environmental noise and friction disturbances. Furthermore, it enhances the efficiency and accuracy of dynamic parameter identification.

Keywords: BP Neural Network; Robot Dynamics; Parameter Identification Introduction.

1. Introduction

New-quality productive forces have emerged in practice, demonstrating strong driving and supporting capabilities for high-quality development. Industrial robots, known for their high precision, flexibility, and adaptability, have been widely applied in various fields, particularly in interactive application scenarios involving rigid contact between the robot and the environment. Such scenarios include processing robots engaged in cutting, grinding, milling, and polishing tasks [1]. Model-based torque control enables real-time adjustments to the robot's motion, allowing it to respond promptly to rigid collisions and meet production requirements, thereby necessitating an accurate dynamic model. Neural networks, with their ability to learn from large volumes of input-output data, offer superior identification and generalization under complex conditions. This paper aims to investigate the problem of dynamic parameter identification using neural networks.

Researchers worldwide have conducted extensive studies on dynamic parameter identification. Early approaches relied on generating dynamic parameters directly from CAD models, but these methods were significantly affected by external disturbances, making it difficult to obtain accurate dynamic models. Gautier et al. [2] improved parameter identification accuracy by optimizing excitation trajectories to obtain smooth and continuous motion paths. However, measurement randomness often led to local optimal solutions. Among existing studies, offline identification methods are considered more reliable. The general process of parameter identification includes establishing a dynamic model, obtaining a minimal dynamic parameter set and linearizing the system, determining and optimizing excitation trajectories, collecting motion and torque data, performing data preprocessing, and solving the dynamic equations to compute torques. The collected data is then fed into the dynamic parameter identification equations, followed by model validation.

Traditional identification methods, such as the least squares method, are commonly used. For example, Zhou Jun et al. [3] applied linear least squares for dynamic parameter identification of flexible joint modular robots. However, the identified parameters were not sufficiently accurate and were highly sensitive to noise. Feng Limin et al. [4] adopted a weighted least squares approach to mitigate the effects of noise interference to some extent. Zhu Zhenbiao et al. [5] generated excitation trajectories using a combination of Fourier series and fifth-order polynomials, achieving precise torque tracking based on joint angle and current measurements, though issues such as trajectory discontinuities and accuracy remained. Zhang Shiyuan et al. [6] optimized excitation trajectory signals using genetic algorithms, effectively reducing errors. Chen Zhihuan et al. [7] proposed an improved version of the whale optimization algorithm. Liu Yu et al. [8] enhanced the genetic algorithm to mitigate premature convergence and local optima, significantly improving parameter identification accuracy. Zhang Luming et al. [9] utilized a neural network-based torque compensation approach for dynamic parameter identification, reducing the impact of nonlinear factors. Zhang Di et al. [10] established a neural network-based trajectory tracking model for robotic arms, enabling real-time trajectory adjustments, though their study did not delve deeply into dynamic parameter identification.

Among these methods, identification approaches derived from inverse dynamic models, particularly the traditional least squares regression-based parameter identification, are highly susceptible to environmental noise and friction disturbances, leading to reduced identification efficiency. To improve parameter identification accuracy, enhance efficiency, and mitigate the issue of local optimal solutions, this paper proposes a dynamic parameter identification method based on the BP neural network. The proposed method utilizes Fourier polynomial trajectory planning and constructs a neural network model. Finally, a six-degree-of-freedom robot is simulated using the Matlab Robotics Toolbox for parameter validation, confirming the accuracy of the identified model.

2. Robotic Arm Identification Model

2.1. Robot Dynamics Model

The Newton-Euler iterative method is used to construct the dynamic equations, with outward iteration for velocity and acceleration, followed by inward iteration for force and torque.

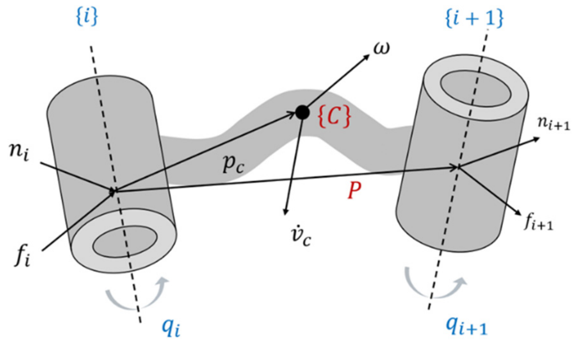

Fig 1. Dynamic parameters of the connecting rod

The outward iteration is used to calculate the angular velocity, linear acceleration, and angular acceleration of each link. The calculation of angular velocity can start from link 1 and proceed to link n. Given that the angular velocity of link i+1i+1i+1 in the iii-coordinate system is:

$$^{i}\omega_{i+1} = {}^{i}\omega_i + {}_{i+1}^{i}R\dot{\theta}_{i+1}\,{}^{i+1}Z_{i+1} \tag{1}$$

Where: $\dot{\theta}_{i+1}$ is the angular velocity of joint i+1, and ${}_{i+1}^{i}R$ is the rotation matrix from coordinate system {i+1} to {i}.

By left-multiplying Equation (1) by the rotation matrix ${}_{i}^{i+1}R$, the angular velocity of link i+1 in the {i+1} coordinate system is given by:

$$^{i+1}\omega_{i+1} = {}_{i}^{i+1}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \tag{2}$$

Where: ${}^{i}\omega_i$ is the angular velocity of link I in the {i} coordinate system, and ${}_{i}^{i+1}R$ is the rotation matrix from the {i} to the {i+1} coordinate system.

The angular acceleration transformation equation between the links can be obtained, which gives the angular acceleration of link i to i+1 as:

$$^{i+1}\dot{\omega}_{i+1} = {}_{i}^{i+1}R\,{}^{i}\dot{\omega}_i + {}_{i}^{i+1}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \tag{3}$$

Where $\dot{\theta}_{i+1}$ is the angular velocity of joint i+1, and $\ddot{\theta}_{i+1}$ is the angular acceleration of joint i+1.

The formula for calculating the linear acceleration of the link in a rotational joint manipulator is given by:v

$$^{A}\dot{V}_Q = {}^{B}\dot{V}_{BORG} + {}^{A}\Omega_B \times \left({}^{A}\Omega_B \times {}_{B}^{A}R\,{}^{B}Q\right) + {}^{A}\dot{\Omega}_B \times {}_{B}^{A}R\,{}^{B}Q \tag{4}$$

The linear acceleration at the origin of each link's coordinate system is given by:

$$^{i+1}\dot{v}_{i+1} = {}_{i}^{i+1}R\left({}^{i}\dot{\omega} \times {}^{i}P_{i+1} + {}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}P_{i+1}\right) + {}^{i}\dot{v}_i\right) \tag{5}$$

Similarly, the linear acceleration of the link's center of mass is given by:

$$^{i}\dot{v}_{c_i} = {}^{i}\dot{\omega}_i \times {}^{i}P_{c_i} + {}^{i}\omega_i \times \left({}^{i}\omega_i + {}^{i}P_{c_i}\right) + {}^{i}\dot{v}_i \tag{6}$$

Where ${}^{i}P_{c_i}$ is the vector of the manipulator.

The inertial force and torque at the center of mass of each link can be determined using the Newton-Euler equations:

$$F_i = m\dot{v}_{c_i} \tag{7}$$

$$N_i = {}^{c_i}I\dot{\omega}_i + \omega_i \times {}^{c_i}I\omega_i \tag{8}$$

After determining the forces and torques on each link, the torques at the joints need to be calculated, which correspond to the forces and torques applied to the links by the joint motors.

By iterating inward from link nnn to the base coordinate system, the inertial forces and torques acting on each link can be obtained:

$$^{i}f_i = {}_{i+1}^{i}R\,{}^{i+1}f_{i+1} + {}^{i}F_i \tag{9}$$

$$^{i}n_i = {}^{i}N_i + {}_{i+1}^{i}R\,{}^{i+1}n_{i+1} + {}^{i}P_{c_i} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}_{i+1}^{i}R\,{}^{i+1}f_{i+1} \tag{10}$$

The joint torque can be obtained by calculating the component of the torque applied by one link to the adjacent link along the Z-axis:

$$\tau_i = {}^{i}n_i^{T}\,{}^{i}\hat{Z}_i \tag{11}$$

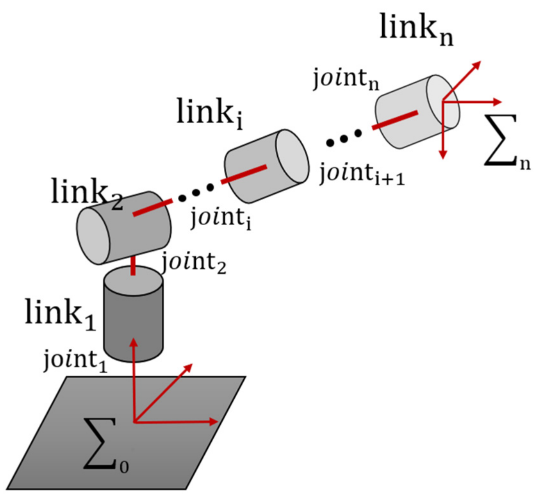

Fig 2. The link model of a robot with nnn degrees of freedom

As shown in the figure, the robot joints are typically connected in a series configuration. Where joint i (i≠0) is the i-th revolute joint, and link i is the i-th link.Units

In the absence of external forces, the torques acting on the center of mass of the link are summed, and their sum is set to zero, thus obtaining the torque balance equation. The dynamic equation is generally expressed as [11]:

$$\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) \tag{12}$$

Here, $\tau = [\tau_1, \tau_2, \tau_3, \cdots\cdots, \tau_n]^T$ represents the joint torque, $M(\theta) \in R^{n\times n}$ is the inertia matrix, $\theta, \dot{\theta}, \ddot{\theta}$ denote the joint position, angular velocity, and angular acceleration, respectively. $V(\theta) \in R^{n\times n}$ represents the Coriolis and centrifugal force matrix, and $G(\theta) \in R^{n\times n}$ represents the gravitational torque.

2.2. Linearization of the Dynamic Model and Solution of The Minimal Set of Dynamic Parameters

In the above dynamic model, the dynamic parameters are in a nonlinear form, which makes parameter identification difficult. Therefore, the dynamic model can be represented as follows:

$$\tau = Y\left(\theta,\dot{\theta},\ddot{\theta}\right)X_{full} \tag{13}$$

Where: $Y\left(\theta,\dot{\theta},\ddot{\theta}\right) \in R^{n\times n}$ is the function matrix of the joint position, angular velocity, and angular acceleration of the robotic arm, and X_{full} represents all dynamic parameters. For a robotic arm with n degrees of freedom, the dynamic inertia parameters of the i-th link of the joint can be expressed as follows:

$$X_{fulli} = \left[m, mx, my, mz, I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}\right] \tag{14}$$

Where: X_{fulli} represents the dynamic parameter of the i-th joint, I_{xx}, I_{yy}, I_{zz} represents the moment of inertia of the joint, and x, y, z represent the position of the joint's center of mass, I_{xy}, I_{xz}, I_{yz} represents the inertia product of the joint.

The dynamic parameters can be expressed as: $X = \left[X_1, X_2, X_3, \cdots, X_n\right]$

The dynamical model can be expressed as:

$$\tau = \begin{bmatrix}\tau_1\\ \vdots\\ \tau_n\end{bmatrix} = \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n}\\ Y_{21} & Y_{22} & \cdots & Y_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ Y_{n1} & Y_{n2} & \cdots & Y_{nn}\end{pmatrix}\begin{bmatrix}X_1\\ X_2\\ \vdots\\ X_n\end{bmatrix} = YX \tag{15}$$

However, the observation matrix of the robotic arm $Y = Y\left(\theta,\dot{\theta},\ddot{\theta}\right)$ is rank-deficient, with some rows and columns being constantly zero, while others are linear combinations of certain other rows and columns. It is necessary to reorganize and identify the set of dynamic parameters to obtain the minimal dynamic parameter set, which is required for the identification of dynamic parameters. In this way, the relationship between the joint torque and the minimal dynamic parameter set is:

$$\tau = Y(\theta,\dot{\theta},\ddot{\theta})X_{\min} \tag{16}$$

By linearizing the dynamical model, the identification of nonlinear dynamic parameters is transformed into the identification of linear dynamic parameters.

2.3. Excitation Trajectory Design and Data Collection

To activate all parameters in the robotic dynamics model and ensure that all parameters are sufficiently "activated" during the identification process, thereby allowing for precise collection of motion, torque, and other related datasets, it is necessary to design appropriate excitation trajectories for the robot. Fourier series-based trajectories [12] can ensure periodic input, with data collected from different cycles being consistent. This makes data collection convenient and highly feasible. Therefore, Fourier series can be used to construct excitation trajectories for collecting joint positions, angular velocities, and angular accelerations of the robotic arm. In the Fourier series, the angular displacement, angular velocity, and angular acceleration of the i-th rotational joint of the robotic arm are given by:

$$\begin{aligned} q_i(t) &= \sum_{l=1}^{N}\left[\frac{a_{il}}{w_f l}\sin\left(w_f lt\right) - \frac{b_{il}}{w_f l}\cos\left(w_f lt\right)\right] + q_{i0}\\ \dot{q}_i(t) &= \sum_{l=1}^{N}\left[a_{il}\cos\left(w_f lt\right) - b_{il}\sin\left(w_f lt\right)\right]\\ \ddot{q}_i(t) &= -w_f\sum_{l=1}^{N}\left[a_{il} l\sin\left(w_f lt\right) + b_{il} l\cos\left(w_f lt\right)\right] \end{aligned} \tag{17}$$

Where: a_{il}, b_{il} represents the amplitude of the cosine and sine components for the I -th joint at different frequencies, q_{i0} denotes the positional deviation of the i -th joint, N is the number of harmonic terms contained in the Fourier series, and w_f is the fundamental angular frequency of the excitation frequency.

For the UR5 robotic arm studied in this paper, a 5th-order Fourier series can be employed, specifically N = 5, where a_{il}, b_{il}, q_{i0} are the parameters to be optimized. Each joint has $2N+1$ parameters that require optimization for the excitation trajectory. The 5th-order Fourier excitation trajectory parameters were obtained using the Fmincon optimization toolbox in MATLAB.

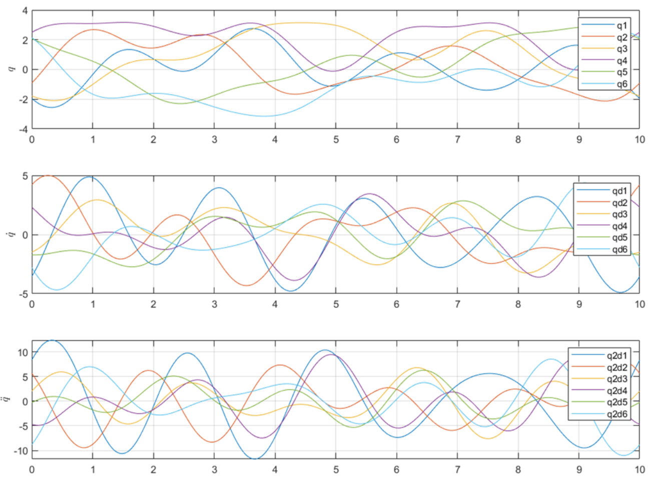

Fig 3. Fourier series excitation trajectories

During the real-time operation of the robotic arm along the excitation trajectory, the joint angles and joint angular velocities can be collected. By differentiating the Fourier series representation, the joint angular acceleration can be obtained. Additionally, the motor current can be used to infer the motor output torque, leading to the calculation of the joint torque.

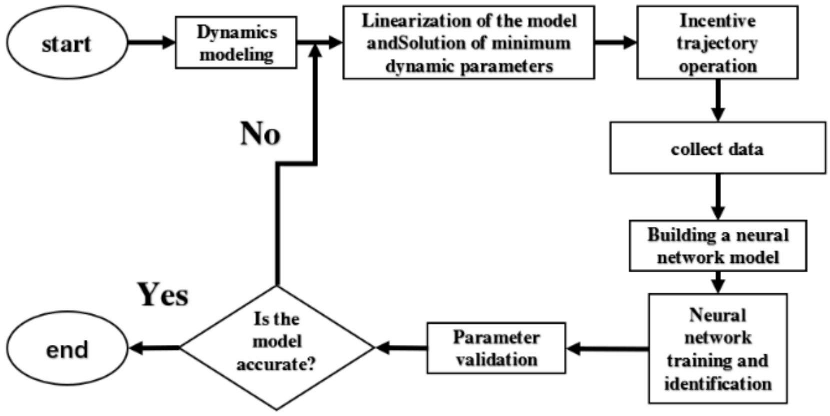

Fig 4. The Dynamic parameter identification process

3. Neural Network Modeling

3.1. BP Neural Network Model

A Back-Propagation Neural Network (BPNN), or back-propagation neural network, constitutes a multi-layer feed-forward neural network with the capacity to accommodate intricate non-linear correlations through the implementation of a back-propagation algorithm for the purposes of network training. A BP neural network typically comprises three layers: an input layer, a hidden layer and an output layer. Each layer is comprised of multiple neurons, which are connected to one another by means of weights. Each neuron performs a weighted sum of the received signals and then performs a non-linear transformation through an activation function to produce an output. The input signals of a BP neural network originate in the input layer and are conveyed to the output layer in a successive manner. Subsequently, the output of the network is compared with the true value, and the loss function is calculated. The weights of the network are then adjusted according to the loss function with the objective of reducing the error and updating the weights. Finally, the iterations are repeated in order to achieve the predetermined accuracy.

From the relationship in Equation (16) between the joint torque and the minimal set of dynamic parameters, it can be inferred that by assigning physical meaning to each weight, the trained weights correspond to the dynamic parameters of the robotic arm.

In BP neural networks, a multitude of neurons are interconnected in this manner to form a sophisticated network structure, enabling the network to discern intricate patterns and characteristics within the data.

Neural networks construct more complex models by introducing hidden layers, which have their own internal weights and biases. These hidden layers enable the network to learn nonlinear relationships and deeper features within the data. The hidden layers are one or more intermediate layers that are not directly connected to the input or output. Neurons in the hidden layers receive signals from the input layer, and after processing with weights and biases, the activation function transforms these signals into new representations.

The inclusion of hidden layers increases the network's parameters, weights, and biases, enhancing its parameter capacity and allowing it to fit more complex functions. Each layer further abstracts and generalizes from the previous layer, which helps the neural network better understand and process the data.

3.2. Neural Network Construction

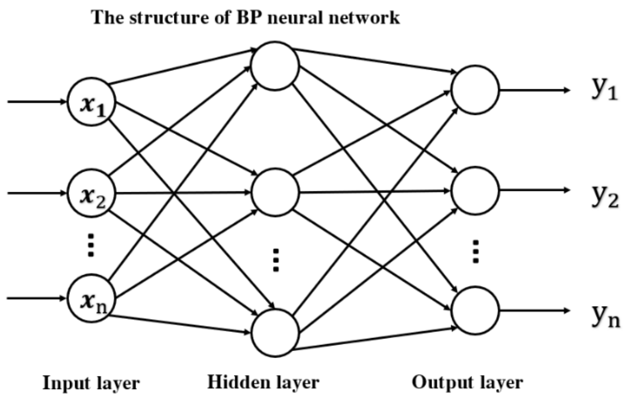

Fig 5. BP Neural network construction

3.3. The Selection of the Activation Function

In neural network models, the choice of activation function has a crucial impact on the model's learning performance and nonlinear fitting ability. The Sigmoid function is the most traditional and commonly used nonlinear activation function, and its mathematical expression is:

$$\sigma(x)=\frac{1}{1+e^{-x}} \tag{18}$$

The Sigmoid function works well for logistic regression and binary classification problems, but it suffers from the vanishing gradient problem when the input values approach infinity or negative infinity. Therefore, this paper uses the ReLU function, which is commonly used in modern deep learning:

$$\sigma(x)=\max(0,x) \tag{19}$$

The ReLU function has a simple expression, involving only a basic threshold operation. It requires less computation during optimization, is more efficient, and offers strong flexibility and generalizability.

3.4. The Selection of Training and Optimization Methods for Neural Network Models

To measure the difference between the predicted values and the actual values of the neural network model, and to accurately predict or classify new data, the loss function J is defined using the Root Mean Square Error RMSE of the joint torque [13]:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\tau_i - \hat{\tau}_i\right)^2} \tag{20}$$

where N is the total number of sample data, τ_i is the actual value of data point i, and $\hat{\tau}_i$ is the predicted value of data point i.

Typically, gradient descent is chosen as the optimization algorithm to minimize the loss function. By randomly initializing the model's weights and biases, the derivative of the loss function with respect to each parameter is computed to obtain the gradient of the loss function. The parameters are then updated based on the gradient and a predefined step size:

$$\theta = \theta - \alpha \bullet \nabla_\theta J(\theta) \tag{21}$$

where θ represents the parameters, ∝ is the learning rate, and $\nabla_\theta J(\theta)$ is the gradient of the loss function J with respect to the parameter θ

By continuously calculating and updating the gradient until the minimum value of the loss function is found, and the termination criteria are met, the training of the BP neural network model is completed.

4. Validation of the BP Neural Network-Based Parameter Identification Model

4.1. Experimental Design

This study validates the accuracy of the identification model through simulation. A six-degree-of-freedom serial robotic arm is constructed using the Matlab R2024a Robotics Toolbox as the parameter identification object, with theoretical dynamic parameters defined. Input data for the identification model is generated through data construction. The process is as follows:

(1) Define the excitation trajectory and obtain the corresponding joint sequence for executing the trajectory through inverse kinematics of the robotic arm, then save the data.

(2) Input the dynamic parameters and joint sequence into the dynamic model to compute the joint output torques, and save the results.

(3) Feed the joint angle and torque data into the proposed identification network. Use the trained neural network to predict the identification results, compare them with the ground truth, and output the identification error.

(4) Provide an arbitrary validation motion trajectory and compare the actual joint output torque with the theoretical torque based on the identification results.

In the neural network model identification experiment, the more trajectory data available, the more accurate the identification model becomes. Multiple sets of trajectory data were collected for the experiment, with each set containing a large number of joint data points. These data sets were divided into a training set (80%) and a validation set (20%), where the training set is used for training the neural network, and the validation set is used to assess the model's prediction performance. During the training process, the size of the loss function and training accuracy are monitored in real time. As the model iterates, its accuracy continuously improves, reaching a maximum value, while the loss function steadily decreases and approaches zero.

4.2. Identification Results

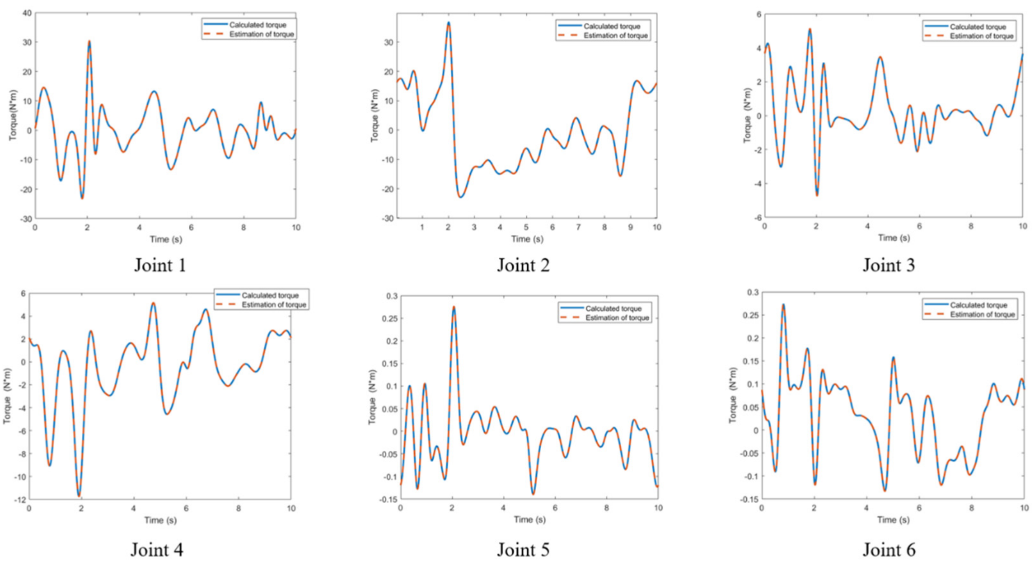

Fig 6. Predicted moments for each joint at the validation trajectory

The model output torques and desired output torques are visualized as shown in the figure. The results show that, on the validation trajectory, the torques predicted by the BP neural network are highly consistent with the theoretical values. The prediction performance on complex motion trajectories is excellent. This can be attributed to the fact that the neural network does not require explicit modeling of nonlinear factors and is capable of handling nonlinear relationships. Additionally, the network demonstrates stronger robustness to noise, leading to better control performance and higher accuracy.

In addition, the torque curve obtained from the model is smooth, with small torque errors and peak errors. Compared to traditional methods, the BP neural network-based dynamic parameter identification method demonstrates stronger robustness and generalization capabilities, making it more suitable for high-precision control scenarios in industrial robots. Future work could focus on further optimizing the network structure and training algorithms to enhance efficiency and adapt to more complex task requirements.

5. Conclusion

This paper proposes a BP neural network-based dynamic parameter identification method aimed at improving the accuracy of robot dynamic parameter identification, thereby achieving high-precision torque control. By utilizing Fourier polynomial trajectory planning and constructing the BP neural network, the identification accuracy is effectively enhanced. Finally, the dynamic parameter identification simulation is conducted using a six-degree-of-freedom robotic arm model constructed in the Matlab Robotics Toolbox, validating the accuracy and effectiveness of the model.

References

[1] Robot Industry, 2022, (04):83-95. DOI:10.19609/j.cnki.cn10-1324/ tp. 2022.04.004.

[2] M. Gautier, W. Khalil. Exciting trajectories for the identification of base initial parameters ofrobots [J]. International journal or robotic research, 1992, 11 (4): 362-375.

[3] Identification of dynamic parameters of modular robot considering joint flexibility by Zhou Jun and Yu Yueqing[J]. Robot, 2011,33(04):440-448.

[4] Feng Limin, Yu Jinghu, Wang Yanyu, et al Identification of Robot Dynamic Parameters Based on IRLS Algorithm [J]. Modern Manufacturing Engineering, 2022(4):37-44.

[5] Modeling and parameter identification of leg dynamics of equirope-driven bipedal robot by Zhu Zhenbiao, Tang Jun, Liang Zhiyuan[J].Mechanical Transmission,2024,48(08):18-25.DOI:10.16578/j.issn.1004.2539.2024.08.003.

[6] Identification of dynamic parameters of Zhang Shiyuan, Dai Jun, and Deng Hua six-degree-of-freedom manipulator [J]. Manufacturing Automation,2021,43(03):35-39.

[7] Chen Zhihuan, Liang Chenglong, Dong Jiashun, Chaotic Quantum Whale Optimization Algorithm and Its Application [J]. Computer Simulation,2024,41(08):439-446+465.

[8] Liu Yu, Li Guixian, Xia Dan, et al. Identification of Dynamic Parameters of Space Robots Based on Improved Genetic Algorithm [J].Journal of Harbin Institute of Technology, 2010, 42(11):1734-1739.

[9] Minglu Zhang, Qing Wang, Xuan Liu, et al. Identification of Dynamic Parameters of Torque Compensation Based on Neural Network[J].Journal of Tianjin University (Natural Science and Engineering Technology), 2024, 57(07): 759-767.

[10] Zhang Di's trajectory tracking control of robot arm based on recurrent neural network[J].Mechanical Engineering and Automation, 2024,(04):19-21.

[11] Craig, J.JAuthors; Translated by Yi Chao et al. . Introduction to Robotics: Mechanics and Control[M]. Beijing: China Machine Press, 2006.

[12] Zhu Miao's research on the identification of dynamic parameters of SCARA robot considering joint flexibility[D]. Guangdong University of Technology,2022. DOI:10. 27029/d. cnki. ggdgu.2022.000121.

[13] Qi Changgui's contour error prediction and compensation of numerical control system based on LSTM neural network [D]. Harbin Institute of Technology,2021. DOI: 10.27061/ d.cnki. ghgdu. 2021.001067.